

---

# **MOTECH Modules Documentation**

***Release 0.24-SNAPSHOT***

**MOTECH Team**

August 19, 2015



<b>1</b>	<b><i>Alerts</i></b>	<b>1</b>
<b>2</b>	<b><i>Appointments</i></b>	<b>3</b>
<b>3</b>	<b><i>Batch</i></b>	<b>5</b>
<b>4</b>	<b><i>CMS Lite</i></b>	<b>7</b>
<b>5</b>	<b><i>CommCare</i></b>	<b>9</b>
<b>6</b>	<b><i>Data Services</i></b>	<b>11</b>
<b>7</b>	<b><i>Email</i></b>	<b>13</b>
<b>8</b>	<b><i>Event Logging</i></b>	<b>15</b>
<b>9</b>	<b><i>Hindi Transliteration</i></b>	<b>17</b>
<b>10</b>	<b><i>Hub</i></b>	<b>19</b>
<b>11</b>	<b><i>Message Campaign</i></b>	<b>21</b>
<b>12</b>	<b><i>mTraining</i></b>	<b>23</b>
<b>13</b>	<b><i>OpenMRS</i></b>	<b>25</b>
<b>14</b>	<b><i>Pill Reminder</i></b>	<b>27</b>
<b>15</b>	<b><i>Schedule Tracking</i></b>	<b>29</b>
<b>16</b>	<b><i>Scheduler</i></b>	<b>31</b>
<b>17</b>	<b><i>IVR</i></b>	<b>33</b>
<b>18</b>	<b><i>SMS</i></b>	<b>35</b>
<b>19</b>	<b><i>Tasks</i></b>	<b>37</b>
19.1	Alerts Module . . . . .	37
19.2	Appointments Module . . . . .	41
19.3	Batch Module . . . . .	44
19.4	CMS Lite Module . . . . .	44

19.5	CommCare Module . . . . .	46
19.6	Event Logging Module . . . . .	46
19.7	Hindi Transliteration Module . . . . .	51
19.8	HTTP Agent Module . . . . .	51
19.9	Hub Module . . . . .	51
19.10	IVR Module . . . . .	52
19.11	Message Campaign Module . . . . .	58
19.12	mTraining Module . . . . .	73
19.13	OpenMRS Module . . . . .	74
19.14	Javadoc . . . . .	74
19.15	Pill Reminder Module . . . . .	525
19.16	Schedule Tracking Module . . . . .	527
19.17	SMS Module . . . . .	535

---

***Alerts***

---

Collects alerts for users in an inbox-like container



---

***Appointments***

---

Provides appointment scheduling and reminders





---

### ***Batch***

---

An implementation of Spring batch (version: 3.0.0.M3); it essentially deals with scheduling triggering of jobs



---

***CMS Lite***

---

Provides basic content storage and retrieval



---

***CommCare***

---

Integrates the MOTECH platform with CommCareHQ, an open-source platform to help manage community health workers



---

***Data Services***

---

Integrates data from external data sources and provides sharable data schemas





---

***Email***

---

Sends and logs email messages



---

## ***Event Logging***

---

Allows MOTECH modules to easily see each others' events



---

## ***Hindi Transliteration***

---

Supports transliteration of English strings to Hindi using ITRANS encoding



---

### *Hub*

---

Provides an implementation of the PubSubHubbub Hub spec; exposes an API so other modules can act as publisher and make contents available to it for distribution





---

***Message Campaign***

---

Enrolls users in message campaigns with flexible content-scheduling rules



---

### ***mTraining***

---

Provides data containers and APIs for defining mobile (e.g. SMS or IVR-based) training courses and tracking user enrollment and progress



---

***OpenMRS***

---

Integrates the MOTECH platform with OpenMRS, an open source electronic medical record platform



---

***Pill Reminder***

---

A flexible reminder system that may be used to alert patients when it is time to take their medications





---

## ***Schedule Tracking***

---

Enrolls users for alerts based on complex scheduling rules



---

### ***Scheduler***

---

Publishes events on a schedule, using the open source Quartz engine.



---

### ***IVR***

---

Integrating the MOTECH platform with a Interactive Voice Response (IVR) providers thus enabling support for voice/audio dialogs



---

### ***SMS***

---

Provides a basic specification for integrating the MOTECH platform with an SMS provider to send/receive SMS messages





---

## Tasks

---

Allows administrative users to author simple “tasks” that wire up different modules; for example, a task can be created to enroll a patient in a message campaign in response to an incoming SMS message containing specific text

## 19.1 Alerts Module

### Contents

- Alerts Module
  - Introduction
  - Alert
    - \* Alert fields
    - \* Alert type
    - \* Alert status
  - OSGI Service API
    - \* Public API
    - \* Criteria
      - Search Criteria
      - Update Criteria
  - Alert Events
    - \* Events
    - \* Creating Alert
    - \* Updating Alert
  - Configuration

### 19.1.1 Introduction

With the Alerts module you can store and edit information about alerts for users identified by a supplied external id. These operations can be performed using an OSGI service or by firing events. The way you choose depends entirely on you and on your implementation.

### 19.1.2 Alert

The Alerts module is a database structure for storing alerts. The module also listens for *the specific events associated with alerts*. The true meaning of some fields located in a single alert depends on your implementation(alertType, priority, externalId).

## Alert fields

Name	Description	Type
id	Alert id generated by the system.	Long
externalId	External id of the person who is associated with this alert. This is an arbitrary value controlled by your application. This field is required.	String
alertType	<i>Alert type</i> , this field is required.	org.motechproject.alerts.domain.AlertType
dateTime	Alert date.	org.joda.time.DateTime
priority	Alert priority, this field is required and it's used in Alert compareTo method(It works in the same way as Long.compare(priority1, priority2)). The true meaning of the priority depends on your implementation.	Long
status	<i>Alert status</i> , this field is required.	org.motechproject.alerts.domain.AlertStatus
description	Alert description.	String
data	Key/Value pairs.	Map<String, String>

## Alert type

org.motechproject.alerts.domain.AlertType

- CRITICAL
- HIGH
- MEDIUM
- LOW

For example CRITICAL may indicate that, if an alert will not be readed by the patient then it can have very bad consequences(For example, a reminder of taking the medicines).

## Alert status

org.motechproject.alerts.domain.AlertStatus

- NEW - Alert has not been read.
- READ - Alert was readed.
- CLOSED - Alert is closed.

## 19.1.3 OSGI Service API

### Public API

- `void create(String entityId, String name, String description, AlertType type, AlertStatus status, int priority, Map<String, String> data);` - create a new alert.
- `List<Alert> search(AlertCriteria alertCriteria)` - finds alerts matching the specified search criteria.

- `Alert get(Long id)` - finds an alert by ID.
- `void update(Long alertId, UpdateCriteria updateCriteria)` - updates an alert with given ID.

Deprecated methods:

- `void update(String alertId, UpdateCriteria updateCriteria)`
- `Alert get(String id)`

## Criteria

### Search Criteria

To search alerts using OSGI Service you must create a `org.motechproject.alerts.contract.AlertCriteria` object which specifies the search criteria. To add filters, use the following methods:

- `AlertCriteria byExternalId(String id)` - Adds criterion to search by external id of alert.
- `AlertCriteria byStatus(AlertStatus status)` - Adds criterion to search by status of alert.
- `AlertCriteria byType(AlertType type)` - Adds criterion to search by type of alert.
- `AlertCriteria byPriority(Integer priority)` - Adds criterion to search by priority of alert.
- `AlertCriteria byDateRange(DateTime from, DateTime to)` - Adds criterion to search by the date of alert in the given range.

```
AlertCriteria searchCriteria = new AlertCriteria();
searchCriteria.byExternalId("personId"); //primary criterion
searchCriteria.byType(AlertType.MEDIUM);
searchCriteria.byStatus(AlertStatus.NEW);
List<Alert> alerts = alertService.search();
```

First added filter is primary criterion. This means that records will be retrieved from the database on the basis of it's value. Fetched records are filtered on the basis of the following criteria. If you do not add any filters, all records will be retrieved.

### Update Criteria

To update alerts using OSGI Service you must create a `org.motechproject.alerts.contract.UpdateCriteria` object which specifies the update criteria. To change an alert values, use the following methods:

- `UpdateCriteria status(AlertStatus newStatus)` - Marks that status field of alert is to be updated and stores the new value.
- `UpdateCriteria name(String newName)` - Marks that name field of alert is to be updated and stores the new value.
- `UpdateCriteria description(String newDescription)` - Marks that description field of alert is to be updated and stores the new value.
- `UpdateCriteria priority(int newPriority)` - Marks that priority field of alert is to be updated and stores the new value.
- `UpdateCriteria data(Map<String, String> newData)` - Marks that data field of alert is to be updated and stores the new value.

```
UpdateCriteria updateCriteria = new UpdateCriteria();
updateCriteria.status(AlertStatus.READ);
updateCriteria.name("newName");
updateCriteria.description("newDescription");
updateCriteria.priority(2);
updateCriteria.data(newData);
alertService.update(alert.getId(), updateCriteria);
```

## 19.1.4 Alert Events

### Events

The module handles the event with the following subjects (In parentheses are the names of constants, which are located in `org.motechproject.alerts.EventKeys`):

- `org.motechproject.alerts.api.Create.Alert` (`CREATE_ALERT_SUBJECT`)
- `org.motechproject.alerts.api.Close.Alert` (`CLOSE_ALERT_SUBJECT`)
- `org.motechproject.alerts.api.Mark.Alert.Read` (`MARK_ALERT_READ_SUBJECT`)

To create new alert using event you must put to the event object parameters corresponding to the alert fields:

- “ExternalId” - required (`EXTERNAL_ID_KEY`)
- “AlertName” - not required (`ALERT_NAME`)
- “AlertType” - required (`ALERT_TYPE`)
- “AlertPriority” - required (`ALERT_PRIORITY`)
- “AlertStatus” - required (`ALERT_STATUS`)
- “AlertDescription” - not required (`ALERT_DESCRIPTION`)
- “AlertData” - not required (`ALERT_DATA`)

To close alert or mark it as read you must put to the event object parameter “AlertId” (`ALERT_ID`). To create and fire events you will need MOTech Platform Event module, so you must add following dependency:

```
<dependency>
  <groupId>${project.groupId}</groupId>
  <artifactId>motech-platform-event</artifactId>
  <version>${motech.version}</version>
</dependency>
```

To fire events you must use OSGI Service, so you will need add reference to it in the blueprint.xml in your module:

```
<osgi:reference id="eventRelay" interface="org.motechproject.event.listener.EventRelay"/>
```

### Creating Alert

```
Map<String, Object> params = new HashMap<>();
params.put(EventKeys.EXTERNAL_ID_KEY, "personId");
params.put(EventKeys.ALERT_NAME, "alertName");
params.put(EventKeys.ALERT_DESCRIPTION, "sample description");
params.put(EventKeys.ALERT_TYPE, AlertType.MEDIUM);
params.put(EventKeys.ALERT_STATUS, AlertStatus.NEW);
params.put(EventKeys.ALERT_PRIORITY, 3);
```

```
params.put(EventKeys.ALERT_DATA, alertData);
MotechEvent createAlertEvent = new MotechEvent(EventKeys.CREATE_ALERT_SUBJECT, params);
eventRelay.sendEventMessage(createAlertEvent);
```

## Updating Alert

Updates in this context is understood as closing alert or marking it as read.

```
Map<String, Object> params = new HashMap<>();
params.put(EventKeys.ALERT_ID, alertId);
MotechEvent updateAlertEvent = new MotechEvent(EventKeys.MARK_ALERT_READ_SUBJECT, params);
eventRelay.sendEventMessage(updateAlertEvent);
```

## 19.1.5 Configuration

This module does not require any custom configuration.

# 19.2 Appointments Module

### Contents

- Appointments Module
  - Description
  - Appointment
    - \* Appointment fields
    - \* Appointment status
    - \* Creating and Updating appointments
  - Scheduled reminders
  - OSGI Service API
  - Configuration

## 19.2.1 Description

The module provides possibility to create appointments. Each appointment is associated with a specific person and contains information about the planned visit. This module does not store the information about patients, or does not specify the way in which the patient receives a reminders about their appointments. The Appointments module creates *for each appointment scheduled reminders* which sends events at a specified time period. So the way in which patient is informed depends only on the manner in which you handled the event. You decide when reminder will start and how often event will be fired, *you must simply set the appropriate parameters*.

**Attention:** This module requires an installed and enabled Scheduler module for scheduling reminders.

## 19.2.2 Appointment

### Appointment fields

Each appointment contains following fields:

Field	Description	Type
externalId	External id of the person who is associated with this appointment. This is an arbitrary value controlled by your application. It allows tying appointment reminders with patients at their fire time.	String
apptId	Appointment id generated by the system	String
appointmentDate	Date for the appointment, events after this date will no longer be sent	org.joda.time.DateTime
visitedDate	Date of the fulfilled visit	org.joda.time.DateTime
status	Appointment status	org.motechproject.appointments.domain.AppointmentStatus
sendReminders	Flag to activate or deactivate sending reminders	bool
reminderInterval	Reminder interval	org.joda.time.Period
reminderStartTime	Time to start firing the reminder events	org.joda.time.DateTime

## Appointment status

Status can take one of the following values:

- NONE - default value for the appointment.
- MISSED - missed appointment.
- UNSCHEDULED - unscheduled visit to the clinic.
- VISITED - patient visited the clinic.
- CONFIRMED - appointment confirmed before the visit.
- REMOVED - appointment was canceled by the patient or provider.

## Creating and Updating appointments

The obligation to create and update an appointments is on the side of implementation. The Appointment module provides the *OSGI service* to perform this operations. When `appointmentDate` or `status` will be changed then the old version of the appointment will be saved to the appointments logs.

**Attention:** Please note that when you update `sendReminders` to false value then reminders for this appointments will not be send

### 19.2.3 Scheduled reminders

For each created appointment Scheduler module performs a period repeating job (job will start if value of the `sendReminders` parameter is true). This job is built on the basis of information contained in each appointment:

- `appointmentDate` - planned date of the visit(reminder end time).
- `reminderStartTime` - reminder start time.
- `reminderInterval` - the interval at which the reminders will be fired by the Scheduler module.

When a reminder triggers, an event will be sent with the Scheduler module will send `event(org.motechproject.event.MotechEvent)` with following subject `Appointment.Reminder.{1}.{2}`. First parameter is patient external id, second is appointment id generated by the system.

### 19.2.4 OSGI Service API

The Appointments module exposes an OSGi services `org.motechproject.appointments.service.AppointmentService` to perform common operations on patients appointments. The service provides the following API:

- `Appointment addAppointment(Appointment appointment)` - create a new appointment, `externalId` param can't be null(it associates appointment with the patient).
- `List<Appointment> addAppointments(List<Appointment> appointments)` - create a new appointments.
- `void removeAppointment(String appointmentId)` - remove appointment with given ID, and remove reminder created for it.
- `void removeAppointments(List<String> appointmentId)` - remove appointments with given IDs.
- `Appointment updateAppointment(Appointment appointment)` - update an appointment.
- `List<Appointment> updateAppointments(List<Appointment> appointments)` - updates the list of appointments.
- `Appointment getAppointment(String appointmentId)` - get the appointment with given ID.
- `List<Appointment> findAppointmentsByExternalId(String externalId)` - get the appointments which are associated with person `externalId`.
- `List<Appointment> findAppointmentsByExternalId(String externalId, AppointmentStatus status)` - get the appointments which are associated with person `externalId` and which have given status.
- `void toggleReminders(String externalId, boolean sendReminders)` - toggle the reminders for a patient with `externalId`. If `sendReminders == true` then send reminders for patient, otherwise stop reminders for patient.

The following is an example of creating a new appointment.

```
Appointment appointment = new Appointment();
DateTime reminderStartTime = DateTime.now().plusDays(1); // start send events tomorrow
DateTime appointmentDate = new DateTime(2014, 6, 30, 17, 0, 0, 0); // 2014-06-30 17:00:00
Period reminderInterval = new Period(24, 0, 0, 0); //reminder will be sent every 24 hours from reminder

appointment.setExternalId(patientID);
appointment.setStatus(AppointmentStatus.CONFIRMED); // Visit is confirmed
appointment.setReminderStartTime(reminderStartTime);
appointment.setAppointmentDate(appointmentDate);
appointment.setReminderInterval(reminderInterval);
appointment.setSendReminders(true); // Enable reminders

appointmentService.addAppointment(appointment); // add new appointment
```

### 19.2.5 Configuration

This module does not require any custom configuration.

## 19.3 Batch Module

To be written.

If you'd like to author this topic, please assign yourself [MOTeCH-963](#).

## 19.4 CMS Lite Module

### Contents

- [CMS Lite Module](#)
  - [Description](#)
  - [Information for Implementation](#)
  - [OSGi Service APIs](#)
  - [Retrieving content through HTTP](#)
  - [Events Consumed and Emitted](#)
  - [Roles and Permissions](#)

### 19.4.1 Description

CMS Lite is a very simple CMS that allows for content to be entered and served. Content should be associated with a key and include attributes such as language and format. Content is provided both via API and HTTP. Content can be accessed in a single return value or streamed.

This module is typically used by a campaign module (e.g. Message Campaign or Pill Reminder) to store the audio or string content used for IVR calls or SMS messages. If a more sophisticated CMS is needed, implementers are encouraged to make use of one of the multitude of existing open source CMS solutions.

### 19.4.2 Information for Implementation

The CMS Lite module allows implementers to:

- Name content and associate it with a language
- Associate either strings or streams with content
- Test for the existence of content
- Load or stream content via API or HTTP

Content is stored in MOTeCH Data Services (MDS).

### 19.4.3 OSGi Service APIs

The CMS Lite module exposes its operations through an OSGi service interface called `CMSLiteService`. It also exports the `StreamContent` and `StringContent` classes as part of its data model, which are used in the APIs and examples below.



## CMSLiteService

CMSLiteService supports storing and retrieving string and stream content, and checking whether content exists in the CMS.

```
public interface CMSLiteService {

    StreamContent getStreamContent(String language, String name) throws ContentNotFoundException;

    StringContent getStringContent(String language, String name) throws ContentNotFoundException;

    void removeStreamContent(String language, String name) throws ContentNotFoundException;

    void removeStringContent(String language, String name) throws ContentNotFoundException;

    void addContent(Content content) throws CMSLiteException;

    boolean isStreamContentAvailable(String language, String name);

    boolean isStringContentAvailable(String language, String name);

    List<Content> getAllContents();

    StringContent getStringContent(String stringContentId);

    StreamContent getStreamContent(String stringContentId);

    Byte[] getDetachedField(StreamContent instance);

    List<StringContent> getAllStringContents();

    List<StreamContent> getAllStreamContents();

    void deleteAllContents();

}
```

### Adding String Content

```
StringContent stringContent = new StringContent("language", "name", "value");
cmsLiteService.addContent(stringContent);
```

### Adding Stream Content

```
// It's assumed that you have created inputStream as an instance of java.io.InputStream
StreamContent streamContent = new StreamContent("language", "name", inputStream, "checksum", "audio/");
cmsLiteService.addContent(streamContent);
```

### Testing Whether Content Exists

```
if (cmsLiteService.isStreamContentAvailable("language", "name")) {
    // Content exists
}
```

### 19.4.4 Retrieving content through HTTP

Content stored in the cms-lite module can be retrieved by doing a single HTTP GET request. Two HTTP endpoints are exposed - one for string content, the other one for stream content. You need to provide the language and name of the desired resource in the url.

- String content:

```
http://{motech-server}/module/cmsliteapi/string/{language}/{name}
```

- Stream content:

```
http://{motech-server}/module/cmsliteapi/stream/{language}/{name}
```

---

**Note:** There are no special security rules defined for these endpoints, meaning that the client must handle authenticating with MOTECH in order to be able to use them. You can set up security rules that will disable the security for these endpoints in case for example your IVR requires to retrieve them without auth. Doing that will make them publicly available though, so use your own judgement regarding this. Check the [Security Rules](#) documentation for more info on setting up custom security rules.

---

### 19.4.5 Events Consumed and Emitted

This module does not consume or emit any events.

### 19.4.6 Roles and Permissions

This module does not define any roles or permissions.

## 19.5 CommCare Module

To be written.

## 19.6 Event Logging Module

**Contents**

- Event Logging Module
  - Introduction
  - LoggableEvent and EventLogger
  - EventLoggingServiceManager
  - Database logging
    - \* Description
    - \* DB Event Converter and Log Mapping
    - \* EventLog
    - \* Browsing Logs
  - File logging
    - \* Description
    - \* FileEventLoggingService
  - Parameters Flags
    - \* ParametersPresentEventFlag
    - \* RegexEventFlag

**19.6.1 Introduction**

The Event Logging Module allows you to log information about events fired in MOTECH. It is possible to filter which events get logged using `EventLoggers`. The module uses logging services to store information about these events. Each service must be registered using the `EventLoggingServiceManager` which will generate the appropriate listeners for the specified subjects. After registering these services, events with the matching subjects will be processed (logged) by them. The Event Logging Module provides two implementations of the logging service, `DbEventLoggingService` and `FileEventLoggingService`, but it is also possible to provide your own implementation.

**19.6.2 LoggableEvent and EventLogger**

The logging services use `EventLogger` implementations to log events. Each `EventLogger` contains data about events to log in the form of `LoggableEvent`. `EventLogger` contains followings methods:

- `void addLoggableEvents(List<LoggableEvent> loggableEvents)` - Adds data about events to log.
- `void removeLoggableEvents(List<LoggableEvent>)` - Removes data about events to log.
- `List<LoggableEvent> getLoggableEvents()` - Returns data about events to log.
- `void clearLoggableEvents()` - Removes all data about events to log.
- `abstract void log(MotechEvent eventToLog)` - Abstract method used to store data about events.

`LoggableEvent` class should be used to check whether the event should be stored by `EventLogger`. `LoggableEvent` contains method to check events(`boolean isLoggableEvent(MotechEvent eventToLog)`). This method checks event subject and compare event parameters with flags.

Field	Description	Type
eventSub- jects flags	List of event subjects which will be logged by <code>EventLogger</code> . Wildcards are allowed. List of <i>special flags</i> . They are used to filter events on the basis of the parameters.	List<String>  List<? extends EventFlag>

The `EventLoggingService` provides a method that allows you to retrieve set of all event subjects for which this service is listening.

```
-Set<String> getLoggedEventSubjects()
```

### 19.6.3 EventLoggingServiceManager

EventLoggingServiceManager is an OSGI service through which you can register or update your logging service. Its main function is to manage logging services. The manager keeps a list of all registered logging services.

- `void registerEventLoggingService(EventLoggingService eventLoggingService)` - registers MOTech Listener in MOTech Platform Event module for events with subjects concluded in EventLogger objects of given logging service.
- `void updateEventLoggingService(EventLoggingService eventLoggingService)` - updates MOTech Listener for events with subjects concluded in EventLogger objects of given logging service(Method not implemented yet).

We can create and register a new logging service as follows:

```
EventLoggingService newLoggingService = new EventLoggingServiceImpl();
eventLoggingServiceManager.registerEventLoggingService(newLoggingService);
```

### 19.6.4 Database logging

#### Description

The Event Logging Module automatically creates one DbEventLoggingService. By default, this service logs all events fired in MOTech. Information about events is stored in the database.

#### DB Event Converter and Log Mapping

##### Description

The DB loggers(DbEventLogger) uses a DefaultDbToLogConverter class to build records from events. The main objective of this component is to map the event parameters. This mapping process can remove, add or replace event parameters.

##### DefaultDbToLogConverter

DbEventLoggingService to convert events is using a DefaultDbToLogConverter. The conversion involves the removal of unwanted event parameters or their replacement to others. The conversion is made on the basis of the information contained in db loggers. This information is represented by LogMappings class which contains following fields.

Field	Description	Type
map-pings	List of KeyValue, which contains informations about <i>replacing event params</i> .	List<KeyValue>
ex-cludes	If events contains a parameter key which is included in this list then this param will be excluded from database log.	List<String>
in-cludes	If events contains a parameter key which is included in this list then this param will be included in database log. By default all parameters are included so you don't need to specify keys. You can use this param for example for include a parameter which was replaced by mappings content.	List<String>

KeyValue class contains following fields:

- `startKey (String)`
- `startValue (Object)`
- `endKey (String)`
- `endValue (Object)`

If we specify mappings section in `LogMappings` then Db log converter will map event parameters as follows. When an event contains parameter with key equals to `startKey` and this parameter has value equal to `startValue` then this parameter is replaced by parameter with key `endKey` and with value from `endValue`. So we can replace some parameters before logging an event.

### DbEventLoggingService configuration

The Event Logging Module to configure default database logging uses a `event-mappings.json` configuration file.

Below find an example of the `event-mappings.json` configuration file:

```
[
  {
    "subjects": [
      "org.sms",
      "org.ivr"
    ],
    "mappings": [
      {
        "status": "ok",
        "result": "success"
      },
      {
        "status": "error",
        "result": "failure"
      }
    ],
    "excludes": [
      "exclude1",
      "exclude2"
    ],
    "includes": [
      "include1",
      "include2"
    ],
    "flags": [
      {
        "keyValuePairsPresent": {
          "requiredParameter1": "value1",
          "requiredParameter2": "value2"
        }
      },
      {
        "keyValuePairsPresent": {
          "requiredParameter3": "value3"
        }
      }
    ]
  }
]
```

Below find a default configuration file:

```
[
  {
    "subjects": [
      "*"
    ]
  }
]
```

This configuration file is represented by `MappingsJson` class:

Field	Description	Type
subject	List of event subjects which will be log.	List<String>
map-pings	List of maps used for creating KeyValue list in LogMappings class.	List<Map<String, String>>
excludes	List of parameter keys which will be excluded from log.	List<String>
includes	List of parameter keys which will be included in log.	List<String>
flags	List of <i>ParametersPresentEventFlag</i> .	List<ParametersPresentEventFlag>

## EventLog

Represents a single, logged event record, that is persisted in the database. Each `EventLog` contains following fields:

Field	Description	Type
subject	Subject of logged event.	String
parameters	The parameters(key value pairs) of the logged event.	Map<String, Object>
timeStamp	Timestamp of logged event.	org.joda.time.DateTime

## Browsing Logs

You can retrieving event logs using the MDS user interface, or using the `DbEventQueryService` which is an OSGI service and allows to query for log records, given certain criteria.

- `getAllEventsBySubject(String subject)` - Retrieves all events that match a given subject.
- `getAllEventsByParameter(String parameter, String value)` - Retrieves all events that match a given parameter and key-value pair.
- `getAllEventsBySubjectAndParameter(String subject, String parameter, String value)` - Retrieves all events that match a given subject and parameter key-value pair.

## 19.6.5 File logging

### Description

The Event Logging Module provides possibility to store logs in files. To do this you must use `FileEventLoggingService`. By default, no `FileEventLoggingService` instance is created.

### FileEventLoggingService

`FileEventLoggingService` is using `FileEventLogger` instances to log events. When event arrives each logger added to `FileEventLoggingService` process it. Each `FileEventLogger` contains list of `File` objects. This list is used to create files if they not exist and for storing information about fired events.

FileEventLogger for converting events use DefaultFileToLogConverter. Which stores all event parameters to file.

To enable file logging you must add @Service annotation to the FileEventLoggingService and implement @PostConstruct method. In this method you should configure file logger. You can also implement constructor which will configure service. Last step is to register service instance by EventLoggingServiceManager.

```
@PostConstruct
public void configureFileLoggingService() {
    List<File> loggingFiles = new ArrayList<>();
    //Here we must add some files in which logs will be saved

    List<LoggableEvent> loggableEvents = new ArrayList<>();
    //Here we must specify some loggable events

    //creating and adding new logger
    FileEventLogger logger = new FileEventLogger(loggableEvents, loggingFiles, eventConverter);
    fileEventLoggers.add(logger);
}
```

Below you can see a sample file content.

```
EVENT: eventSubject1 at TIME: 2015-04-17T14:20:45.713+02:00 with PARAMETERS: param1/1 param2/5 param3/6
EVENT: eventSubject2 at TIME: 2015-04-17T14:21:45.713+02:00 with PARAMETERS: param1/3 param2/value param3/6
EVENT: eventSubject3 at TIME: 2015-04-17T14:22:45.713+02:00 with PARAMETERS: param1/4 param2/value param3/6
```

## 19.6.6 Parameters Flags

### ParametersPresentEventFlag

This class is used by LoggableEvent class for filter events by parameters.

Field	Description	Type
keyValuePairsPresent	Key-value pairs representing parameters and their values which are required in the event to log.	Map<String, String>

### RegexEventFlag

Not implemented yet.

## 19.7 Hindi Transliteration Module

To be written.

If you'd like to author this topic, please assign yourself [MOTECH-1275](#).

## 19.8 HTTP Agent Module

## 19.9 Hub Module

To be written.

If you'd like to author this topic, please assign yourself [MOTech-964](#).

## 19.10 IVR Module

### 19.10.1 Overview

The IVR module can initiate outbound IVR calls and receive call status from IVR vendors. Text templates, typically CCXML/VXML, are stored in `Template` entities. Specifics about vendors are stored in `Config` entities.

### 19.10.2 /status REST request

IVR Call status information is sent to the IVR module by the IVR provider using REST calls, typically using CCXML's `<send>` element or VXML's `<data>` element. The IVR module stores the call status information in `CallDetailRecord` entities (CDR) in the database.

#### REST call

The IVR provider makes a GET or POST HTTP call to Motech & the IVR module:

```
http://{motech-server}/{motech-war-file}/module/ivr/status/{config}
```

Where `{motech-server}` is the address of the Motech server, `{motech-war-file}` is the name of the Motech war file, by default `motech-platform-server` and `{config}` is the name of the configuration representing the IVR provider.

The GET or POST parameters provide Motech with the information about the IVR call status. The following parameters are interpreted by Motech:

**from** is the phone number making the call.

**to** is the phone number receiving the call.

**callDirection** should be either `INCOMING` or `OUTGOING`. Any other string will be added to the `providerExtraData` map and `callDirection` will be `UNKNOWN`.

**callStatus** should be either one of `INITIATED`, `IN_PROGRESS`, `ANSWERED`, `BUSY`, `FAILED`, `NO_ANSWER`. Any other string will be added to the `providerExtraData` map and `callStatus` will be `UNKNOWN`.

**motechCallId** if available (ie: included as a parameter in the outgoing call HTTP call), the Motech originated GUID uniquely identifying this call.

**providerCallId** if available, the provider's call id. For example, CCXML's `session.id` would be a good choice.

**timestamp** if provided, a string representing the time at which the status is being sent, if not, the time at which the status is being received by Motech.

---

**Note:** Any parameters not recognized in the list above will be added to the `providerExtraData` map. Parameters listed in the config's `ignoreStatusFields` list are ignored. Should the IVR provider not be able to name some of the REST call parameters, the config's `statusFieldMap` can be used.

---



Each successful REST call results in one new `CallDetailRecord` database record.

## Motech Event

In addition to storing CDR in the database, the IVR module also sends a `org.motechproject.event.MotechEvent` for each IVR Call Status REST call:

```
subject ivr_call_status

payload TIMESTAMP, CONFIG, FROM, TO, CALL_DIRECTION,
         CALL_STATUS, MOTECH_CALL_ID, PROVIDER_CALL_ID,
         PROVIDER_EXTRA_DATA
```

## Motech Task Trigger

The `ivr_call_status` event subject is also a Motech Task Trigger.

### 19.10.3 /template REST request

Text<sup>1</sup> template requests are sent to the IVR module by IVR providers using REST calls, typically using VXML's `<submit>` element. Apart from returning text templates, this REST request works exactly the same as the */status REST request*, it stores `CallDetailRecord` entities, sends Motech events and triggers Motech Tasks.

## Velocity

Essentially, `Velocity` is a template engine which enables you to dynamically return text. Here's a simple template named `hello`:

```
Hello, $world.
```

Assuming Motech is running on your local machine and you created an IVR config named `voxeo`, you can get the IVR module to return the template by CURLing:

```
$curl -w "\n" "http://localhost:8080/motech-platform-server/module/ivr/template/voxeo/hello"
Hello, $world.
$
```

Nothing special, eh? But now add a query parameter named `world` with some value:

```
$curl -w "\n" "http://localhost:8080/motech-platform-server/module/ivr/template/voxeo/hello?world=Frank"
Hello, Frank.
$
```

All query parameters are available. But wait, there's more! Read on...

## The `$dataServices` element

In addition to query parameters, the special element `$dataServices`<sup>2</sup> is available inside your templates. It can be used to query the *database* using the following methods:

---

<sup>1</sup> Text or CCXML or VXML or anything really. Some non VXML providers like, for example, India's `KooKoo`, operate the same way except the templates are not VXML but generally some sort of proprietary XML.

<sup>2</sup> Using a query parameter named `$dataServices` is not a good idea and will produced undefined results.

- `findOne(entityClassName, lookupName, params)`: returns one entity instance
- `findMany(entityClassName, lookupName, params)`: returns a list of entity instances
- `count(entityClassName, lookupName, params)`: returns a number of entity instances
- `retrieveAll(entityClassName)`: returns all instances of an entity
- `countAll(entityClassName)`: returns the number of all instances of an entity

The methods above use the following arguments:

**entityClassName** the fully qualified class name for that entity, for example for a DDE<sup>3</sup> `org.motechproject.ivr.domain.CallDetailRecord` or for a EUDE<sup>4</sup> named `Patient`: `org.motechproject.mds.entity.Patient`

**lookupName** the name<sup>5</sup> of the lookup to use

**params** a map containing zero or more key:value pairs corresponding to the arguments required by the given lookup, see how to use a map in the following sample template.

So, let's say, for example, we created a `Patient` MDS entity with a `name` and a `number` field and a 'Lookup by Number' lookup which takes a `number` argument. The following template would extract the name of the patient whose number is '123':

```
Hello, $dataServices.findOne("org.motechproject.mds.entity.Patient", "Lookup by Number", {"number": "123"})
```

## Injecting custom services

Not only the `$dataServices` element can be used in IVR templates. It is also possible to inject any arbitrary OSGi service into the Velocity context. All services configured in the `servicesMap` field of the *configuration* field will be available to the template executed with that configuration. All these services will be injected as variables, so for example if your configuration is as follows: `myService:org.example.service.MyService`, then `org.example.service.MyService` will be available as `$myService` in the Velocity template.

## REST call

The IVR provider makes a GET or POST HTTP call to Motech & the IVR module:

```
http://{motech-server}/{motech-war-file}/module/ivr/template/{config}
```

See */status REST request* for additional details.

## Motech Event

The event sent is similar to that in */status REST request* with two exceptions: the subject is **ivr\_template\_request** and the event payload contains an additional `template` element which contains the name of the requested template.

---

<sup>3</sup> Developer Defined Entity

<sup>4</sup> End User Defined Entity

<sup>5</sup> Don't confuse the lookup name (ie: 'Find by name') with the lookup method name (ie: 'findByName').

## Motech Task Trigger

The Motech Task Trigger is also similar to that in */status REST request* with the same two exceptions as above, a different title and an additional element, you guessed it: the template name, to the payload.

### 19.10.4 Initiating Outbound Calls

To initiate an outbound call from an IVR provider, the IVR makes a REST call to the IVR provider. The following two parameters are required:

**configName** the name of the IVR provider config where `outgoingCallUriTemplate` specifies the IVR provider outbound call URI and `jsonRequest` specifies the HTTP POST method format

**params** the parameters needed by the IVR provider to make the call, eg: destination number, resource id, status callback URI, security credentials, etc...

The REST call to the IVR provider is constructed by using the config's `outgoingCallUriTemplate` field as the base URI, substituting any [xxx] placeholders with the values in `params`. If `jsonRequest` is selected `params` are converted into JSON Object otherwise `params` are added to the HTTP request parameters.

There are three ways to have the IVR module initiate a call.

#### Initiating an outbound call via an API call

Module writers can use the `org.motechproject.ivr.service.OutboundCallService` `initiateCall` method.

#### Initiating an outbound call via a REST call

GET or POST HTTP call to: `http://{motech-server}/{motech-war-file}/module/ivr/call/{config}`

Where `{config}` is used for `configName` and the HTTP query parameters are used for `params`

---

**Note:** The default security rules for the `/call` http endpoint are `USERNAME_PASSWORD`.

---

#### Initiating an outbound call via the *tasks*

Create a task where the action is IVR - Initiate Call. Use the UI to specify the `config` and `params` parameters:

**MOTECH New Task**

Task Name: Outbound IVR call on SMS

Task Description: Initiates an outbound IVR call upon receipt of an incoming SMS. The number called is that of the sender of the incoming SMS.

Trigger » SMS : Inbound SMS

+ Add filter set + Add data source + Add action

▼ Action » IVR : Initiate Call

Available Fields: Configuration Sender Recipient Message Provider Message ID Timestamp (DATETIME) Timestamp (UTC TIME)

Channel: IVR Action: Initiate Call

Config: Voxeo

Parameters to: Sender

Help: modifications Help: custom syntax

**Save**

When everything looks good to you, click the Save & Enable button below to turn on your new Task or Save to save an in-progress Task without enabling. Click Cancel button to discard

Save & Enable Save Cancel

### 19.10.5 Settings

IVR provider Configs are created in the Settings tab. Click on **Modules / IVR / Settings**:

pivo

Name: pivo

Authentication Required? ☒

Credentials: Username Password

Outgoing call URI template

Outgoing call HTTP method: ☒ GET ☐ POST

Ignored status fields

Status field map

Injected services map: service.org.motechproject.osgi.web.UIFrameworkService

Expect JSON response? ☐

⊞ Delete

Configs consist of:

- **name:** The config name
- **authenticationRequired:** Select if the IVR provider requires authentication headers when initiating outbound calls.
- **username:** Optional username for providers that require authentication.
- **password:** Optional password for providers that require authentication.
- **outgoingCallMethod:** Which HTTP method to use, either GET or POST.
- **jsonRequest:** Select if the IVR provider requires HTTP POST method using JSON format.

- **statusFieldMap:** A map where each key corresponds to a field name coming from the IVR provider and each value corresponds to the matching IVR `CallDetailRecord` field.
- **outgoingCallUriTemplate:** A URI template where any `[xxx]` string will be replaced by the value identified by the `xxx`<sup>6</sup> key in the provided `params` map. Additionally, the entire `params` map is added as request parameters to the HTTP call.
- **ignoredStatusFields:** A list of fields to be ignored when receiving IVR Call Status from the provider. All other fields received during IVR Call Status and not mapped to CDR fields are added to the `providerExtraData`

`CallDetailRecord` map field.

- **servicesMap:** A map (in the “key1: value1, key2: value2” notation) of services that can be injected in Velocity templates where key is the name used in Velocity and value is the class of the OSGi service, for example to inject `org.motechproject.mds.service.EntityService` as `entityService`, use `entityService: org.motechproject.mds.service.EntityService`
- **jsonResponse:** Select if the provider returns JSON data after placing an outbound call.

### 19.10.6 Call Detail Records

Like configs, `CallDetailRecord` fields are viewed using the `data_services` Data Browser:

Data Browser Schema Editor Settings										
IVR Module - CallDetailRecord Instances										
<a href="#">Back to entity list</a> <a href="#">+ Add</a> <a href="#">Lookup</a> <a href="#">Fields</a> <a href="#">Export CSV</a> <a href="#">View trash</a>										
Id	Config Name	From	To	Call Direction	Call Status	Template Name	Provider Ext	Motech Call	Provider Call	Motech Time
1	kookoo	3444			UNKNOWN	helloworld	cid_type: event: NewCa provider_call_ cid_e164: circle: operator:			2014-09-2
2	kookoo	3444			UNKNOWN	helloworld	cid_type: event: NewCa provider_call_ cid_e164:			2014-09-2

Call Detail Records consist of:

- **timestamp:** The time at which the event happened, if not supplied by the provider, then supplied by the IVR module.
- **configName:** Name of the config that this CDR pertains to.
- **from:** Phone number which originated the call.
- **to:** Phone number which received the call.
- **callDirection:** INBOUND or OUTBOUND, relatively to the IVR module. Or UNKNOWN.
- **callStatus:** MOTECH\_INITIATED, INITIATED, IN\_PROGRESS, ANSWERED, BUSY, FAILED, NO\_ANSWER, or UNKNOWN.

<sup>6</sup> Note: no square brackets

- `templateName`: The name of the requested template. Only for `/template` requests.
- `motechCallId`: A Motech (ie IVR Module) generated GUID uniquely identifying a call.
- `providerCallId`: An IVR provider generated identifier, useful to query the provider (who generally has some kind of a web interface) about a specific call.
- `providerExtraData`: A map containing any additional parameter received from the IVR provider and not mapped to any of the above fields.

### 19.10.7 Custom exception

There is a custom exception class **IvrTemplateException** that can be thrown from templates. Using this exception you can control the error code in the HTTP response. In the following constructor you can provide a suitable error code:

```
public IvrTemplateException(String message, HttpStatus errorCode) {
    super(message);
    this.errorCode = errorCode;
}
```

This will allow you to return an error code that your IVR provider can understand.

## 19.11 Message Campaign Module

### Table of Contents

- Message Campaign Module
  - Introduction
  - Types of the campaigns
    - \* Campaign common fields
    - \* Absolute campaign
    - \* Offset campaign
    - \* Repeat Interval campaign
    - \* Cron campaign
    - \* Day Of Week campaign
  - Creating campaigns
    - \* Through the UI **Admin** tab
    - \* Through the UI **Settings** tab
    - \* Using configuration file
  - Editing campaigns
    - \* Editing Campaign
    - \* Editing Message
  - Enrolling a user into a campaign
    - \* Through the UI
    - \* Using CampaignRequest
  - Example
  - Querying enrollments
  - Fired events
  - Integration with the Tasks module
    - \* Triggers
    - \* Actions

### 19.11.1 Introduction

The MOTech Message Campaign module is used to enroll users into messaging campaigns. A user may be a patient, worker, lab, facility, or other desired recipient of the information to be disseminated. A campaign is a course of informational messages that are sent to the target user on an assigned date or schedule. A user can be entered into a campaign at any point during its duration via a Campaign Request. Users may be removed or re-enrolled into campaigns. The languages and formats of the campaign messages can be specified (IVR, SMS, English, etc). However when a campaign message fires, the handling of that event, including the actual messaging of the end user(IVR, SMS, etc.) is left to the implementation. Campaign messages follow a clearly defined schedule and are automatically triggered by the scheduling system.

### 19.11.2 Types of the campaigns

Each campaign has a type, which is determined by an enumeration called **CampaignType**. The CampaignType determines the type of message that will apply to the campaign. In a particular campaign, all campaign messages must be of the same campaign type. The module contains five different campaign types: absolute, offset, repeat interval, day of week, and cron. All campaigns contain a field for the campaign name and a list of campaign messages. Each type of campaign has additional fields that correspond to its type specification. Each campaign message has the fields name, formats, languages, messageKey, and startTime, along with additional fields depending on its CampaignType.

#### Campaign common fields

##### Campaign fields

- **name** The name of the campaign. Campaign requests use this parameter to associate a user with a given campaign
- **type** Specifies the type of the campaign
- **messages** An array of messages of the campaign.

##### Message fields

- **name** The name of the campaign message
- **messageType** Specifies the type of the message (used only when creating messages through UI)
- **formats** The format(s) for the message
- **languages** The language(s) for the message
- **messageKey** A key uniquely identifying this message
- **startTime** The time at which the message will be sent.

#### Absolute campaign

In this type of campaign all messages have absolute, predefined delivery dates and times. Absolute campaign messages contain the additional field: date, which indicates the date that they will be sent on at the time specified by the startTime.

### Example

```
[{
  "name" : "Absolute Dates Message Program",
  "type" : "ABSOLUTE",
  "messages" : [
    {
      "name" : "First",
      "formats" : ["IVR", "SMS"],
      "languages" : ["en"],
      "messageKey": "random-1",
      "date" : "2015-01-15",
      "startTime" : "10:30"
    },
    {
      "name" : "Second",
      "formats" : ["IVR"],
      "languages" : ["en"],
      "messageKey": "random-2",
      "startTime" : "10:30",
      "date" : "2015-01-12"
    }
  ]
}]
```

### Offset campaign

In this type of campaign all messages have delivery dates and times with defined delay. Offset campaigns include an additional field: `maxDuration`, which is optional. This field specifies the maximum duration of the campaign. Offset campaign messages contain the additional field `timeOffset`, which defines the amount of time from the reference date or current date that will elapse before the message is sent. If no reference date is supplied, then the message campaign scheduler uses the current date as the reference date. OFFSET campaigns allow clients to enroll into the sequence of messages at any point within the sequence.

### Example

```
[{
  "name" : "Relative Dates Message Program",
  "type" : "OFFSET",
  "messages" : [
    {
      "name" : "Week 1",
      "formats" : ["IVR"],
      "languages" : ["en"],
      "messageKey": "child-info-week-1",
      "timeOffset" : "1 Week",
      "startTime" : "10:30"
    },
    {
      "name" : "Week 1A",
      "formats" : ["SMS"],
      "languages" : ["en"],
      "messageKey": "child-info-week-1a",
      "timeOffset" : "1 Week",
      "startTime" : "10:30"
    }
  ]
}]
```



```
    },
    {
      "name" : "Week 1B",
      "formats" : ["SMS"],
      "languages" : ["en"],
      "messageKey": "child-info-week-1b",
      "timeOffset" : "9 Days",
      "startTime" : "10:30"
    }
  ]
}]
```

### Repeat Interval campaign

In this type of campaign all messages repeat periodically based on their interval value. Repeat interval campaigns include an additional field: `maxDuration`, which is optional and specifies the maximum duration in which the messages will be repeated. Repeating campaign messages contain the following additional message field: `repeatEvery`. The repeat interval combined with the maximum duration determines how many messages will be scheduled at the time specified by the `startTime`.

### Example

```
[{
  "name" : "Relative Parameterized Dates Message Program",
  "type" : "REPEAT_INTERVAL",
  "maxDuration" : "5 weeks",
  "messages" : [
    {
      "name" : "Weekly Message #1",
      "formats" : ["IVR", "SMS"],
      "languages" : ["en"],
      "messageKey": "child-info-week-{Offset}-1",
      "repeatEvery" : "1 Week",
      "startTime" : "10:30"
    },
    {
      "name" : "Weekly Message #2",
      "formats" : ["SMS"],
      "languages" : ["en"],
      "messageKey": "child-info-week-{Offset}-2",
      "repeatEvery" : "9 Days",
      "startTime" : "10:30"
    },
    {
      "name" : "Weekly Message #3",
      "formats" : ["SMS"],
      "languages" : ["en"],
      "messageKey": "child-info-week-{Offset}-3",
      "repeatEvery" : "12 Days",
      "startTime" : "10:30"
    }
  ]
}]
```

## Cron campaign

In this type of campaign all messages are scheduled using a cron expression. Each cron campaign message contains an additional field that identifies a cron expression. The cron expression determines the periodic schedule the messages will follow. Here you can find more information about cron expressions [Cron](#)

### Example

```
[{
  "name" : "Cron based Message Program",
  "type" : "CRON",
  "maxDuration" : "5 weeks",
  "messages" : [
    {
      "name" : "First",
      "formats" : ["IVR", "SMS"],
      "languages" : ["en"],
      "messageKey": "cron-message",
      "cron" : "0 0 12 * * ?"
    }
  ]
}]
```

## Day Of Week campaign

In this type of campaign all messages fire on the days of week(Monday, Tuesday, etc.) specified. Each day of week campaign message contains an additional field: maxDuration, which specifies the maximum duration of the campaign. Day of week campaign messages contain the additional field: repeatOn, which specifies which days of the week the message will be sent on.

### Example

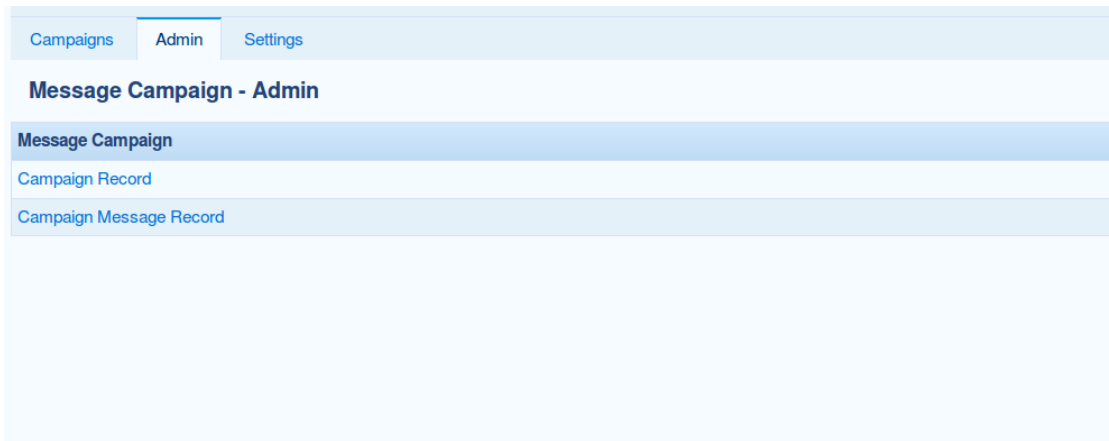
```
[{
  "name" : "PREGNANCY",
  "type" : "DAY_OF_WEEK",
  "maxDuration" : "40 Weeks",
  "messages" : [
    {
      "name" : "Pregnancy Message",
      "formats" : ["SMS"],
      "languages" : ["en"],
      "messageKey": "PREGNANCY",
      "repeatOn" : ["Monday", "Wednesday", "Friday"],
      "startTime" : "10:30"
    }
  ]
}]
```

### 19.11.3 Creating campaigns

There are three possibilities to create **Message Campaigns** :

## Through the UI Admin tab

You can create Campaigns and Messages through **Admin** tab.



## Creating Messages

When you're creating Message you need choose the message type, depending on chosen type appropriate fields will be enabled.

### Absolute Message

The screenshot shows the 'Absolute Message' form in the 'Admin' tab. The form has several fields: 'Message Type' (dropdown menu set to 'ABSOLUTE'), 'Name' (text input with placeholder 'Name of the campaign message'), 'Message Key' (text input with placeholder 'MESSAGEKEY'), 'Start Time' (text input with placeholder 'hh:mm'), 'Date' (text input with placeholder 'yyyy-mm-dd'), 'Formats' (dropdown menu set to 'Select' with a '+ Add option' button), 'Languages' (dropdown menu set to 'Select' with a '+ Add option' button), 'Campaign' (button with '+ Add related instance'), 'Owner' (dropdown menu set to 'motech'), and two buttons at the bottom: 'Zapisz' (Save) and 'Anuluj' (Cancel).

### Offset Message

Campaigns Admin Settings

Message Type \* OFFSET ▾

Name

Message Key \* ⓘ

Start Time \* ⓘ

Time Offset \* ⓘ

Formats

Languages

Campaign

Owner

### Repeat Interval Message

Campaigns Admin Settings

Message Type \* REPEAT\_INTERVAL ▾

Name

Message Key \* ⓘ

Start Time \* ⓘ

Repeat Every \* ⓘ

Formats

Languages

Campaign

Owner

## Cron Message

[Campaigns](#) [Admin](#) [Settings](#)

Message Type \*

CRON ▾

Name

Message Key \* ⓘ

Cron \* ⓘ

Every 

month

 on the 

1st

 at 

05

 : 

03

Formats

Select ▾

+ Add option

Languages

Select ▾

+ Add option

Campaign

+ Add related instance

Owner

motech ▾

Zapisz

Anuluj

## Day Of Week Message

[Campaigns](#) [Admin](#) [Settings](#)

Message Type \*

DAY\_OF\_WEEK ▾

Name

Message Key \* ⓘ

Start Time \* ⓘ

Repeat On \* ⓘ

Select ▾

Formats

Select ▾

+ Add option

Languages

Select ▾

+ Add option

Campaign

+ Add related instance

Owner

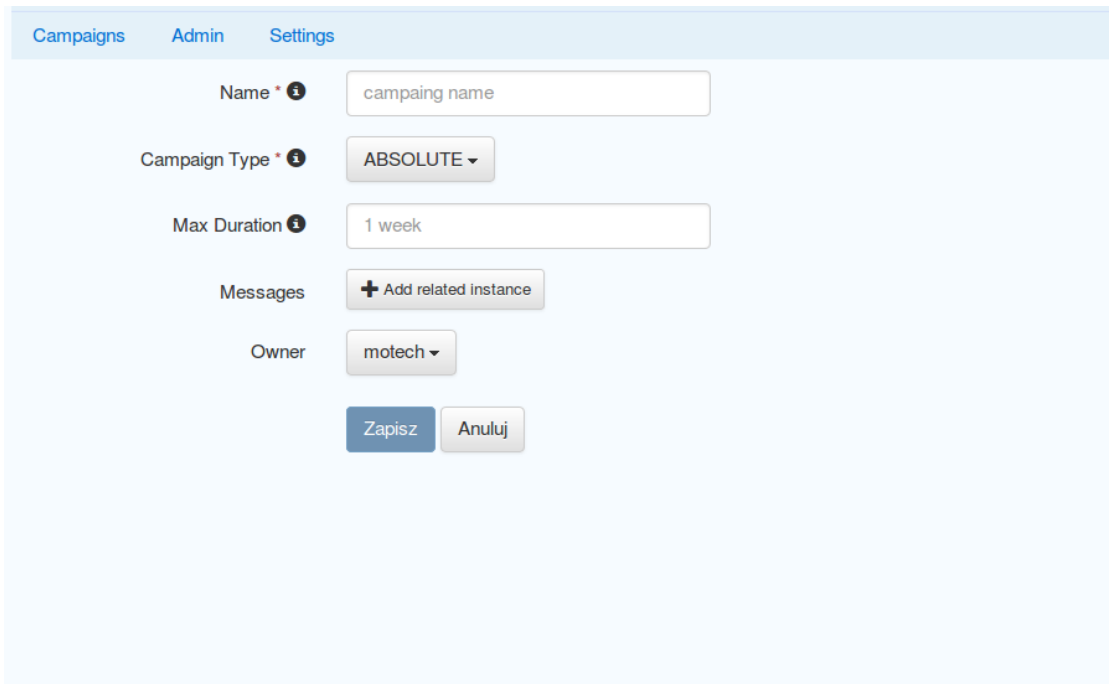
motech ▾

Zapisz

Anuluj

## Creating Campaigns

You can add Messages to the Campaign only if they are the same type as the Campaign.

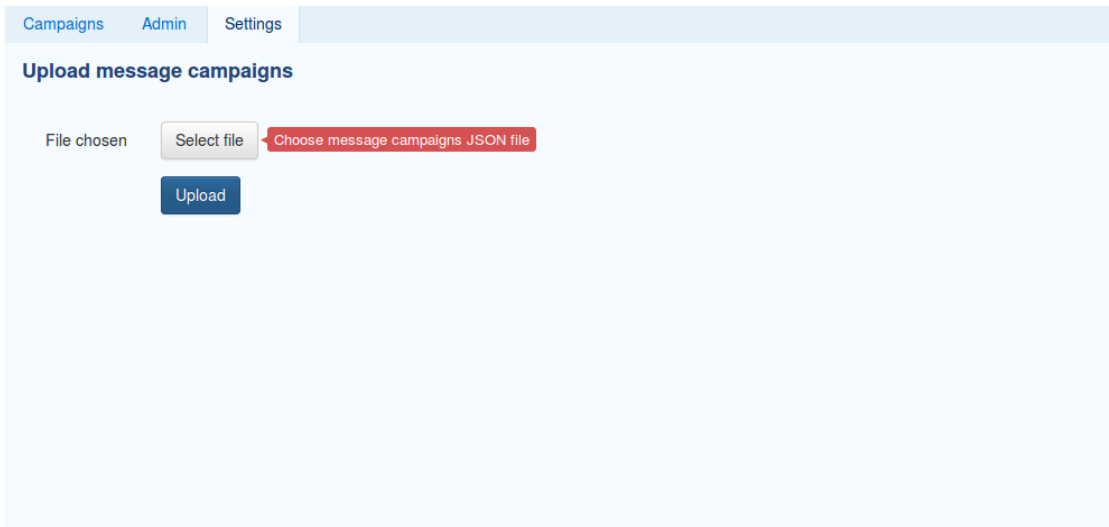


The screenshot shows a web interface for creating a campaign. At the top, there are three tabs: "Campaigns", "Admin", and "Settings", with "Settings" being the active tab. The form contains the following fields and controls:

- Name**: A text input field with the value "campaign name".
- Campaign Type**: A dropdown menu currently showing "ABSOLUTE".
- Max Duration**: A text input field with the value "1 week".
- Messages**: A button labeled "+ Add related instance".
- Owner**: A dropdown menu currently showing "motech".
- At the bottom, there are two buttons: "Zapisz" (Save) and "Anuluj" (Cancel).

## Through the UI Settings tab

You can upload JSON files entering the **Settings** tab.



The screenshot shows the "Upload message campaigns" section within the "Settings" tab. It includes the following elements:

- A header "Upload message campaigns".
- A "File chosen" label next to a "Select file" button.
- A red tooltip message that says "Choose message campaigns JSON file" pointing to the "Select file" button.
- An "Upload" button below the "Select file" button.

## Using configuration file

You can copy a file to the message-campaign directory. The name of the file has to be **message-campaigns.json**. Message campaign records will be created after MOTECH started.

### 19.11.4 Editing campaigns

Campaigns and Messages can be edited through UI.

#### Editing Campaign

When you're editing campaign you cannot change campaign name. After saving changes to the campaign all scheduled jobs will be removed and new appropriate jobs will be created. When you remove campaign all enrollments and messages belonging to that campaign will be removed as well.

The screenshot shows the 'Campaigns' admin interface. At the top, there are tabs for 'Campaigns', 'Admin', and 'Settings'. Below the tabs is a red button labeled 'Delete this instance'. The main form contains the following fields and controls:

- Name:** A text input field containing 'Absolute Dates Message Program'.
- Campaign Type:** A dropdown menu set to 'ABSOLUTE'.
- Max Duration:** A text input field containing '1 week'.
- Messages:** A section with a '+ Add related instance' button. Below it, there are two message entries:
  - 'First - random-1 - ABSOLUTE' with an 'Edit' button (gear icon) and a red minus button.
  - 'Second - random-2 - ABSOLUTE' with an 'Edit' button (gear icon) and a red minus button.
- Owner:** A dropdown menu set to 'motech'.
- Creation Date:** '2015-06-01, 14:47'
- Created By:** 'motech'
- Modification Date:** '2015-06-01, 14:47'
- Modified By:** 'motech'

At the bottom of the form are two buttons: 'Zapisz' (Save) and 'Anuluj' (Cancel).

#### Editing Message

When you're editing message you cannot change message type and key. After saving changes to the message all jobs scheduled for that message will be removed and new appropriate jobs will be created. When you remove message all jobs scheduled for that message will be removed.

Campaigns Admin Settings

Delete this instance

Message Type \*

ABSOLUTE

Name

First

Message Key \* ⓘ

random-1

Start Time \* ⓘ

10:30

Date \* ⓘ

yyyy-mm-dd

Formats

IVR, SMS ▾

+ Add option

Languages

en ▾

+ Add option

Campaign

+ Add related instance

Absolute Dates Message Program

Edit -

Owner

motech ▾

Creation Date:

2015-06-01, 14:47

Created By:

motech

Modification Date:

2015-06-01, 14:47

Modified By:

motech

### 19.11.5 Enrolling a user into a campaign

To enroll a user into a campaign you have two options :

#### Through the UI

Go to the **Campaigns** tab, choose a message campaign and add an enrollee.



The screenshot shows a web application interface for managing message campaign enrollments. At the top, there are navigation tabs for 'Campaigns', 'Admin', and 'Settings'. The main heading is 'Message Campaign - Enrollments'. Below this, a sub-header indicates the specific campaign: 'Enrollments for Campaign - Absolute Dates Messaga Program'. A table is displayed with columns for 'ID', 'Edit', and 'Delete'. Below the table, there are two buttons: 'Delete Selected' and 'Add Enrollee'.

## Using CampaignRequest

A campaign request associates a user (worker, patient, lab, etc.) with a unique campaign name, start time, and reference date. The campaign name determines which campaign will be retrieved or unscheduled from the JSON document. The reference date determines the calendar date that the campaign will begin for that user. If no reference date is supplied, then the current date upon enrollment is used in its place. The start time specifies what time of day in hours and minutes that the message will be sent. **CampaignRequest** has following fields.

Name	Type	Description
externalId	String	A client defined id to identify the enrollment
campaign-Name	String	The campaign into which the entity should be enrolled
reference-Date	Local-Date	The date the campaign has started for this enrollment. It can be in the past resulting in a delayed enrollment.
startTime	Time	Time of the day at which the alert must be raised. This overrides the campaign's deliverTime.

The message campaign module exposes the **org.motechproject.messagecampaign.service.MessageCampaignService** interface. There are few methods provided for enrolling/unenrolling a user into a campaign:

```
/**
 * Enrolls the external id into the campaign as specified in the request. The enrolled entity will be
 * against it according to the campaign definition.
 */
void enroll(CampaignRequest enrollRequest);

/**
 * Unenrolls an external from the campaign as specified in the request. The entity will no longer receive
 * the campaign.
 */
void unenroll(String externalId, String campaignName);

/**
 * Update existing campaign enrollment with data specified in the request.
 */
void updateEnrollment(CampaignRequest enrollRequest, Long enrollmentId);

/**
```

```
* Unenrolls all campaigns which match criteria provided by a CampaignEnrollmentsQuery object.
*/
void stopAll(CampaignEnrollmentsQuery query);
```

### 19.11.6 Example

Below is an example of a JSON document that includes two campaigns, each with a number of campaign messages. Each campaign object has three fields: name, type and an array of messages. The REPEAT\_INTERVAL campaign has an additional maxDuration field. Each message contains fields determined by its type. Campaign requests associate the user with these campaigns. A reference to the campaign's name, such as "Absolute Dates Message Program" must be included in the CampaignRequest. This allows the system to schedule jobs for the user based on the associated campaign. In the below example, if a user is enrolled in "Absolute Dates Message Program", two separate jobs (messages) will be scheduled. In the "Relative Parameterized Dates Message Program", twelve jobs will be scheduled for the user. The number of repeating messages is determined by the repeat intervals compared with the maximum duration. In the example below, Repeating Message #1 would have five scheduled messages due to a maximum duration of five weeks and repeat intervals of 1 week. Repeating Message #2 would have four scheduled messages due to a maximum duration of five weeks and repeat intervals of nine days.

```
[
  /* Absolute Campaign */
  {
    "name" : "Absolute Dates Message Program",
    "type" : "ABSOLUTE",
    "messages" : [
      {
        "name" : "First",
        "formats" : ["IVR", "SMS"],
        "languages" : ["en"],
        "messageKey": "random-1",
        "date" : "2013-06-15",
        "startTime" : "10:30"
      },
      {
        "name" : "Second",
        "formats" : ["IVR"],
        "languages" : ["en"],
        "messageKey": "random-2",
        "startTime" : "10:30",
        "date" : "2013-06-22"
      }
    ]
  },
  /* Repeat Interval Campaign */
  {
    "name" : "Relative Parameterized Dates Message Program",
    "type" : "REPEAT_INTERVAL",
    "maxDuration" : "5 weeks",
    "messages" : [
      {
        "name" : "Weekly Message #1",
        "formats" : ["IVR", "SMS"],
        "languages" : ["en"],
        "messageKey": "child-info-week-{Offset}-1",
        "repeatEvery" : "1 Week",
        "startTime" : "10:30"
      }
    ]
  },
]
```

```

    {
      "name" : "Weekly Message #2",
      "formats" : ["SMS"],
      "languages" : ["en"],
      "messageKey": "child-info-week-{Offset}-2",
      "repeatEvery" : "9 Days",
      "startTime" : "10:30"
    },
    {
      "name" : "Weekly Message #3",
      "formats" : ["SMS"],
      "languages" : ["en"],
      "messageKey": "child-info-week-{Offset}-3",
      "repeatEvery" : "12 Days",
      "startTime" : "10:30"
    }
  ]
}
]

```

### 19.11.7 Querying enrollments

The `org.motechproject.messagecampaign.service.MessageCampaignService` interface, in addition to its capability to enroll/unenroll all message(s), is capable of searching Campaign Enrollment Records based on particular criteria provided by a Campaign Enrollments Query object. This service's `search()` allows the client to query for enrollment records using various criteria: external ID, status, and/or campaign name. Queries are built by adding one or more of these criteria to the Campaign Enrollment Query's list of criteria. The clauses together define specific search criteria. The method returns a list of matching Campaign Enrollment Records.

Examples:

```
messageCampaignService.search(new CampaignEnrollmentsQuery().havingState(CampaignEnrollmentStatus.ACTIVE))
```

will find all active enrollments

```
messageCampaignService.search(
    new CampaignEnrollmentsQuery()
        .withCampaignName("My Program")
        .havingState(CampaignEnrollmentStatus.ACTIVE))
```

will find active enrollments enrolled into “My Program” campaign.

CampaignEnrollmentsQuery provided following methods :

```

/**
 * This provides the method for the Status Criterion using which campaign enrollments are filtered by
 */
public CampaignEnrollmentsQuery havingState(CampaignEnrollmentStatus campaignEnrollmentStatus)

/**
 * This provides the method for the ExternalId Criterion using which campaign enrollments for an ExternalId
 */
public CampaignEnrollmentsQuery withExternalId(String externalId)

/**
 * This provides the method for the CampaignName Criterion using which campaign enrollments belongs to
 * campaign are filtered
 */

```

```
public CampaignEnrollmentsQuery withCampaignName(String campaignName)

/**
 * This gives all the criterion which are present in the built query
 */
public List<Criterion> getCriteria()

/**
 * This gives the primary criterion in the built query, which is used to fetch the results from data
 */
public Criterion getPrimaryCriterion()

/**
 * This gives all the criterion other than primary criterion in the built query, which are used to f
 * of the primary criterion
 */
public List<Criterion> getSecondaryCriteria()
```

### 19.11.8 Fired events

Message Campaign module fires following events.

Subject	Info
org.motechproject.server.messagecampaign.fired-campaign-message	Fired, when a new message job of the appropriate type is scheduled

**Payload :**

- EventKeys.SCHEDULE\_JOB\_ID\_KEY (JobID),
- EventKeys.CAMPAIGN\_NAME\_KEY (CampaignName),
- EventKeys.EXTERNAL\_ID\_KEY (ExternalID),
- EventKeys.MESSAGE\_KEY (MessageKey).

Subject	Info
org.motechproject.server.messagecampaign.campaign-completed	Fired, when a campaign is completed

**Payload :**

- EventKeys.SCHEDULE\_JOB\_ID\_KEY (JobID),
- EventKeys.CAMPAIGN\_NAME\_KEY (CampaignName),
- EventKeys.EXTERNAL\_ID\_KEY (ExternalID).

Subject	Info
org.motechproject.server.messagecampaign.enrolled-user	Fired, when a new user is enrolled into a campaign

**Payload :**

- EventKeys.CAMPAIGN\_NAME\_KEY (CampaignName),
- EventKeys.EXTERNAL\_ID\_KEY (ExternalID).

Subject	Info
org.motechproject.server.messagecampaign.unenrolled-user	Fired, when a user is unenrolled from a campaign.

**Payload :**

- EventKeys.CAMPAIGN\_NAME\_KEY (CampaignName),

- EventKeys.EXTERNAL\_ID\_KEY (ExternalID).

### 19.11.9 Integration with the Tasks module

#### Triggers

For the **Message Campaign** module you can create tasks with two triggers : **Send Message** and **Campaign completed**. To do it go to the Task module, click 'New task' and you should see the Message Campaign trigger list:

The screenshot shows the 'MOTECHE New Task' form. At the top, there are tabs for 'Tasks' and 'Settings'. Below the title bar, there are input fields for 'Task Name' and 'Task Description'. A dropdown menu titled 'Available triggers' is open, showing two options: 'Send Message' and 'Campaign completed'. Below the dropdown, there is a button with an envelope icon and the text 'Message Campaign'.

#### Actions

In the Task module, you can also use Message Campaign as a channel and select an action you want :

The screenshot shows the 'Action' configuration section in the Task module. At the top, there are tabs for 'Tasks' and 'Settings'. Below the title bar, there is a section titled 'Trigger » Message Campaign : Send Message'. Below this, there are buttons for '+ Add filter set', '+ Add data source', and '+ Add action'. The 'Action' section is expanded, showing 'Available Fields: Campaign name, External ID, Message key'. Below this, there is a 'Channel:' dropdown set to 'Message Campaign' and an 'Action:' dropdown set to 'Select'. A dropdown menu is open for the 'Action:' field, showing four options: 'Send Message', 'Campaign completed', 'Enroll user', and 'Unenroll user'. At the bottom, there are buttons for 'Save & Enable', 'Save', and 'Cancel'.

## 19.12 mTraining Module

To be written.

If you'd like to author this topic, please assign yourself [MOTECH-962](#).

## 19.13 OpenMRS Module

To be written.

If you'd like to author this topic, please assign yourself [MOTECH-772](#).

## 19.14 Javadoc

### 19.14.1 org.motechproject.alerts

#### EventKeys

public final class **EventKeys**

They keys and subjects for alert fields in the event parameter map. These are used for task actions.

#### Fields

##### ALERT\_DATA

public static final [String](#) **ALERT\_DATA**

##### ALERT\_DESCRIPTION

public static final [String](#) **ALERT\_DESCRIPTION**

##### ALERT\_ID

public static final [String](#) **ALERT\_ID**

##### ALERT\_NAME

public static final [String](#) **ALERT\_NAME**

##### ALERT\_PRIORITY

public static final [String](#) **ALERT\_PRIORITY**

##### ALERT\_STATUS

public static final [String](#) **ALERT\_STATUS**

##### ALERT\_TYPE

public static final [String](#) **ALERT\_TYPE**

##### BASE\_SUBJECT

public static final [String](#) **BASE\_SUBJECT**

##### CLOSE\_ALERT\_SUBJECT

public static final [String](#) **CLOSE\_ALERT\_SUBJECT**

**CREATE\_ALERT\_SUBJECT**

public static final [String](#) **CREATE\_ALERT\_SUBJECT**

**EXTERNAL\_ID\_KEY**

public static final [String](#) **EXTERNAL\_ID\_KEY**

**MARK\_ALERT\_READ\_SUBJECT**

public static final [String](#) **MARK\_ALERT\_READ\_SUBJECT**

## 19.14.2 org.motechproject.alerts.contract

### AlertCriteria

public class **AlertCriteria**

Maintains the criteria by which search for alerts to be done

#### Constructors

**AlertCriteria**

public **AlertCriteria** ()

Instantiates alert criteria

#### Methods

**alertPriority**

public [Integer](#) **alertPriority** ()

**alertStatus**

public [AlertStatus](#) **alertStatus** ()

**alertType**

public [AlertType](#) **alertType** ()

**byDateRange**

public [AlertCriteria](#) **byDateRange** ([DateTime](#) *from*, [DateTime](#) *to*)

Adds criterion to search by the date of alert in the given range

**Parameters**

- **from** – Start date of the alerts to be searched for
- **to** – End date of the alerts to be searched for

**Returns** Instance with the current criterion added to it.

**byExternalId**

```
public AlertCriteria byExternalId (String id)
```

Adds criterion to search by external id of alert

**Parameters**

- **id** – external id of the alert to be searched for

**Returns** Instance with the current criterion added to it.

**byPriority**

```
public AlertCriteria byPriority (Integer priority)
```

Adds criterion to search by priority of alert

**Parameters**

- **priority** – priority of the alert to be searched for

**Returns** Instance with the current criterion added to it.

**byStatus**

```
public AlertCriteria byStatus (AlertStatus status)
```

Adds criterion to search by status of alert

**Parameters**

- **status** – status of the alert to be searched for

**Returns** Instance with the current criterion added to it.

**byType**

```
public AlertCriteria byType (AlertType type)
```

Adds criterion to search by type of alert

**Parameters**

- **type** – type of the alert to be searched for

**Returns** Instance with the current criterion added to it.

**externalId**

```
public String externalId ()
```

**fromDate**

```
public DateTime fromDate ()
```

**getFilters**

```
public List<Criterion> getFilters ()
```

Gets all the search criteria that have been added so far

**Returns** List of search criteria

**toDate**

```
public DateTime toDate ()
```



## AlertService

public interface **AlertService**

Interface to handle alerts.

### Methods

#### create

void **create** (*String entityId*, *String name*, *String description*, *AlertType type*, *AlertStatus status*, int *priority*,  
              *Map<String, String> data*)

Creates an alert

##### Parameters

- **entityId** – External id for alert
- **name** – Name of the alert
- **description** – Description of alert
- **type** – Alert type like critical, high, medium and low
- **status** – Status of the alert like new, read and closed
- **priority** – priority of the alert specified by integer values
- **data** – Extra information of the alert stored as property => value pairs

#### get

*Alert* **get** (*String id*)

Fetches an alert by id

##### Parameters

- **id** – Id of the alert

**Returns** Alert object with the given id if found

#### get

*Alert* **get** (*Long id*)

Fetches an alert by id

##### Parameters

- **id** – Id of the alert

**Returns** Alert object with the given id if found

#### search

*List<Alert>* **search** (*AlertCriteria alertCriteria*)

Searches for alerts by the given criteria

##### Parameters

- **alertCriteria** – Criteria which is to be used to search for alerts

**Returns** List of matched alerts for the given search criteria.

**update**

void **update** (*String alertId*, *UpdateCriteria updateCriteria*)

Updates an alert by alert id

**Parameters**

- **alertId** – Id of the alert to be updated
- **updateCriteria** – criteria which specifies the fields to be updated and their new values

**update**

void **update** (*Long alertId*, *UpdateCriteria updateCriteria*)

Updates an alert by alert id

**Parameters**

- **alertId** – Id of the alert to be updated
- **updateCriteria** – criteria which specifies the fields to be updated and their new values

**AlertsDataService**

public interface **AlertsDataService** extends *MotechDataService<Alert>*

Data Service interface for *org.motechproject.alerts.domain.Alerts*. The implementation is provided by the Motech Data Services module.

**Methods****countFindByStatus**

long **countFindByStatus** (*AlertStatus alertStatus*)

**findByAlertType**

*List<Alert>* **findByAlertType** (*AlertType alertType*)

**findByDateTime**

*List<Alert>* **findByDateTime** (*Range<DateTime> dateTimeRange*)

**findByExternalId**

*List<Alert>* **findByExternalId** (*String externalId*)

**findByPriority**

*List<Alert>* **findByPriority** (*Long priority*)

**findByStatus**

*List<Alert>* **findByStatus** (*AlertStatus alertStatus*)

## Criterion

public enum **Criterion**

An enum that represents a search criteria for alerts. The enum has two methods - `fetch` and `filter`, that will retrieve or filter the data for a criterion respectively. To be used with `org.motechproject.alerts.contract.AlertCriteria`.

### Enum Constants

#### **ALERT\_PRIORITY**

public static final [Criterion](#) **ALERT\_PRIORITY**

A criterion that will return alerts with matching **priority**.

#### **ALERT\_STATUS**

public static final [Criterion](#) **ALERT\_STATUS**

A criterion that will return alerts with matching **status**.

#### **ALERT\_TYPE**

public static final [Criterion](#) **ALERT\_TYPE**

A criterion that will return alerts with matching **type**.

#### **DATE\_RANGE**

public static final [Criterion](#) **DATE\_RANGE**

A criterion that will return alerts with their **dateTime** within the date range from the criteria.

#### **EXTERNAL\_ID**

public static final [Criterion](#) **EXTERNAL\_ID**

A criterion that will return alerts with the matching **externalId**.

## Equal

class **Equal**<T> implements [Predicate](#)

Checks if the given property of item inside a collection has the given value.

### Parameters

- **<T>** – the type of object value.

### Constructors

#### **Equal**

**Equal** (*String* *propertyName*, *T* *propertyValue*)

### Methods

#### **evaluate**

public boolean **evaluate** (*Object* *object*)

### **evaluateValue**

protected boolean **evaluateValue** (*T expected, T actual*)

## **GreaterOrEqual**

class **GreaterOrEqual**<T extends Number> extends [Equal](#)<T>

Checks if the given number property of item inside a collection has value that is greater or equal to the given number value.

### **Constructors**

#### **GreaterOrEqual**

**GreaterOrEqual** (*String propertyName, T propertyValue*)

### **Methods**

#### **evaluateValue**

protected boolean **evaluateValue** (*T expected, T actual*)

## **LessOrEqual**

class **LessOrEqual**<T extends Number> extends [Equal](#)<T>

Checks if the given number property of item inside a collection has value that is less or equal to the given number value.

### **Constructors**

#### **LessOrEqual**

**LessOrEqual** (*String propertyName, T propertyValue*)

### **Methods**

#### **evaluateValue**

protected boolean **evaluateValue** (*T expected, T actual*)

## **UpdateCriteria**

public class **UpdateCriteria**

Maintains the fields of alert to be updated and the new values for the them

### **Constructors**

#### **UpdateCriteria**

public **UpdateCriteria** ()

## Methods

### data

public [UpdateCriteria](#) **data** ([Map](#)<[String](#), [String](#)> *newData*)

Marks that data field of alert is to be updated and stores the new value

#### Parameters

- **newData** – New property => value pairs to be added / updated to the data field of alert

**Returns** Instance with the current criterion added to it.

### description

public [UpdateCriteria](#) **description** ([String](#) *newDescription*)

Marks that description field of alert is to be updated and stores the new value

#### Parameters

- **newDescription** – New value for the description field of alert

**Returns** Instance with the current criterion added to it.

### getAll

public [Map](#)<[UpdateCriterion](#), [Object](#)> **getAll** ()

Gets all the update criteria that have been added so far

**Returns** A map with criterion type as key and new value for the field as value

### name

public [UpdateCriteria](#) **name** ([String](#) *newName*)

Marks that name field of alert is to be updated and stores the new value

#### Parameters

- **newName** – New value for the name field of alert

**Returns** Instance with the current criterion added to it.

### priority

public [UpdateCriteria](#) **priority** (int *newPriority*)

Marks that priority field of alert is to be updated and stores the new value

#### Parameters

- **newPriority** – New value for the priority field of alert

**Returns** Instance with the current criterion added to it.

### status

public [UpdateCriteria](#) **status** ([AlertStatus](#) *newStatus*)

Marks that status field of alert is to be updated and stores the new value

#### Parameters

- **newStatus** – New value for the status field of alert

**Returns** Instance with the current criterion added to it.

**toString**

public **String** **toString** ()

### 19.14.3 org.motechproject.alerts.domain

#### Alert

public class **Alert** implements **Comparable**<**Alert**>

The alert class, the core of this module. Note that the alert module does not impose any interpretations on the fields of this class, that is left to the implementation that will make use of this module.

#### Constructors

##### Alert

public **Alert** ()

Constructs a new alert with medium type, new status and 0 priority.

##### Alert

public **Alert** (**String** *externalId*, **AlertType** *alertType*, **AlertStatus** *status*, int *priority*, **Map**<**String**, **String**> *data*)

Creates a new instance of the alert without the name and description.

##### Parameters

- **externalId** – the external id of this alert
- **alertType** – the type of this alert
- **status** – the status of this alert
- **priority** – the number representing the alert priority
- **data** – the additional data for this alert

##### Alert

public **Alert** (**String** *externalId*, **String** *name*, **String** *description*, **AlertType** *alertType*, **AlertStatus** *status*, int *priority*, **Map**<**String**, **String**> *data*)

Creates a new instance of an alert.

##### Parameters

- **externalId** – the external id of this alert
- **name** – the short name of this alert
- **description** – the description of this alert
- **alertType** – the type of this alert
- **status** – the status of this alert
- **priority** – the number representing the alert priority
- **data** – the additional data for this alert

## Methods

### **compareTo**

public int **compareTo** (*Alert o*)

### **equals**

public boolean **equals** (*Object o*)

### **getAlertType**

public *AlertType* **getAlertType** ()

Returns the type of this alert. The type describes the importance of this alert.

**Returns** the type of this alert

### **getData**

public *Map*<*String*, *String*> **getData** ()

Returns the additional data for this alert. This a string-string map that allows implementations to place additional data in the alert.

**Returns** the data map for this alert

### **getDateTime**

public *DateTime* **getDateTime** ()

Returns the datetime of this alert. Usually describes when the alert should be/was delivered.

**Returns** the datetime of this alert

### **getDateTimeInMillis**

public long **getDateTimeInMillis** ()

Returns the value of *getDateTime* () as milliseconds from 1970-01-01T00:00:00Z.

**Returns** the datetime of this alert in milliseconds

### **getDescription**

public *String* **getDescription** ()

**Returns** the description of this alert

### **getExternalId**

public *String* **getExternalId** ()

Gets the external id for this alert. The external id can be used for associating this alert with different external concepts, such as patients, doctors, etc.

**Returns** the external id of this alert

### **getId**

public *Long* **getId** ()

**Returns** the id of this alert (auto-generated field)

### **getName**

public **String** **getName** ()

Returns the name of this alert. Can be used for giving alerts short, human-friendly descriptions.

**Returns** the name of this alert

### **getPriority**

public **Long** **getPriority** ()

Returns the priority of this value in number format (long). This is independent from the alert type.

**Returns** the priority of this alert

### **getStatus**

public **AlertStatus** **getStatus** ()

Returns the status of this alert, describing whether it is a new alert, a read alert or a closed alert. The actual meaning of those values is left to the implementation.

**Returns** the status of this alert

### **hashCode**

public int **hashCode** ()

### **setAlertType**

public void **setAlertType** (**AlertType** *alertType*)

Sets the type of this alert. The type describes the importance of this alert.

#### **Parameters**

- **alertType** – the type of this alert

### **setData**

public void **setData** (**Map**<**String**, **String**> *data*)

Sets the additional data for this alert. This a string-string map that allows implementations to place additional data in the alert.

#### **Parameters**

- **data** – the data map for this alert

### **setDateTime**

public void **setDateTime** (**DateTime** *dateTime*)

Sets the datetime of this alert. Usually describes when the alert should be/was delivered.

#### **Parameters**

- **dateTime** – the datetime of this alert

### **setDescription**

public void **setDescription** (**String** *description*)

#### **Parameters**

- **description** – a description of this alert



**setExternalId**

```
public void setExternalId (String externalId)
```

Sets the external id for this alert. The external id can be used for associating this alert with different external concepts, such as patients, doctors, etc.

**Parameters**

- **externalId** – external id of this alert

**setId**

```
public void setId (Long id)
```

**Parameters**

- **id** – the id of this alert (auto-generated field)

**setName**

```
public void setName (String name)
```

Sets the name of this alert. Can be used for giving alerts short, human-friendly descriptions.

**Parameters**

- **name** – the name of this alert

**setPriority**

```
public void setPriority (Long priority)
```

Sets the priority of this value in number format (long). This is independent from the alert type.

**Parameters**

- **priority** – the priority of this alert

**setStatus**

```
public void setStatus (AlertStatus status)
```

Sets the status of this alert, describing whether it is a new alert, a read alert or a closed alert. The actual meaning of those values is left to the implementation.

**Parameters**

- **status** – the status of this alert

**toString**

```
public String toString ()
```

**AlertStatus**

```
public enum AlertStatus
```

Represents the status of an alert.

**Enum Constants****CLOSED**

```
public static final AlertStatus CLOSED
```

#### **NEW**

public static final [AlertStatus](#) **NEW**

#### **READ**

public static final [AlertStatus](#) **READ**

### **AlertType**

public enum **AlertType**

The type of the alert.

#### **Enum Constants**

##### **CRITICAL**

public static final [AlertType](#) **CRITICAL**

##### **HIGH**

public static final [AlertType](#) **HIGH**

##### **LOW**

public static final [AlertType](#) **LOW**

##### **MEDIUM**

public static final [AlertType](#) **MEDIUM**

### **AlertUpdater**

public enum **AlertUpdater**

An enum that represents an update action on alert. Each enum value corresponds to an alert field that will be updated using the `update (Alert, Object)` method.

#### **Enum Constants**

##### **DATA\_UPDATER**

public static final [AlertUpdater](#) **DATA\_UPDATER**

Updates the data of the alert, by putting all the new data in the map.

##### **DESCRIPTION\_UPDATER**

public static final [AlertUpdater](#) **DESCRIPTION\_UPDATER**

Updates the description of the alert.

##### **NAME\_UPDATER**

public static final [AlertUpdater](#) **NAME\_UPDATER**

Updates the name of the alert.

**PRIORITY\_UPDATER**

public static final [AlertUpdater](#) **PRIORITY\_UPDATER**

Updates the priority of the alert.

**STATUS\_UPDATER**

public static final [AlertUpdater](#) **STATUS\_UPDATER**

Updates the status of the alert.

**UpdateCriterion**

public enum **UpdateCriterion**

An update criterion that describes which field of an `org.motechproject.alerts.domain.Alert` should be updated.

**Enum Constants****DATA**

public static final [UpdateCriterion](#) **DATA**

**DESCRIPTION**

public static final [UpdateCriterion](#) **DESCRIPTION**

**NAME**

public static final [UpdateCriterion](#) **NAME**

**PRIORITY**

public static final [UpdateCriterion](#) **PRIORITY**

**STATUS**

public static final [UpdateCriterion](#) **STATUS**

## 19.14.4 org.motechproject.alerts.service

**AlertEventHandler**

public class **AlertEventHandler**

This events handler acts as an entry point for task actions for the alert module. It also allows creating, closing or marking alerts as read using MOTECH events.

**Methods****create**

public void **create** ([MotechEvent](#) *event*)

Creates a new alert from the received event.

**Parameters**

- **event** – the event to build the alert from

### setAlertService

public void **setAlertService** ([AlertService](#) *alertService*)

### updateStatus

public void **updateStatus** ([MotechEvent](#) *event*)

Closes or marks the alert as read based on the received event.

#### Parameters

- **event** – the event - id of the alert to update is the only expected parameter

## AlertFilter

public class **AlertFilter**

This class is responsible for taking an [org.motechproject.alerts.contract.AlertCriteria](#) object and returning the list of matching alerts.

### Constructors

#### AlertFilter

public **AlertFilter** ([AlertsDataService](#) *alertsDataService*)

Constructs an instance using the provided data service. The data service will be used for retrieving alerts.

#### Parameters

- **alertsDataService** – the data service for alerts

### Methods

#### search

public [List](#)<[Alert](#)> **search** ([AlertCriteria](#) *alertCriteria*)

Retrieves the list of alerts matching the provided criteria.

#### Parameters

- **alertCriteria** – the criteria for alert retrieval

**Returns** the list of matching alerts

## AlertServiceImpl

public class **AlertServiceImpl** implements [AlertService](#)

Creates and maintains alerts.

### Methods

#### create

public void **create** ([String](#) *entityId*, [String](#) *name*, [String](#) *description*, [AlertType](#) *type*, [AlertStatus](#) *status*, int *priority*, [Map](#)<[String](#), [String](#)> *data*)

Creates an alert

#### Parameters

- **entityId** – External id for alert

- **name** – Name of the alert
- **description** – Description of alert
- **type** – Alert type like critical, high, medium and low
- **status** – Status of the alert like new, read and closed
- **priority** – priority of the alert specified by integer values
- **data** – Extra information of the alert stored as property => value pairs

#### **get**

public [Alert](#) **get** ([String](#) *id*)

#### **get**

public [Alert](#) **get** ([Long](#) *id*)

Fetches an alert by id

#### **Parameters**

- **id** – Id of the alert

**Returns** Alert object with the given id if found

#### **search**

public [List](#)<[Alert](#)> **search** ([AlertCriteria](#) *alertCriteria*)

Searches for alerts by the given criteria

#### **Parameters**

- **alertCriteria** – Criteria which is to be used to search for alerts. (Cannot be null)

**Returns** List of matched alerts for the given search criteria. If no criterion is added to the alert criteria, it returns all the alerts.

#### **setAlertFilter**

public void **setAlertFilter** ([AlertFilter](#) *alertFilter*)

#### **setAlertsDataService**

public void **setAlertsDataService** ([AlertsDataService](#) *alertsDataService*)

#### **update**

public void **update** ([String](#) *alertId*, [UpdateCriteria](#) *updateCriteria*)

#### **update**

public void **update** ([Long](#) *alertId*, [UpdateCriteria](#) *updateCriteria*)

Updates an alert by alert id

#### **Parameters**

- **alertId** – Id of the alert to be updated
- **updateCriteria** – criteria which specifies the fields to be updated and their new values (Cannot be null)

### 19.14.5 org.motechproject.alerts.web

#### AlertsController

public class **AlertsController**

#### Constructors

##### AlertsController

public **AlertsController** (*AlertsDataService alertsDataService*)

#### Methods

##### closedAlerts

public *String* **closedAlerts** ()  
returns the number of closed alerts

##### newAlerts

public *String* **newAlerts** ()  
returns the number of new alerts

##### newOrReadAlerts

public *String* **newOrReadAlerts** ()  
returns the number of new or read alerts

##### readAlerts

public *String* **readAlerts** ()  
returns the number of read alerts

### 19.14.6 org.motechproject.appointments.domain

#### Appointment

public class **Appointment**  
Appointment class to track the appointments of a person identified by externalId

#### Constructors

##### Appointment

public **Appointment** ()  
Constructs an empty appointment object with its status set to - *org.motechproject.appointments.domain.Appointment*

##### Appointment

public **Appointment** (*String externalId*, *DateTime appointmentDate*, *DateTime visitedDate*, *AppointmentStatus status*, boolean *sendReminders*, *Period reminderInterval*, *DateTime reminderStartTime*)  
Constructs an appointment.

### Parameters

- **externalId** – external id of the person for whom this appointment tracks
- **appointmentDate** – due date for the appointment
- **visitedDate** – date of the fulfilled visit
- **status** – status flag to indicate if the appointment was missed
- **sendReminders** – flag to activate or deactivate sending reminders
- **reminderInterval** – the interval between reminders - a period of time
- **reminderStartTime** – time to start firing the reminder events

### Methods

#### getAppointmentDate

public [DateTime](#) **getAppointmentDate** ()

**Returns** due date for the appointment

#### getApptId

public [String](#) **getApptId** ()

**Returns** appointment id generated by the system

#### getExternalId

public [String](#) **getExternalId** ()

**Returns** external id of the person for whom this appointment tracks

#### getReminderInterval

public [Period](#) **getReminderInterval** ()

**Returns** the interval between reminders - a period of time

#### getReminderStartTime

public [DateTime](#) **getReminderStartTime** ()

**Returns** time to start firing the reminder events

#### getSendReminders

public boolean **getSendReminders** ()

**Returns** flag to activate or deactivate sending reminders

#### getStatus

public [AppointmentStatus](#) **getStatus** ()

**Returns** status flag to indicate if the appointment was missed

#### **getVisitedDate**

public **DateTime** **getVisitedDate** ()

**Returns** date of the fulfilled visit

#### **setAppointmentDate**

public void **setAppointmentDate** (**DateTime** *appointmentDate*)

##### **Parameters**

- **appointmentDate** – due date for the appointment

#### **setExternalId**

public void **setExternalId** (**String** *externalId*)

##### **Parameters**

- **externalId** – external id of the person for whom this appointment tracks

#### **setReminderInterval**

public void **setReminderInterval** (**Period** *interval*)

##### **Parameters**

- **interval** – the interval between reminders - a period of time

#### **setReminderStartTime**

public void **setReminderStartTime** (**DateTime** *startTime*)

##### **Parameters**

- **startTime** – time to start firing the reminder events

#### **setSendReminders**

public void **setSendReminders** (boolean *sendReminders*)

##### **Parameters**

- **sendReminders** – flag to activate or deactivate sending reminders

#### **setStatus**

public void **setStatus** (**AppointmentStatus** *status*)

##### **Parameters**

- **status** – status flag to indicate if the appointment was missed

#### **setVisitedDate**

public void **setVisitedDate** (**DateTime** *visitedDate*)

##### **Parameters**

- **visitedDate** – date of the fulfilled visit



## AppointmentChangeRecord

public class **AppointmentChangeRecord**

AppointmentChangeRecord class to capture the changes in the status of the appointment

### Constructors

#### AppointmentChangeRecord

public **AppointmentChangeRecord** (*String externalId, String appointmentId, DateTime appointmentDate, AppointmentStatus fromStatus, AppointmentStatus toStatus*)

Constructs a new record.

#### Parameters

- **externalId** – external Id of the patient
- **appointmentId** – appointment id of the appointment being changed
- **appointmentDate** – date of the appointment creation/change
- **fromStatus** – previous appointment status being changed
- **toStatus** – current appointment status to change to

### Methods

#### getAppointmentDate

public *DateTime* **getAppointmentDate** ()

**Returns** date of the appointment creation/change

#### getAppointmentId

public *String* **getAppointmentId** ()

**Returns** appointment id of the appointment being changed

#### getExternalId

public *String* **getExternalId** ()

**Returns** external Id of the patient

#### getFromStatus

public *AppointmentStatus* **getFromStatus** ()

**Returns** previous appointment status being changed

#### getToStatus

public *AppointmentStatus* **getToStatus** ()

**Returns** current appointment status to change to

#### **setAppointmentDate**

public void **setAppointmentDate** (*DateTime appointmentDate*)

##### **Parameters**

- **appointmentDate** – date of the appointment creation/change

#### **setAppointmentId**

public void **setAppointmentId** (*String appointmentId*)

##### **Parameters**

- **appointmentId** – appointment id of the appointment being changed

#### **setExternalId**

public void **setExternalId** (*String externalId*)

##### **Parameters**

- **externalId** – external Id of the patient

#### **setFromStatus**

public void **setFromStatus** (*AppointmentStatus fromStatus*)

##### **Parameters**

- **fromStatus** – previous appointment status being changed

#### **setToStatus**

public void **setToStatus** (*AppointmentStatus toStatus*)

##### **Parameters**

- **toStatus** – current appointment status to change to

### **AppointmentException**

public class **AppointmentException** extends *Exception*

Appointment exception class to wrap errors in the Module with custom messages

#### **Constructors**

##### **AppointmentException**

public **AppointmentException** (*String message*)

##### **AppointmentException**

public **AppointmentException** (*String message, Throwable throwable*)

### **AppointmentStatus**

public enum **AppointmentStatus**

Enum class to set the status of an appointment

## Enum Constants

### CONFIRMED

public static final [AppointmentStatus](#) **CONFIRMED**

### MISSED

public static final [AppointmentStatus](#) **MISSED**

### NONE

public static final [AppointmentStatus](#) **NONE**

### REMOVED

public static final [AppointmentStatus](#) **REMOVED**

### UNSCHEDULED

public static final [AppointmentStatus](#) **UNSCHEDULED**

### VISITED

public static final [AppointmentStatus](#) **VISITED**

## 19.14.7 org.motechproject.appointments.repository

### AppointmentChangeRecordDataService

public interface **AppointmentChangeRecordDataService** extends [MotechDataService](#)<[AppointmentChangeRecord](#)>  
Interface to utilize the MDS CRUD operations for appointment record service (logging & reporting)

#### Methods

#### findByCriteria

[List](#)<[AppointmentChangeRecord](#)> **findByCriteria** ([String](#) *externalId*, [String](#) *appointmentId*, [DateTime](#) *appointmentDate*, [AppointmentStatus](#) *fromStatus*, [AppointmentStatus](#) *toStatus*)

### AppointmentDataService

public interface **AppointmentDataService** extends [MotechDataService](#)<[Appointment](#)>  
Interface to utilize the MDS CRUD operations for appointment service

#### Methods

#### findAppointmentByExternalIdAndStatus

[List](#)<[Appointment](#)> **findAppointmentByExternalIdAndStatus** ([String](#) *externalId*, [AppointmentStatus](#) *status*)

**findAppointmentById**

`Appointment findAppointmentById (String apptId)`

**findAppointmentByStatus**

`List<Appointment> findAppointmentByStatus (AppointmentStatus status)`

**findAppointmentsByExternalId**

`List<Appointment> findAppointmentsByExternalId (String externalId)`

## 19.14.8 org.motechproject.appointments.service

### AppointmentService

public interface **AppointmentService**

AppointmentService interface to add/remove/update/find appointments

#### Methods

**addAppointment**

`Appointment addAppointment (Appointment appointment)`

Adds one instance of an appointment

**Parameters**

- **appointment** – Appointment object to add

**Returns** appointment object that is returned from the data store

**addAppointments**

`List<Appointment> addAppointments (List<Appointment> appointments)`

Add appointments for users with external id set in the appointment objects

**Parameters**

- **appointments** – list of appointment objects to add

**Returns** the newly added appointments

**findAppointmentsByExternalId**

`List<Appointment> findAppointmentsByExternalId (String externalId)`

Find the list of appointments for a given external id (overloaded)

**Parameters**

- **externalId** – external id related to the implementation

**Returns** List of appointments that belong to the external id

**findAppointmentsByExternalId**

List<Appointment> **findAppointmentsByExternalId** (String *externalId*, AppointmentStatus *status*)

Find visit for given user identifier (external id) and appointment status (overloaded)

**Parameters**

- **externalId** – external id related to the implementation
- **status** – status of the appointment to filter on

**Returns** List of appointments that belong to the external id, filtered by status

**getAppointment**

Appointment **getAppointment** (String *appointmentId*)

Get the appointment for the Appointment Id

**Parameters**

- **appointmentId** – Appointment to search for

**Returns** Appointment object for id

**removeAppointment**

void **removeAppointment** (String *appointmentId*)

Remove a given appointment from the store

**Parameters**

- **appointmentId** – Appointment to remove

**Throws**

- **AppointmentException** – if one of the provided appointments does not exist

**removeAppointments**

void **removeAppointments** (List<String> *appointmentId*)

Removes all appointments and reminders for given user (identified by externalId)

**Parameters**

- **appointmentId** – list of appointments to remove

**Throws**

- **AppointmentException** – if one of the provided ids does not match an existing appointment

**toggleReminders**

void **toggleReminders** (String *externalId*, boolean *sendReminders*)

Toggle the reminders for a user with external id. True == send reminders for patient (externalId), false == stop reminders for patient

**Parameters**

- **externalId** – External id of the user
- **sendReminders** – boolean flag to start or stop reminders based on the appointment interval field

**Throws**

- **AppointmentException** – if the appointment is not found

**updateAppointment**

[Appointment](#) **updateAppointment** ([Appointment](#) *appointment*)

Update the appointment object in the store

**Parameters**

- **appointment** – Appointment object to update

**Throws**

- **AppointmentException** – if the provided appointment does not exist

**Returns** updated Appointment object to return

**updateAppointments**

[List<Appointment>](#) **updateAppointments** ([List<Appointment>](#) *appointments*)

Updates the list of appointments

**Parameters**

- **appointments** – List of appointment objects

**Throws**

- **AppointmentException** – if one of the provided appointments does not exist

**Returns** Updated list of appointments

## 19.14.9 org.motechproject.batch.exception

### ApplicationErrors

public enum **ApplicationErrors** implements [BatchErrors](#)

Custom Exception messages for Rest services

**Author** naveen

**Enum Constants****BAD\_REQUEST**

public static final [ApplicationErrors](#) **BAD\_REQUEST**

**DATABASE\_OPERATION\_FAILED**

public static final [ApplicationErrors](#) **DATABASE\_OPERATION\_FAILED**

**DUPLICATE\_JOB**

public static final [ApplicationErrors](#) **DUPLICATE\_JOB**

**FILE\_READING\_WRITING\_FAILED**

public static final [ApplicationErrors](#) **FILE\_READING\_WRITING\_FAILED**

**JOB\_NOT\_FOUND**

public static final [ApplicationErrors](#) **JOB\_NOT\_FOUND**

**JOB\_TRIGGER\_FAILED**

public static final [ApplicationErrors](#) **JOB\_TRIGGER\_FAILED**

**UNSCHEDULE\_JOB\_FAILED**

public static final [ApplicationErrors](#) **UNSCHEDULE\_JOB\_FAILED**

**BatchError**

public class **BatchError**

JSON element for Error when there is an exception in rest call.

**Author** naveen

**Constructors****BatchError**

public **BatchError** ()

Constructs the error, setting the timestamp to now, and the hostname to the one of this instance.

**Methods****getApplication**

public [String](#) **getApplication** ()

**Returns** the name of the application that caused the error

**getErrorCode**

public [String](#) **getErrorCode** ()

**Returns** the code of this error

**getErrorMessage**

public [String](#) **getErrorMessage** ()

**Returns** a short error message

**getHostName**

public [String](#) **getHostName** ()

**Returns** the hostname of the server on which this error occurred

**getTimeStamp**

public [Date](#) **getTimeStamp** ()

**Returns** the timestamp of this error

**setApplication**

public void **setApplication** ([String](#) *application*)

**Parameters**

- **application** – the name of the application that caused the error

#### **setErrorCode**

public void **setErrorCode** (*String errorCode*)

##### **Parameters**

- **errorCode** – the code of this error

#### **setErrorMessage**

public void **setErrorMessage** (*String errorMessage*)

##### **Parameters**

- **errorMessage** – a short error message

#### **setHostName**

public void **setHostName** (*String hostName*)

##### **Parameters**

- **hostName** – the hostname of the server on which this error occurred

#### **setTimeStamp**

public void **setTimeStamp** (*Date timeStamp*)

##### **Parameters**

- **timeStamp** – the timestamp of this error

### **BatchErrors**

public interface **BatchErrors**

Interface for custom errors

**Author** naveen

### **Methods**

#### **getCode**

int **getCode** ()

**Returns** the code associated with this error

#### **getHttpStatus**

*HttpStatus* **getHttpStatus** ()

**Returns** the HTTP status that should be used for representing this error

#### **getMessage**

*String* **getMessage** ()

**Returns** a short description of the error



## BatchException

public class **BatchException** extends [Exception](#)  
Custom exception for the batch module.

**Author** naveen

### Constructors

#### BatchException

public **BatchException** ([BatchErrors](#) *batchErrors*)

##### Parameters

- **batchErrors** – the error that caused the exception

#### BatchException

public **BatchException** ([BatchErrors](#) *batchErrors*, [String](#) *reason*)

##### Parameters

- **batchErrors** – the error that caused the exception
- **reason** – reason for the error

#### BatchException

public **BatchException** ([BatchErrors](#) *batchErrors*, [Throwable](#) *throwable*)

##### Parameters

- **batchErrors** – the error that caused the exception
- **throwable** – the underlying exception

#### BatchException

public **BatchException** ([BatchErrors](#) *batchErrors*, [Throwable](#) *throwable*, [String](#) *reason*)

##### Parameters

- **batchErrors** – the error that caused the exception
- **throwable** – the underlying exception
- **reason** – the reason for this error

### Methods

#### getError

public [BatchErrors](#) **getError** ()

**Returns** the underlying error

#### getErrorCode

public int **getErrorCode** ()

Return the error code for the underlying error.

**Returns** the error code

**getErrorMessage**

```
public String getErrorMessage ()
```

Returns a short description of the error.

**Returns** the message representing this error

**getErrorMessageDetails**

```
public String getErrorMessageDetails ()
```

**Returns** the stacktrace for this exception as a string

**getReason**

```
public String getReason ()
```

**Returns** the reason of this error

**setReason**

```
public void setReason (String reason)
```

**Parameters**

- **reason** – the reason of this error

**RestException**

```
public class RestException extends RuntimeException
```

Exception thrown for Http Requests in case of any exception.

**Author** naveen

**Constructors****RestException**

```
public RestException (BatchException exception, List<String> errors)
```

Builds this exception from the provided batch exception and list of errors.

**Parameters**

- **exception** – the underlying exception
- **errors** – list of errors that will be treated as the reason for this exception

**RestException**

```
public RestException (BatchException batchException, String reason)
```

Builds this exception given an exception and a reason for it.

**Parameters**

- **batchException** – the underlying exception
- **reason** – the reason of the error

## Methods

### getBatchException

public [BatchException](#) **getBatchException** ()

**Returns** the underlying batch exception

### getHttpStatus

public [HttpStatus](#) **getHttpStatus** ()

**Returns** the HTTP status that should be returned for this exception

### getReason

public [String](#) **getReason** ()

**Returns** the reason of the error

## 19.14.10 org.motechproject.batch.mds

### BatchJob

public class **BatchJob** extends [MdsEntity](#) implements [java.io.Serializable](#)

Class containing all the fields related to a batch job. This is a domain object persisted using MDS.

**Author** naveen

## Constructors

### BatchJob

public **BatchJob** ()

Constructs an instance without setting any fields.

### BatchJob

public **BatchJob** ([Integer](#) *batchJobStatusId*, [String](#) *jobName*)

Constructs a job using the given id and job name.

#### Parameters

- **batchJobStatusId** – the status ID for the batch job
- **jobName** – the name of the job

### BatchJob

public **BatchJob** ([Integer](#) *batchJobStatusId*, [String](#) *jobName*, [String](#) *cronExpression*)

Constructs a job using the given id, job name and cron expression.

#### Parameters

- **batchJobStatusId** – the status ID for the batch job
- **jobName** – the name of the job
- **cronExpression** – the cron expression behind this batch job

## Methods

### **getBatchJobStatusId**

public [Integer](#) **getBatchJobStatusId** ()

**Returns** the status ID for the batch job

### **getCronExpression**

public [String](#) **getCronExpression** ()

**Returns** the cron expression behind this batch job

### **getJobContent**

public [Byte](#)[] **getJobContent** ()

Returns the content for the job. This will return a copy of the array from this entity.

**Returns** the content for this job, serialized to bytes

### **getJobName**

public [String](#) **getJobName** ()

**Returns** the name of the job

### **setBatchJobStatusId**

public void **setBatchJobStatusId** ([Integer](#) *batchJobStatusId*)

#### **Parameters**

- **batchJobStatusId** – the status ID for the batch job

### **setCronExpression**

public void **setCronExpression** ([String](#) *cronExpression*)

#### **Parameters**

- **cronExpression** – the cron expression behind this batch job

### **setJobContent**

public void **setJobContent** ([Byte](#)[] *jobContent*)

Sets the content for the job. This will copy the provided content array.

#### **Parameters**

- **jobContent** – the content for this job, serialized to bytes

### **setJobName**

public void **setJobName** ([String](#) *jobName*)

#### **Parameters**

- **jobName** – the name of the job

## BatchJobParameters

public class **BatchJobParameters** implements `java.io.Serializable`  
Class containing parameters as fields to run the job.

### Constructors

#### BatchJobParameters

public **BatchJobParameters** ()  
Constructs an instance without setting any fields.

#### BatchJobParameters

public **BatchJobParameters** (*Integer batchJobId*, *String parameterName*)  
Constructs a parameter without a value.

##### Parameters

- **batchJobId** – the id of the batch job
- **parameterName** – the name of this parameter

#### BatchJobParameters

public **BatchJobParameters** (*Integer batchJobId*, *String parameterName*, *String parameterValue*)

##### Parameters

- **batchJobId** – the id of the batch job
- **parameterName** – the name of this parameter
- **parameterValue** – the value of the parameter

### Methods

#### getBatchJobId

public *Integer* **getBatchJobId** ()  
**Returns** the id of the batch job for this parameter

#### getParameterName

public *String* **getParameterName** ()  
**Returns** the name of this parameter

#### getParameterValue

public *String* **getParameterValue** ()  
**Returns** the value of this parameter

#### setBatchJobId

public void **setBatchJobId** (*Integer batchJobId*)

##### Parameters

- **batchJobId** – the id of the batch job for this parameter

**setParameterName**

public void **setParameterName** (*String parameterName*)

**Parameters**

- **parameterName** – the name of this parameter

**setParameterValue**

public void **setParameterValue** (*String parameterValue*)

**Parameters**

- **parameterValue** – the value of this parameter

## 19.14.11 org.motechproject.batch.mds.service

**BatchJobMDSService**

public interface **BatchJobMDSService** extends MotechDataService<*BatchJob*>

Class to query the *BatchJob* entity. Implementation will be generated at runtime by MDS.

**Methods****findByJobName**

*BatchJob* **findByJobName** (*String jobName*)

Finds a batch job by name.

**Parameters**

- **jobName** – the name to query for

**Returns** the found job, or null if no found

**BatchJobParameterMDSService**

public interface **BatchJobParameterMDSService** extends MotechDataService<*BatchJobParameters*>

Class to query *BatchJobParameters* entity. The implementation will be generated at runtime by MDS.

**Author** naveen

**Methods****findByJobId**

List<*BatchJobParameters*> **findByJobId** (*Integer batchJobId*)

Finds batch job parameters using the given job ID.

**Parameters**

- **batchJobId** – the id of the batch job

**Returns** list of parameter entities for the batch job

## 19.14.12 org.motechproject.batch.model

### BatchJobDTO

public class **BatchJobDTO**

Class containing fields of Batch Job which is to be sent as response through REST.

**Author** Naveen

### Methods

#### getCreateTime

public **DateTime** **getCreateTime** ()

**Returns** the date time when this job was created

#### getCreatedBy

public **String** **getCreatedBy** ()

**Returns** the username of the user who created this job

#### getCronExpression

public **String** **getCronExpression** ()

**Returns** the cron expression behind this batch job

#### getJobId

public long **getJobId** ()

**Returns** the id of the batch job

#### getJobName

public **String** **getJobName** ()

**Returns** the name of the batch job

#### getLastUpdated

public **DateTime** **getLastUpdated** ()

**Returns** the last modification date for this job

#### getLastUpdatedBy

public **String** **getLastUpdatedBy** ()

**Returns** the username of the user who last updated this job

#### getParameters

public **Map**<**String**, **String**> **getParameters** ()

**Returns** a map of parameters for this batch job

#### **getStatus**

public **String** **getStatus** ()

**Returns** the status of the batch job

#### **setCreateTime**

public void **setCreateTime** (**DateTime** *createTime*)

##### **Parameters**

- **createTime** – the date time when this job was created

#### **setCreatedBy**

public void **setCreatedBy** (**String** *createdBy*)

##### **Parameters**

- **createdBy** – the username of the user who created this job

#### **setCronExpression**

public void **setCronExpression** (**String** *cronExpression*)

##### **Parameters**

- **cronExpression** – the cron expression behind this batch job

#### **setJobId**

public void **setJobId** (long *jobId*)

##### **Parameters**

- **jobId** – the id of the batch job

#### **setJobName**

public void **setJobName** (**String** *jobName*)

##### **Parameters**

- **jobName** – the name of the batch job

#### **setLastUpdated**

public void **setLastUpdated** (**DateTime** *lastUpdated*)

##### **Parameters**

- **lastUpdated** – the last modification date for this job

#### **setLastUpdatedBy**

public void **setLastUpdatedBy** (**String** *lastUpdatedBy*)

##### **Parameters**

- **lastUpdatedBy** – the username of the user who last updated this job



**setParameters**

public void **setParameters** ([Map<String, String>](#) *parameters*)

**Parameters**

- **parameters** – a map of parameters for this batch job

**setStatus**

public void **setStatus** ([String](#) *status*)

**Parameters**

- **status** – the status of the batch job

**BatchJobListDTO**

public class **BatchJobListDTO**

Class containing list of [BatchJobDTO](#) to be sent across in a REST response.

**Author** Naveen

**Methods****getBatchJobDtoList**

public [List<BatchJobDTO>](#) **getBatchJobDtoList** ()

**Returns** the list of jobs

**setBatchJobDtoList**

public void **setBatchJobDtoList** ([List<BatchJobDTO>](#) *batchJobDtoList*)

**Parameters**

- **batchJobDtoList** – the list of jobs

**BatchJobUpdateParams**

public class **BatchJobUpdateParams**

A request for updating params of a batch job.

**Methods****getJobName**

public [String](#) **getJobName** ()

**Returns** the name of the job to update

**getParamsMap**

public [Map<String, String>](#) **getParamsMap** ()

**Returns** the map representing the params to update

#### **setJobName**

public void **setJobName** (*String jobName*)

##### **Parameters**

- **jobName** – the name of the job to update

#### **setParamsMap**

public void **setParamsMap** (*Map<String, String> paramsMap*)

##### **Parameters**

- **paramsMap** – the map representing the params to update

### **CronJobScheduleParam**

public class **CronJobScheduleParam**

A representation of a cron job to be scheduled.

#### **Methods**

##### **getCronExpression**

public *String* **getCronExpression** ()

**Returns** the cron expression for this job

##### **getJobName**

public *String* **getJobName** ()

**Returns** the name of the job

##### **getParamsMap**

public *Map<String, String>* **getParamsMap** ()

**Returns** a map containing parameters for this job

##### **setCronExpression**

public void **setCronExpression** (*String cronExpression*)

##### **Parameters**

- **cronExpression** – the cron expression for this job

##### **setJobName**

public void **setJobName** (*String jobName*)

##### **Parameters**

- **jobName** – the name of the job

**setParamsMap**

public void **setParamsMap** ([Map](#)<[String](#), [String](#)> *paramsMap*)

**Parameters**

- **paramsMap** – a map containing parameters for this job

**JobExecutionHistoryListDTO**

public class **JobExecutionHistoryListDTO**

A representation of batch job executions that took place.

**Methods****getJobExecutionHistoryList**

public [List](#)<[JobExecution](#)> **getJobExecutionHistoryList** ()

**Returns** a list containing job execution history

**setJobExecutionHistoryList**

public void **setJobExecutionHistoryList** ([List](#)<[JobExecution](#)> *jobExecutionHistoryList*)

**Parameters**

- **jobExecutionHistoryList** – a list containing job execution history

**JobStatusLookup**

public enum **JobStatusLookup**

A representation of batch job statuses.

**Enum Constants****ACTIVE**

public static final [JobStatusLookup](#) **ACTIVE**

**INACTIVE**

public static final [JobStatusLookup](#) **INACTIVE**

**OneTimeJobScheduleParams**

public class **OneTimeJobScheduleParams** extends [BatchJobUpdateParams](#)

A representation of a one time job to be scheduled.

**Methods****getDate**

public [String](#) **getDate** ()

**Returns** the date to run this job on

**setDate**

public void **setDate** (*String date*)

**Parameters**

- **date** – the date run this job on

### 19.14.13 org.motechproject.batch.service

**JobService**

public interface **JobService**

Interface to schedule reschedule jobs or update job parameters

**Author** Naveen

**Methods****countJobs**

long **countJobs** ()

Returns the total count of batch jobs.

**Returns** the number of batch jobs

**getListOfJobs**

*BatchJobListDTO* **getListOfJobs** ()

Get the list of scheduled jobs/

**Throws**

- **BatchException** – if unable to retrieve the job list

**Returns** *org.motechproject.batch.mds.BatchJob* which contains list of  
*org.motechproject.batch.model.BatchJobDTO*(contains fields from  
*org.motechproject.batch.mds.BatchJob*)

**rescheduleJob**

void **rescheduleJob** (*String jobName*, *String cronExpression*)

Re-schedule an existing batch job with given job name and cron expression.

**Parameters**

- **jobName** – job name for the job to be scheduled
- **cronExpression** – cron expression for the job (specified for the timely run of the job)

**scheduleJob**

void **scheduleJob** (*CronJobScheduleParam params*)

Schedule a new cron job with given job name and cron expression.

**Parameters**

- **params** –
  - *CronJobScheduleParam* object containing jobName, paramsMap and cronExpression

**Throws**

- **BatchException** – if unable to schedule the job

**scheduleOneTimeJob**

void **scheduleOneTimeJob** (*OneTimeJobScheduleParams params*)

Schedule a one time job, to be run once in the future.

**Parameters**

- **params** –
  - *OneTimeJobScheduleParams* object containing jobName, paramsMap and date

**Throws**

- **BatchException** – if unable to schedule the job

**unscheduleJob**

void **unscheduleJob** (*String jobName*)

Unschedule an existing batch job with given job name.

**Parameters**

- **jobName** – job name for the job to be scheduled

**Throws**

- **BatchException** – if unable to unschedule the job

**updateJobProperty**

void **updateJobProperty** (*String jobName, Map<String, String> paramsMap*)

Update the job parameters of the scheduled job.

**Parameters**

- **jobName** – job name for the job for which parameters to be updated
- **paramsMap** – list of parameters to be added or changed

**Throws**

- **BatchException** – if unable to update the job

**JobTriggerService**

public interface **JobTriggerService**

Interface to perform the trigger operation for all types of jobs.

**Author** Naveen

**Methods****getJobExecutionHistory**

*JobExecutionHistoryListDTO* **getJobExecutionHistory** (*String jobName*)

Returns the execution history for a job.

**Parameters**

- **jobName** – name of the job to query for history

**Throws**

- **BatchException** – if history retrieval failed

**Returns** the history of executions for this job

**restart**

long **restart** (*String jobName*, *Integer executionId*)

Restarts the execution of a batch job.

**Parameters**

- **jobName** – the name of batch job
- **executionId** – the id of the execution to restart

**Throws**

- **BatchException** – if restarting the job failed

**Returns** the id of the execution

**triggerJob**

long **triggerJob** (*String jobName*)

Trigger a job instantly.

**Parameters**

- **jobName** – name of the job to be triggered

**Throws**

- **BatchException** – if triggering the job failed

**Returns** the id of the execution

## 19.14.14 org.motechproject.cmslite.model

### CMSLiteException

public class **CMSLiteException** extends *Exception*

Wrapper for any exception during adding or retrieving content.

**Constructors****CMSLiteException**

public **CMSLiteException** (*String message*)

**CMSLiteException**

public **CMSLiteException** (*String message*, *Throwable e*)

## Content

public interface **Content**

Abstract representation of CMS Lite content. Identified by name and language.

## Methods

**getLanguage**

`String getLanguage ()`

Returns the language of this content. Each content should have a language associated with it.

**Returns** the language of this content

**getMetadata**

`Map<String, String> getMetadata ()`

Returns the metadata for this content. The metadata is a string-string map.

**Returns** the metadata for this content

**getName**

`String getName ()`

Returns the name identifying this content.

**Returns** the name of this content

## ContentNotFoundException

public class **ContentNotFoundException** extends `Exception`

Thrown when content is not available.

## Constructors

**ContentNotFoundException**

public **ContentNotFoundException** ()

## StreamContent

public class **StreamContent** implements `Content`

Represents a stream content along with checksum. Any file can be stored as a stream content in CMS-Lite.

## Constructors

**StreamContent**

public **StreamContent** ()

### StreamContent

public **StreamContent** (*String language*, *String name*, *Byte[] content*, *String checksum*, *String contentType*)

#### Parameters

- **language** – the language of the content
- **name** – the name identifying this content
- **content** – the actual can as an array of bytes
- **checksum** – the MD5 checksum for the content
- **contentType** – the mime type of the content

#### Methods

##### equals

public boolean **equals** (*Object obj*)

##### getChecksum

public *String* **getChecksum** ()

**Returns** the MD5 checksum for the content

##### getContent

public *Byte[]* **getContent** ()

**Returns** the actual can as an array of bytes

##### getContentType

public *String* **getContentType** ()

**Returns** the mime type of the content

##### getLanguage

public *String* **getLanguage** ()

**Returns** the language of the content

##### getMetadata

public *Map*<*String*, *String*> **getMetadata** ()

**Returns** the additional metadata for the content

##### getName

public *String* **getName** ()

**Returns** the name identifying this content

##### hashCode

public int **hashCode** ()



**setChecksum**

public void **setChecksum** (*String checksum*)

**Parameters**

- **checksum** – the MD5 checksum for the content

**setContent**

public void **setContent** (*Byte[] content*)

**Parameters**

- **content** – the actual can as an array of bytes

**setContentType**

public void **setContentType** (*String contentType*)

**Parameters**

- **contentType** – the mime type of the content

**setLanguage**

public void **setLanguage** (*String language*)

**Parameters**

- **language** – the language of the content

**setMetadata**

public void **setMetadata** (*Map<String, String> metadata*)

**Parameters**

- **metadata** – the additional metadata for the content

**setName**

public void **setName** (*String name*)

**Parameters**

- **name** – the name identifying this content

**toString**

public *String* **toString** ()

**StringContent**

public class **StringContent** implements *Content*

Represents text content. This class is more suited for storing text content than *org.motechproject.cmslite.model.StreamContent*.

**Constructors****StringContent**

public **StringContent** ()

### StringContent

public **StringContent** (*String language*, *String name*, *String value*)

#### Parameters

- **language** – the language for the content
- **name** – the name identifying the content
- **value** – the content text

### StringContent

public **StringContent** (*String language*, *String name*, *String value*, *Map<String, String> metadata*)

#### Parameters

- **language** – the language for the content
- **name** – the name identifying the content
- **value** – the content text
- **metadata** – the additional metadata for the content

### Methods

#### equals

public boolean **equals** (*Object obj*)

#### getLanguage

public *String* **getLanguage** ()

**Returns** the language for the content

#### getMetadata

public *Map<String, String>* **getMetadata** ()

**Returns** the additional metadata for the content

#### getName

public *String* **getName** ()

**Returns** the name identifying the content

#### getValue

public *String* **getValue** ()

**Returns** the content text

#### hashCode

public int **hashCode** ()

**setLanguage**

public void **setLanguage** (*String language*)

**Parameters**

- **language** – the language for the content

**setMetadata**

public void **setMetadata** (*Map<String, String> metadata*)

**Parameters**

- **metadata** – the additional metadata for the content

**setName**

public void **setName** (*String name*)

**Parameters**

- **name** – the name identifying the content

**setValue**

public void **setValue** (*String value*)

**Parameters**

- **value** – the content text

**toString**

public *String* **toString** ()

## 19.14.15 org.motechproject.cmslite.service

### CMSLiteService

public interface **CMSLiteService**

CMS Lite is a lightweight content management system based on MDS storage. It supports storing and retrieving of stream / text content along with custom metadata for each language.

#### Methods

**addContent**

void **addContent** (*Content content*)

Add content to the CMS data-store.

**Parameters**

- **content** – the content to add

**See also:** `org.motechproject.cmslite.model.StreamContent`,  
`org.motechproject.cmslite.model.StringContent`

**getAllContents****List<Content>** **getAllContents** ()

Retrieves a list of all String and Stream contents. Returned list is defined as a list of generic Content instances, containing basic information such as language, name and metadata. To make better use of instances retrieved this way an instanceof check should be performed, followed by a cast to either StringContent or StreamContent

**Returns** list of all contents

**See also:** `org.motechproject.cmslite.model.StringContent`,  
`org.motechproject.cmslite.model.StreamContent`

**getStreamContent****StreamContent** **getStreamContent** (*String language*, *String name*)

Get stream content with the given name and language.

**Parameters**

- **language** – the language of the content
- **name** – the name of the content

**Throws**

- **ContentNotFoundException** – if no such content exists

**Returns** StreamContent that matches the parameters

**getStreamContent****StreamContent** **getStreamContent** (long *streamContentId*)

Retrieved StreamContent by its ID.

**Parameters**

- **streamContentId** – ID of a StreamContent to retrieved

**Returns** StreamContent of a given ID

**See also:** `org.motechproject.cmslite.model.StreamContent`

**getStringContent****StringContent** **getStringContent** (*String language*, *String name*)

Get text content for given name and language.

**Parameters**

- **language** – the language of the content
- **name** – the name of the content

**Throws**

- **ContentNotFoundException** – if no such content exists

**Returns** StringContent that matches the parameters

**getStringContent**

`StringContent` **getStringContent** (long *stringContentId*)

Retrieved `StringContent` by its ID.

**Parameters**

- **stringContentId** – ID of a `StringContent` to retrieved

**Returns** `StringContent` of a given ID

**See also:** `org.motechproject.cmslite.model.StringContent`

**isStreamContentAvailable**

boolean **isStreamContentAvailable** (`String` *language*, `String` *name*)

Check if a stream content with the given name and language exists.

**Parameters**

- **language** – the language of the content
- **name** – the name of the content

**Returns** true if a matching stream content exists, false otherwise

**isStringContentAvailable**

boolean **isStringContentAvailable** (`String` *language*, `String` *name*)

Check if a text content with the given name and language exists.

**Parameters**

- **language** – the language of the content
- **name** – the name of the content

**Returns** true if a matching text content exists, false otherwise

**removeStreamContent**

void **removeStreamContent** (`String` *language*, `String` *name*)

Remove stream content for given name and language.

**Parameters**

- **language** – the language of the content to remove
- **name** – the name of the content to remove

**Throws**

- **ContentNotFoundException** – if no such content exists

**removeStringContent**

void **removeStringContent** (`String` *language*, `String` *name*)

Remove text content for given name and language.

**Parameters**

- **language** – the language of the content to remove
- **name** – the name of the content to remove

**Throws**

- **ContentNotFoundException** – if no such content exists

#### retrieveStreamContentData

Byte[] **retrieveStreamContentData** (*StreamContent instance*)

Helper method, retrieving data of a StreamContent. The data is returned as a byte array.

##### Parameters

- **instance** – Instance of a StreamContent

**Returns** byte array, containing data of a StreamContent

### CMSLiteServiceImpl

public class **CMSLiteServiceImpl** implements **CMSLiteService**

The implementation of `org.motechproject.cmslite.service.CMSLiteService`. It uses MDS for storing content in the database.

#### Methods

##### addContent

public void **addContent** (*Content content*)

##### getAllContents

public *List<Content>* **getAllContents** ()

##### getStreamContent

public *StreamContent* **getStreamContent** (*String language, String name*)

##### getStreamContent

public *StreamContent* **getStreamContent** (long *streamContentId*)

##### getStringContent

public *StringContent* **getStringContent** (*String language, String name*)

##### getStringContent

public *StringContent* **getStringContent** (long *stringContentId*)

##### isStreamContentAvailable

public boolean **isStreamContentAvailable** (*String language, String name*)

##### isStringContentAvailable

public boolean **isStringContentAvailable** (*String language, String name*)

##### removeStreamContent

public void **removeStreamContent** (*String language, String name*)

**removeStringContent**

public void **removeStringContent** (*String language, String name*)

**retrieveStreamContentData**

public *Byte[]* **retrieveStreamContentData** (*StreamContent instance*)

**setStreamContentService**

public void **setStreamContentService** (*StreamContentService streamContentService*)

**setStringContentService**

public void **setStringContentService** (*StringContentService stringContentService*)

**StreamContentService**

public interface **StreamContentService** extends MotechDataService<*StreamContent*>

Data Service interface for *StreamContents*. The implementation is provided by the Motech Data Services module.

**Methods****findByLanguageAndName**

*StreamContent* **findByLanguageAndName** (*String language, String name*)

Finds content by language and name.

**Parameters**

- **language** – the language of the content
- **name** – the name of content

**Returns** the matching content or null if it doesn't exist

**StringContentService**

public interface **StringContentService** extends MotechDataService<*StringContent*>

Data Service interface for *StringContents*. The implementation is provided by the Motech Data Services module.

**Methods****findByLanguageAndName**

*StringContent* **findByLanguageAndName** (*String language, String name*)

Finds content by language and name.

**Parameters**

- **language** – the language of the content
- **name** – the name of content

**Returns** the matching content or null if doesn't exist

## 19.14.16 org.motechproject.cmslite.web

### GridSettings

public class **GridSettings**

The grid settings coming from the jqgrid plugin.

### Fields

#### DEFAULT\_ROWS

public static final int **DEFAULT\_ROWS**

The default number of rows displayed in the grid.

### Methods

#### containsLanguage

public boolean **containsLanguage** (*String language*)

Checks whether the given language matches these grid settings and should be displayed.

#### Parameters

- **language** – the language to check

**Returns** true if it matches the settings, false otherwise

#### getLanguages

public *String* **getLanguages** ()

**Returns** the languages for which we should filter contents

#### getName

public *String* **getName** ()

**Returns** the name for which we should filter contents

#### getPage

public *Integer* **getPage** ()

**Returns** the page to display, starting from 1

#### getRows

public *Integer* **getRows** ()

**Returns** the number of rows to display per page

#### getSidx

public *String* **getSidx** ()

**Returns** the name of the field on which sorting should be performed



**getSord**

```
public String getSord ()
```

**Returns** the sorting direction for these settings, either asc or desc

**isCorrect**

```
public boolean isCorrect (String contentName, String contentLanguage, String contentType)
```

Checks whether given parameters from a content match these grid settings.

**Parameters**

- **contentName** – the name of the content
- **contentLanguage** – the language of the content
- **contentType** – the type of the content (stream or string)

**Returns** true if the content matches these grid settings, false otherwise

**isDescending**

```
public boolean isDescending ()
```

Checks whether the sorting direction for these settings is descending (equal to “desc”).

**Returns** true if ordering direction is descending, false otherwise

**isStream**

```
public Boolean isStream ()
```

**Returns** true if these settings accept (display) stream contents, false otherwise

**isString**

```
public Boolean isString ()
```

**Returns** true if these settings accept (display) string contents, false otherwise

**setLanguages**

```
public void setLanguages (String languages)
```

Sets the languages for which we should filter. Blank languages will be ignored.

**Parameters**

- **languages** – the languages for which we should filter contents

**setName**

```
public void setName (String name)
```

Sets the name for which we should filter contents. Blank name will be ignored.

**Parameters**

- **name** – the name for which we should filter contents

### setPage

public void **setPage** (*Integer page*)

Sets the page to display. Null will be ignored.

#### Parameters

- **page** – the page to display, starting from 1

### setRows

public void **setRows** (*Integer rows*)

Sets the number of rows to display per page. Null will be ignored.

#### Parameters

- **rows** – the number of rows to display per page

### setSidx

public void **setSidx** (*String sidx*)

Sets the name of the field on which sorting should be performed. Blank column name will be ignored.

#### Parameters

- **sidx** – the name of the field on which sorting should be performed

### setSord

public void **setSord** (*String sord*)

Sets the sorting direction for these settings. Blank values will be ignored.

#### Parameters

- **sord** – the sorting direction for these settings, either asc or desc

### setStream

public void **setStream** (*Boolean stream*)

Set whether these settings should accept (display) stream contents. Null will be ignored.

#### Parameters

- **stream** – true if these settings accept (display) stream contents, false otherwise

### setString

public void **setString** (*Boolean string*)

Set whether these settings should accept (display) string contents. Null will be ignored.

#### Parameters

- **string** – true if these settings accept (display) string contents, false otherwise

## ResourceComparator

public class **ResourceComparator** implements *Comparator<ResourceDto>*, *Serializable*

The comparator used for ordering content for the UI grid. It can compare content by either using the content name or content type.

## Constructors

### ResourceComparator

public **ResourceComparator** (*GridSettings settings*)

Constructs an instance using taking the sorting settings from the provided grid settings.

#### Parameters

- **settings** – grid settings based on which this comparator will be created

## Methods

### compare

public int **compare** (*ResourceDto first, ResourceDto second*)

### ResourceController

public class **ResourceController**

The controller responsible for serving CMS-Lite resources to users through HTTP. Handles both the grid in the cms-lite UI and REST requests for particular resources.

## Methods

### addContent

public void **addContent** (*String type, String name, String language, String value, MultipartFile contentFile*)

Creates a new content in the system. Performed by the UI.

#### Parameters

- **type** – the type of the content, either **string** or **stream**
- **name** – the name of the content
- **language** – the language of the content
- **value** – the value of the content in case of a string content
- **contentFile** – the file to be saved as the content in case of a stream content

#### Throws

- **IOException** – if there were IO issues with the uploaded file
- **CMSLiteException** – if a wrong combination of parameters was passed

### autocompleteField

public *Set<String>* **autocompleteField** (*String field, String term*)

Handles the auto-complete feature for cms-lite. Will return auto-completions for the given field.

#### Parameters

- **field** – the field to auto-complete, name and language are currently supported
- **term** – what should be auto-complete, what the user has entered

#### Throws

- **CMSLiteException** – if the field is not supported

**Returns** a set of auto-completion values

#### **editStreamContent**

public void **editStreamContent** (*String language*, *String name*, *MultipartFile contentFile*)

Updates an existing stream content in the CMS system. Performed by the UI.

##### **Parameters**

- **language** – the language for the content
- **name** – the name of the content
- **contentFile** – the file that should be persisted as this content

##### **Throws**

- **IOException** – if there were IO issues with the uploaded file
- **ContentNotFoundException** – if the content does not exist

#### **editStringContent**

public void **editStringContent** (*String language*, *String name*, *String value*)

Updates an existing string content in the CMS system. Performed by the UI.

##### **Parameters**

- **language** – the language for the content
- **name** – the name of the content
- **value** – the value of the content

##### **Throws**

- **ContentNotFoundException** – if the content does not exist

#### **getAllLanguages**

public *Set<String>* **getAllLanguages** ()

Returns all languages associated with any content in cms-lite.

**Returns** all languages from the system

#### **getContent**

public *Content* **getContent** (*String type*, *String language*, *String name*)

Retrieves a content of a given type for the UI.

##### **Parameters**

- **type** – the type of the content, either **stream** or **string**
- **language** – the language of the content
- **name** – the name of the content

##### **Throws**

- **ContentNotFoundException** – if the content does not exist
- **CMSLiteException** – if the provided content type is invalid

**Returns** the matching content object

**getContents**

public [Resources](#) **getContents** ([GridSettings](#) settings)

Retrieves resource entries for the jqgrid. The returned object will not contain the content, only their types, names and languages.

**Parameters**

- **settings** – the settings of the grid, filtering and paging information

**Returns** all resources matching the criteria

**getStreamContent**

public void **getStreamContent** ([String](#) language, [String](#) name, [HttpServletResponse](#) response)

Retrieves the given stream content in raw form. Used by external callers to retrieve the content. This writes the content to the HttpServletResponse object and also sets the mime on that response to the type of the content.

**Parameters**

- **language** – the language of the content
- **name** – the name of the content
- **response** – the http response into which the content will be written to

**Throws**

- **IOException** – if there was an IO error writing to the response
- **ContentNotFoundException** – if the content does not exist

**getStringContent**

public void **getStringContent** ([String](#) language, [String](#) name, [HttpServletResponse](#) response)

Retrieves the given string content in raw form. Used by external callers to retrieve the content. This writes the content as text to the HttpServletResponse object and also sets the mime to text/plain.

**Parameters**

- **language** – the language of the content
- **name** – the name of the content
- **response** – the http response into which the content will be written to

**Throws**

- **IOException** – if there was an IO error writing to the response
- **ContentNotFoundException** – if the content does not exist

**handleCMSLiteException**

public [String](#) **handleCMSLiteException** ([Exception](#) e)

Handler for the [org.motechproject.cmslite.model.CMSLiteException](#). Will return 400 (Bad Request) as the HTTP response code and the message from the exception as the body.

**Parameters**

- **e** – the exception to handle

**Returns** the message from the exception, to be displayed as the response body

**handleContentNotFoundException**

```
public String handleContentNotFoundException (Exception e)
```

Handler for the `org.motechproject.cmslite.model.ContentNotFoundException`. Will return 404 (Not Found) as the HTTP response code and the message from the exception as the body.

**Parameters**

- **e** – the exception to handle

**Returns** the message from the exception, to be displayed as the response body

**removeContent**

```
public void removeContent (String type, String language, String name)
```

Removes the content of the given type from the CMS system. Performed by the UI.

**Parameters**

- **type** – the type of the content
- **language** – the language of the content
- **name** – the name of the content

**Throws**

- **ContentNotFoundException** – if the content does not exist
- **CMSLiteException** – if a wrong type was provided

**ResourceDto**

```
public class ResourceDto implements Serializable
```

Represent a content from CMS-Lite in the jqgrid. Does not carry any actual content, only information about the name, type and languages of the content.

**Constructors****ResourceDto**

```
public ResourceDto (Content content)
```

Constructs an instance for the given content.

**Parameters**

- **content** – the content this instance should represent

**ResourceDto**

```
public ResourceDto (String name, String type, String... languages)
```

**Parameters**

- **name** – the name of the content
- **type** – the type of the content, either **string** or **stream**
- **languages** – the languages supported by this content

## Methods

### addLanguage

public void **addLanguage** (*String language*)  
Adds a language to the set of languages for this content.

#### Parameters

- **language** – the language to add

### equals

public boolean **equals** (*Object obj*)

### getLanguages

public *Set<String>* **getLanguages** ()

**Returns** the languages supported by this content

### getName

public *String* **getName** ()

**Returns** the name of the content

### getType

public *String* **getType** ()

**Returns** the type of the content, either **string** or **stream**

### hashCode

public int **hashCode** ()

### toString

public *String* **toString** ()

## ResourceFilter

public final class **ResourceFilter**

This class is responsible for filtering content instances based on the provided grid settings.

## Methods

### equalsContent

public static boolean **equalsContent** (*ResourceDto dto*, *String contentName*, *String contentType*)  
Checks whether the `org.motechproject.cmslite.web.ResourceDto` matches the given content name and content type.

#### Parameters

- **dto** – the dto to check
- **contentName** – the expected name
- **contentType** – the expected type, **string** or **stream**

**Returns** true if the dto matches the parameters, false otherwise

#### **filter**

public static `List<ResourceDto>` **filter** (`GridSettings settings`, `List<Content> contents`)

Filters the given list of contents based on the grid settings.

#### **Parameters**

- **settings** – the grid settings to use
- **contents** – a list of all contents

**Returns** a new list with the matching content instances

#### **getContentType**

public static `String` **getContentType** (`Content content`)

Returns the type, as in string or stream, for this content.

#### **Parameters**

- **content** – the content to check

**Returns** **string** for string content, **stream** for stream content and a blank string for other cases (null content)

## **Resources**

public class **Resources** implements `Serializable`

A collection of `org.motechproject.cmslite.web.ResourceDto` objects to be displayed on the jq-grid. It also contains paging information for the grid.

## **Constructors**

### **Resources**

public **Resources** (`GridSettings settings`, `List<ResourceDto> list`)

Constructs an instance from the list of resources (by creating a sub-list) and the given grid settings.

#### **Parameters**

- **settings** – the grid settings to use
- **list** – the list of all available records

## **Methods**

### **getPage**

public `Integer` **getPage** ()

**Returns** the page that is being displayed

### **getRecords**

public `Integer` **getRecords** ()

**Returns** the total number of records in the system



**getRows**

```
public List<ResourceDto> getRows ()
```

**Returns** list of `org.motechproject.cmslite.web.ResourceDto` instances that will displayed on the grid

**getTotal**

```
public Integer getTotal ()
```

**Returns** the total number of pages available

**toString**

```
public String toString ()
```

### 19.14.17 org.motechproject.commcare.builder

#### CommcareFormBuilder

```
public class CommcareFormBuilder
```

Can be used for creating instances of the `CommcareForm` class based on the parameters stored in the passed event.

#### Methods

**buildFrom**

```
public CommcareForm buildFrom (MotechEvent motechEvent)
```

Creates an instance of the `CommcareForm` class based on the parameters stored in the passed event.

**Parameters**

- **motechEvent** – the event storing form information

**Returns** the instance of the `CommcareForm` built based on the parameters stored in the passed event

### 19.14.18 org.motechproject.commcare.config

#### AccountConfig

```
public class AccountConfig
```

Stores information about single commcare account(URL, domain, username and password).

#### Methods

**getBaseUrl**

```
public String getBaseUrl ()
```

**getDomain**

```
public String getDomain ()
```

#### **getPassword**

public [String](#) **getPassword** ()

#### **getUsername**

public [String](#) **getUsername** ()

#### **setBaseUrl**

public void **setBaseUrl** ([String](#) *baseUrl*)

#### **setDomain**

public void **setDomain** ([String](#) *domain*)

#### **setPassword**

public void **setPassword** ([String](#) *password*)

#### **setUsername**

public void **setUsername** ([String](#) *username*)

### **Config**

public class **Config**

Represents a single Commcare configuration.

#### **Fields**

##### **CONFIG\_BASE\_URL**

public static final [String](#) **CONFIG\_BASE\_URL**

##### **CONFIG\_DOMAIN**

public static final [String](#) **CONFIG\_DOMAIN**

##### **CONFIG\_NAME**

public static final [String](#) **CONFIG\_NAME**

##### **CONFIG\_PASSWORD**

public static final [String](#) **CONFIG\_PASSWORD**

##### **CONFIG\_USERNAME**

public static final [String](#) **CONFIG\_USERNAME**

#### **Methods**

##### **getAccountConfig**

public [AccountConfig](#) **getAccountConfig** ()

**getEventStrategy**

```
public String getEventStrategy ()
```

**getName**

```
public String getName ()
```

**isForwardCases**

```
public boolean isForwardCases ()
```

**isForwardForms**

```
public boolean isForwardForms ()
```

**isForwardSchema**

```
public boolean isForwardSchema ()
```

**isForwardStubs**

```
public boolean isForwardStubs ()
```

**setAccountConfig**

```
public void setAccountConfig (AccountConfig accountConfig)
```

**setEventStrategy**

```
public void setEventStrategy (String eventStrategy)
```

**setForwardCases**

```
public void setForwardCases (boolean forwardCases)
```

**setForwardForms**

```
public void setForwardForms (boolean forwardForms)
```

**setForwardSchema**

```
public void setForwardSchema (boolean forwardSchema)
```

**setForwardStubs**

```
public void setForwardStubs (boolean forwardStubs)
```

**setName**

```
public void setName (String name)
```

**Configs**

```
public class Configs
```

```
    Stores information about all the created Commcare configurations.
```

## Constructors

### Configs

public **Configs** ()

### Configs

public **Configs** (List<Config> configs)

## Methods

### contains

public boolean **contains** (Config configuration)

Checks whether given configuration already exists.

#### Parameters

- **configuration** – the configuration to be checked

**Returns** true if configuration exists, false otherwise

### deleteConfig

public void **deleteConfig** (String name)

Deletes the configuration with the given name.

#### Parameters

- **name** – the name of the configuration

### getByName

public Config **getByName** (String name)

Returns configuration by its name.

#### Parameters

- **name** – the name of the configuration, null will return default configuration

#### Throws

- **ConfigurationNotFoundException** – when configuration with the given name doesn't exist or if no default configuration was defined (only if null was passed as parameter)

**Returns** the configuration with given name or default configuration if null was given

### getConfigs

public List<Config> **getConfigs** ()

### getDefault

public Config **getDefault** ()

Returns default configuration.

#### Throws

- **ConfigurationNotFoundException** – if there were no default configuration defined

**Returns** the default configuration

**getDefaultConfigName**

```
public String getDefaultConfigName ()
```

**hasDefault**

```
public boolean hasDefault ()
```

Checks whether default configuration has been set.

**Returns** true if there is default configuration, else otherwise

**nameInUse**

```
public boolean nameInUse (String name)
```

Checks whether configuration with given name already exists.

**Parameters**

- **name** – the name of the configuration

**Returns** true if configuration with given name already exists, false otherwise

**saveConfig**

```
public void saveConfig (Config config)
```

Saves given configuration.

**Parameters**

- **config** – the configuration to save

**setConfigs**

```
public void setConfigs (List<Config> configs)
```

**setDefaultConfigName**

```
public boolean setDefaultConfigName (String name)
```

**updateConfig**

```
public void updateConfig (Config config)
```

Updates given configuration

**Parameters**

- **config** – the update source

**validateUrlAndDomain**

```
public boolean validateUrlAndDomain (String baseUrl, String domain)
```

Checks if there is configuration for given URL and domain combination.

**Parameters**

- **baseUrl** – the base URL of the configuration
- **domain** – the domain of the configuration

**Returns** true of configuration exists, false otherwise

## 19.14.19 org.motechproject.commcare.domain

### AppStructureResponseJson

public class **AppStructureResponseJson**

Wrapper class for storing list of instances of the `CommcareApplicationJson` class and their metadata. It's a class representing data as sent from the CommCareHQ server.

#### Methods

##### getApplications

public `List<CommcareApplicationJson>` **getApplications** ()

##### getMetadata

public `CommcareMetadataJson` **getMetadata** ()

### CaseInfo

public class **CaseInfo**

Represents a single CommCareHQ case. A case can be literally anything that one would want to track and monitor over time. It's a part of the MOTech model.

#### Methods

##### getCaseId

public `String` **getCaseId** ()

##### getCaseName

public `String` **getCaseName** ()

##### getCaseType

public `String` **getCaseType** ()

##### getConfigName

public `String` **getConfigName** ()

##### getDateClosed

public `String` **getDateClosed** ()

##### getDateOpened

public `String` **getDateOpened** ()

##### getDomain

public `String` **getDomain** ()

**getFieldValues**

```
public Map<String, String> getFieldValues ()
```

**getIndices**

```
public Map<String, Map<String, String>> getIndices ()
```

**getOwnerId**

```
public String getOwnerId ()
```

**getServerDateModified**

```
public String getServerDateModified ()
```

**getServerDateOpened**

```
public String getServerDateOpened ()
```

**getUserId**

```
public String getUserId ()
```

**getVersion**

```
public String getVersion ()
```

**getXformIds**

```
public List<String> getXformIds ()
```

**isClosed**

```
public boolean isClosed ()
```

**setCaseId**

```
public void setCaseId (String caseId)
```

**setCaseName**

```
public void setCaseName (String caseName)
```

**setCaseType**

```
public void setCaseType (String caseType)
```

**setClosed**

```
public void setClosed (boolean closed)
```

**setConfigName**

```
public void setConfigName (String configName)
```

**setDateClosed**

```
public void setDateClosed (String dateClosed)
```

**setDateOpened**

public void **setDateOpened** (*String dateOpened*)

**setDomain**

public void **setDomain** (*String domain*)

**setFieldValues**

public void **setFieldValues** (*Map<String, String> fieldValues*)

**setIndices**

public void **setIndices** (*Map<String, Map<String, String>> indices*)

**setOwnerId**

public void **setOwnerId** (*String ownerId*)

**setServerDateModified**

public void **setServerDateModified** (*String serverDateModified*)

**setServerDateOpened**

public void **setServerDateOpened** (*String serverDateOpened*)

**setUserId**

public void **setUserId** (*String userId*)

**setVersion**

public void **setVersion** (*String version*)

**setXformIds**

public void **setXformIds** (*List<String> xformIds*)

**CaseJson**

public class **CaseJson**

Represents a single CommCareHQ case. A case can be literally anything that one would want to track and monitor over time. It's a part of the CommCareHQ model.

**Methods**

**getCaseId**

public *String* **getCaseId** ()

**getDateClosed**

public *String* **getDateClosed** ()



**getDateModified**

```
public String getDateModified ()
```

**getDomain**

```
public String getDomain ()
```

**getIndices**

```
public Map<String, Map<String, String>> getIndices ()
```

**getProperties**

```
public Map<String, String> getProperties ()
```

**getServerDateModified**

```
public String getServerDateModified ()
```

**getServerDateOpened**

```
public String getServerDateOpened ()
```

**getUserId**

```
public String getUserId ()
```

**getVersion**

```
public String getVersion ()
```

**getXformIds**

```
public List<String> getXformIds ()
```

**isClosed**

```
public boolean isClosed ()
```

**setCaseId**

```
public void setCaseId (String caseId)
```

**setClosed**

```
public void setClosed (boolean closed)
```

**setDateClosed**

```
public void setDateClosed (String dateClosed)
```

**setDateModified**

```
public void setDateModified (String dateModified)
```

**setDomain**

```
public void setDomain (String domain)
```

#### **setIndices**

public void **setIndices** (Map<String, Map<String, String>> *indices*)

#### **setServerDateModified**

public void **setServerDateModified** (String *serverDateModified*)

#### **setServerDateOpened**

public void **setServerDateOpened** (String *serverDateOpened*)

#### **setUserId**

public void **setUserId** (String *userId*)

#### **setVersion**

public void **setVersion** (String *version*)

#### **setXformIds**

public void **setXformIds** (List<String> *xformIds*)

### **CaseResponseJson**

public class **CaseResponseJson**

Wrapper class for storing list of instances of the [CaseJson](#) class and their metadata. It's part of the Comm-CareHQ model.

#### **Methods**

##### **getCases**

public List<[CaseJson](#)> **getCases** ()

##### **getMetadata**

public [CommcareMetadataJson](#) **getMetadata** ()

### **CaseTask**

public class **CaseTask**

Domain class that represents a series of case information to be converted into case XML for upload to Comm-CareHQ.

#### **Methods**

##### **getCaseId**

public String **getCaseId** ()

##### **getCloseTask**

public [CloseTask](#) **getCloseTask** ()

**getCreateTask**

public [CreateTask](#) **getCreateTask** ()

**getDateModified**

public [String](#) **getDateModified** ()

**getIndexTask**

public [IndexTask](#) **getIndexTask** ()

**getUpdateTask**

public [UpdateTask](#) **getUpdateTask** ()

**getUserId**

public [String](#) **getUserId** ()

**getXmlns**

public [String](#) **getXmlns** ()

**setCaseId**

public void **setCaseId** ([String](#) *caseId*)

**setCloseTask**

public void **setCloseTask** ([CloseTask](#) *closeTask*)

**setCreateTask**

public void **setCreateTask** ([CreateTask](#) *createTask*)

**setDateModified**

public void **setDateModified** ([String](#) *dateModified*)

**setIndexTask**

public void **setIndexTask** ([IndexTask](#) *indexTask*)

**setUpdateTask**

public void **setUpdateTask** ([UpdateTask](#) *updateTask*)

**setUserId**

public void **setUserId** ([String](#) *userId*)

**setXmlns**

public void **setXmlns** ([String](#) *xmlns*)

## CaseXml

public class **CaseXml**

Represents a single case that it sent to the MOTech endpoint by the CommCareHQ server.

### Constructors

#### CaseXml

public **CaseXml** ()

### Methods

#### addFieldValue

public void **addFieldValue** (*String nodeName*, *String nodeValue*)

#### getAction

public *String* **getAction** ()

#### getApiKey

public *String* **getApiKey** ()

#### getCaseDataXmlns

public *String* **getCaseDataXmlns** ()

#### getCaseId

public *String* **getCaseId** ()

#### getCaseName

public *String* **getCaseName** ()

#### getCaseType

public *String* **getCaseType** ()

#### getDateModified

public *String* **getDateModified** ()

#### getFieldValues

public *Map*<*String*, *String*> **getFieldValues** ()

#### getOwnerId

public *String* **getOwnerId** ()

#### getServerModifiedOn

public *String* **getServerModifiedOn** ()

**getUserId**

```
public String getUserId ()
```

**setAction**

```
public void setAction (String action)
```

**setApiKey**

```
public void setApiKey (String apiKey)
```

**setCaseDataXmlns**

```
public void setCaseDataXmlns (String caseDataXmlns)
```

**setCaseId**

```
public void setCaseId (String caseId)
```

**setCaseName**

```
public void setCaseName (String caseName)
```

**setCaseType**

```
public void setCaseType (String caseType)
```

**setDateModified**

```
public void setDateModified (String dateModified)
```

**setFieldValues**

```
public void setFieldValues (Map<String, String> fieldValues)
```

**setOwnerId**

```
public void setOwnerId (String ownerId)
```

**setServerModifiedOn**

```
public void setServerModifiedOn (String serverModifiedOn)
```

**setUserId**

```
public void setUserId (String userId)
```

**CasesInfo**

```
public class CasesInfo
```

Wrapper class for storing list of instances of the `CaseInfo` class and their metadata. It's a part of the MOTECH model.

**Constructors****CasesInfo**

```
public CasesInfo (List<CaseInfo> caseInfoList, CommcareMetadataInfo metadataInfo)
```

## Methods

### getCaseInfoList

public [List](#)<[CaseInfo](#)> **getCaseInfoList** ()

### getMetadataInfo

public [CommcareMetadataInfo](#) **getMetadataInfo** ()

### setCaseInfoList

public void **setCaseInfoList** ([List](#)<[CaseInfo](#)> *caseInfoList*)

### setMetadataInfo

public void **setMetadataInfo** ([CommcareMetadataInfo](#) *metadataInfo*)

## CloseTask

public class **CloseTask**

A domain object to be included in a CaseTask in order to generate a close element in case XML.

## Constructors

### CloseTask

public **CloseTask** (boolean *close*)

## Methods

### isClose

public boolean **isClose** ()

### setClose

public void **setClose** (boolean *close*)

## CommcareAccountSettings

public class **CommcareAccountSettings**

Stores information about a single CommCareHQ account. This information can then be used for connecting to CommCareHQ server and fetching/sending data from/to it.

## Methods

### canMakeConnection

public boolean **canMakeConnection** ()

### getCommcareBaseUrl

public [String](#) **getCommcareBaseUrl** ()

**getCommcareDomain**

```
public String getCommcareDomain ()
```

**getPassword**

```
public String getPassword ()
```

**getUsername**

```
public String getUsername ()
```

**setCommcareBaseUrl**

```
public void setCommcareBaseUrl (String commcareBaseUrl)
```

**setCommcareDomain**

```
public void setCommcareDomain (String commcareDomain)
```

**setPassword**

```
public void setPassword (String password)
```

**setUsername**

```
public void setUsername (String username)
```

**CommcareApplicationJson**

```
public class CommcareApplicationJson
```

Represents a single CommCare application. CommCare application is a complete CommCare executable that gets downloaded and installed on a phone. This is a schema representation of such an application. It's part of the MOTech model.

**Constructors****CommcareApplicationJson**

```
public CommcareApplicationJson ()
```

**CommcareApplicationJson**

```
public CommcareApplicationJson (String commcareAppId, String applicationName, String re-  
sourceUri, List<CommcareModuleJson> modules)
```

**Methods****equals**

```
public boolean equals (Object obj)
```

**getApplicationName**

```
public String getApplicationName ()
```

**getCommcareAppId**

```
public String getCommcareAppId ()
```

**getConfigName**

```
public String getConfigName ()
```

**getModules**

```
public List<CommcareModuleJson> getModules ()
```

**getResourceUri**

```
public String getResourceUri ()
```

**getSerializedModules**

```
public String getSerializedModules ()
```

**hashCode**

```
public int hashCode ()
```

**serializeModules**

```
public final void serializeModules ()
```

**setApplicationName**

```
public void setApplicationName (String applicationName)
```

**setCommcareAppId**

```
public void setCommcareAppId (String commcareAppId)
```

**setConfigName**

```
public void setConfigName (String domain)
```

**setModules**

```
public final void setModules (List<CommcareModuleJson> modules)
```

**setResourceUri**

```
public void setResourceUri (String resourceUri)
```

**setSerializedModules**

```
public void setSerializedModules (String serializedModules)
```

**CommcareDataForwardingEndpoint**

```
public class CommcareDataForwardingEndpoint
```

A domain class that represents the information and properties of a data forwarding endpoint in CommCareHQ. It's part of the CommCareHQ model.



## Constructors

### **CommcareDataForwardingEndpoint**

public **CommcareDataForwardingEndpoint** ()

### **CommcareDataForwardingEndpoint**

public **CommcareDataForwardingEndpoint** (*String domain, String type, String url, String version*)

## Methods

### **getDomain**

public *String* **getDomain** ()

### **getId**

public *String* **getId** ()

### **getResourceUri**

public *String* **getResourceUri** ()

### **getType**

public *String* **getType** ()

### **getUrl**

public *String* **getUrl** ()

### **getVersion**

public *String* **getVersion** ()

### **setDomain**

public void **setDomain** (*String domain*)

### **setId**

public void **setId** (*String id*)

### **setResourceUri**

public void **setResourceUri** (*String resourceUri*)

### **setType**

public void **setType** (*String type*)

### **setUrl**

public void **setUrl** (*String url*)

### **setVersion**

public void **setVersion** (*String version*)

## CommcareDataForwardingEndpointsJson

public class **CommcareDataForwardingEndpointsJson**

Wrapper class for storing list of instances of the `CommcareDataForwardingEndpoint` class and their metadata. It's part of the CommCareHQ model.

### Methods

#### getMeta

public `CommcareMetadataJson` **getMeta** ()

#### getObjects

public `List<CommcareDataForwardingEndpoint>` **getObjects** ()

## CommcareFixture

public class **CommcareFixture**

A domain class that represents the information and properties of a fixture from CommCareHQ.

### Methods

#### getConfigName

public `String` **getConfigName** ()

#### getFields

public `Map<String, String>` **getFields** ()

#### getFixtureType

public `String` **getFixtureType** ()

#### getId

public `String` **getId** ()

#### setConfigName

public void **setConfigName** (`String configName`)

#### setFields

public void **setFields** (`Map<String, String> fields`)

#### setFixtureType

public void **setFixtureType** (`String fixtureType`)

#### setId

public void **setId** (`String id`)

## CommcareFixturesJson

public class **CommcareFixturesJson**

Wrapper class for storing a list of instances of the `CommcareFixture` class and their metadata. It's part of the CommCareHQ model.

### Methods

#### getMeta

public `Map<String, String>` **getMeta** ()

#### getObjects

public `List<CommcareFixture>` **getObjects** ()

## CommcareForm

public class **CommcareForm**

Represents a single CommCare form. Form is a series of questions and prompts that the user views sequentially on the phone. This is pretty much the core content of CommCare applications. When users submit data, they submit completed forms. Forms are based on the XForms standard.

### Methods

#### getConfigName

public `String` **getConfigName** ()

#### getForm

public `FormValueElement` **getForm** ()

#### getId

public `String` **getId** ()

#### getMd5

public `String` **getMd5** ()

#### getMetadata

public `Map<String, MetadataValue>` **getMetadata** ()

#### getReceivedOn

public `String` **getReceivedOn** ()

#### getResourceUri

public `String` **getResourceUri** ()

#### getType

public `String` **getType** ()

**getUiversion**

public [String](#) **getUiversion** ()

**getVersion**

public [String](#) **getVersion** ()

**isArchived**

public boolean **isArchived** ()

**setArchived**

public void **setArchived** (boolean *archived*)

**setConfigName**

public void **setConfigName** ([String](#) *configName*)

**setForm**

public void **setForm** ([FormValueElement](#) *form*)

**setId**

public void **setId** ([String](#) *id*)

**setMd5**

public void **setMd5** ([String](#) *md5*)

**setMetadata**

public void **setMetadata** ([Map](#)<[String](#), [MetadataValue](#)> *metadata*)

**setReceivedOn**

public void **setReceivedOn** ([String](#) *receivedOn*)

**setResourceUri**

public void **setResourceUri** ([String](#) *resourceUri*)

**setType**

public void **setType** ([String](#) *type*)

**setUiversion**

public void **setUiversion** ([String](#) *uiversion*)

**setVersion**

public void **setVersion** ([String](#) *version*)

## CommcareFormList

public class **CommcareFormList**

Represents a list of forms with metadata, as returned by Commcare.

### Methods

#### getMeta

public [CommcareMetadataJson](#) **getMeta** ()

**Returns** the metadata associated with this list

#### getObjects

public [List](#)<[CommcareForm](#)> **getObjects** ()

**Returns** the list of forms

#### setMeta

public void **setMeta** ([CommcareMetadataJson](#) *meta*)

##### Parameters

- **meta** – the metadata associated with this list

#### setObjects

public void **setObjects** ([List](#)<[CommcareForm](#)> *objects*)

##### Parameters

- **objects** – the list of forms

## CommcareLocation

public class **CommcareLocation**

Represents CommCare location - a part of the Locations API JSON, retrieved from CommCare.

**See also:** [CommcareLocationsJson](#)

### Methods

#### getCreatedAt

public [String](#) **getCreatedAt** ()

#### getDomain

public [String](#) **getDomain** ()

#### getExternalId

public [String](#) **getExternalId** ()

#### getId

public long **getId** ()

**getLastModified**

public [String](#) **getLastModified** ()

**getLatitude**

public [String](#) **getLatitude** ()

**getLocationData**

public [Map](#)<[String](#), [String](#)> **getLocationData** ()

**getLocationId**

public [String](#) **getLocationId** ()

**getLocationType**

public [String](#) **getLocationType** ()

**getLongitude**

public [String](#) **getLongitude** ()

**getName**

public [String](#) **getName** ()

**getParent**

public [String](#) **getParent** ()

**getResourceUri**

public [String](#) **getResourceUri** ()

**getSiteCode**

public [String](#) **getSiteCode** ()

**CommcareLocationsJson**

public class **CommcareLocationsJson**

Representation of a JSON response, retrieved from CommCareHQ for Locations API. Contains [CommcareMetadataJson](#) and a list of [CommcareLocation](#).

**Methods**

**getMeta**

public [CommcareMetadataJson](#) **getMeta** ()

**getObjects**

public [List](#)<[CommcareLocation](#)> **getObjects** ()

### CommcareMetadataInfo

public class **CommcareMetadataInfo**

Domain class representing case metadata, retrieved from CommCareHQ server.

#### Methods

##### **getLimit**

public int **getLimit** ()

##### **getNextPageQueryString**

public [String](#) **getNextPageQueryString** ()

##### **getOffset**

public int **getOffset** ()

##### **getPreviousPageQueryString**

public [String](#) **getPreviousPageQueryString** ()

##### **getTotalCount**

public int **getTotalCount** ()

##### **setLimit**

public void **setLimit** (int *limit*)

##### **setNextPageQueryString**

public void **setNextPageQueryString** ([String](#) *nextPageQueryString*)

##### **setOffset**

public void **setOffset** (int *offset*)

##### **setPreviousPageQueryString**

public void **setPreviousPageQueryString** ([String](#) *previousPageQueryString*)

##### **setTotalCount**

public void **setTotalCount** (int *totalCount*)

### CommcareMetadataJson

public class **CommcareMetadataJson**

Representation of the CommCare metadata, present in various Data APIs exposed by CommCare.

#### Methods

##### **getLimit**

public int **getLimit** ()

**getNextPageQueryString**

```
public String getNextPageQueryString ()
```

**getOffset**

```
public int getOffset ()
```

**getPreviousPageQueryString**

```
public String getPreviousPageQueryString ()
```

**getTotalCount**

```
public int getTotalCount ()
```

**CommcareModuleJson**

```
public class CommcareModuleJson implements Serializable
```

Represents a single CommCare module. A CommCare module is a set of forms related to one topic area. A single CommCare application can contain multiple modules. It's part of the MOTech model.

**Methods****getCaseProperties**

```
public List<String> getCaseProperties ()
```

**getCaseType**

```
public String getCaseType ()
```

**getFormSchemas**

```
public List<FormSchemaJson> getFormSchemas ()
```

**setCaseProperties**

```
public void setCaseProperties (List<String> caseProperties)
```

**setCaseType**

```
public void setCaseType (String caseType)
```

**setFormSchemas**

```
public void setFormSchemas (List<FormSchemaJson> formSchemas)
```

**CommcareUser**

```
public class CommcareUser
```

A domain class that represents the information and properties of a user from CommCareHQ.



## Methods

### **getDefaultPhoneNumber**

public [String](#) **getDefaultPhoneNumber** ()

### **getEmail**

public [String](#) **getEmail** ()

### **getFirstName**

public [String](#) **getFirstName** ()

### **getGroups**

public [List](#)<[String](#)> **getGroups** ()

### **getId**

public [String](#) **getId** ()

### **getLastName**

public [String](#) **getLastName** ()

### **getPhoneNumbers**

public [List](#)<[String](#)> **getPhoneNumbers** ()

### **getResourceUi**

public [String](#) **getResourceUi** ()

### **getUserData**

public [Map](#)<[String](#), [String](#)> **getUserData** ()

### **getUsername**

public [String](#) **getUsername** ()

### **setDefaultPhoneNumber**

public void **setDefaultPhoneNumber** ([String](#) *defaultPhoneNumber*)

### **setEmail**

public void **setEmail** ([String](#) *email*)

### **setFirstName**

public void **setFirstName** ([String](#) *firstName*)

### **setGroups**

public void **setGroups** ([List](#)<[String](#)> *groups*)

#### **setId**

public void **setId** (*String id*)

#### **setLastName**

public void **setLastName** (*String lastName*)

#### **setPhoneNumbers**

public void **setPhoneNumbers** (*List<String> phoneNumbers*)

#### **setResourceUi**

public void **setResourceUi** (*String resourceUi*)

#### **setUserData**

public void **setUserData** (*Map<String, String> userData*)

#### **setUsername**

public void **setUsername** (*String username*)

### **CommcareUsersJson**

public class **CommcareUsersJson**

A wrapper class for storing list of instances of the *CommcareUser* class and their metadata. It's part of the MOTECH model.

#### **Methods**

##### **getMeta**

public *Map<String, String>* **getMeta** ()

##### **getObjects**

public *List<CommcareUser>* **getObjects** ()

### **CreateTask**

public class **CreateTask**

A domain class to include in a CaseTask in order to generate a create block. Create blocks must have a case type and case name to be valid case XML for uploading to CommCareHQ.

#### **Methods**

##### **getCaseName**

public *String* **getCaseName** ()

##### **getCaseType**

public *String* **getCaseType** ()

**getOwnerId**

public [String](#) **getOwnerId** ()

**setCaseName**

public void **setCaseName** ([String](#) *caseName*)

**setCaseType**

public void **setCaseType** ([String](#) *caseType*)

**setOwnerId**

public void **setOwnerId** ([String](#) *ownerId*)

**FormNode**

public interface **FormNode**

Interface for form nodes. Form node can be either form element or form attribute.

**Fields****PREFIX\_ATTRIBUTE**

[String](#) **PREFIX\_ATTRIBUTE**

**PREFIX\_SEARCH\_FROM\_ROOT**

[String](#) **PREFIX\_SEARCH\_FROM\_ROOT**

**PREFIX\_SEARCH\_RELATIVE**

[String](#) **PREFIX\_SEARCH\_RELATIVE**

**PREFIX\_VALUE**

[String](#) **PREFIX\_VALUE**

**Methods****getValue**

[String](#) **getValue** ()

Returns the value of this node.

**Returns** the value of this node

**FormSchemaJson**

public class **FormSchemaJson** implements [Serializable](#)

Represents a single CommCare form schema.

## Methods

### getFormName

public [String](#) **getFormName** ()

Gets form name for English locale, or if English locale is not present, then returns the first one found.

**Returns** Form name for English locale, or random one, if not found

### getFormNames

public [Map](#)<[String](#), [String](#)> **getFormNames** ()

### getQuestions

public [List](#)<[FormSchemaQuestionJson](#)> **getQuestions** ()

### setFormNames

public void **setFormNames** ([Map](#)<[String](#), [String](#)> *formNames*)

### setQuestions

public void **setQuestions** ([List](#)<[FormSchemaQuestionJson](#)> *questions*)

## FormSchemaQuestionJson

public class **FormSchemaQuestionJson** implements [Serializable](#)

Represents a single question from a CommCare form.

## Methods

### getOptions

public [List](#)<[FormSchemaQuestionOptionJson](#)> **getOptions** ()

### getQuestionLabel

public [String](#) **getQuestionLabel** ()

### getQuestionRepeat

public [String](#) **getQuestionRepeat** ()

### getQuestionTag

public [String](#) **getQuestionTag** ()

### getQuestionValue

public [String](#) **getQuestionValue** ()

### setOptions

public void **setOptions** ([List](#)<[FormSchemaQuestionOptionJson](#)> *options*)

**setQuestionLabel**

public void **setQuestionLabel** (*String questionLabel*)

**setQuestionRepeat**

public void **setQuestionRepeat** (*String questionRepeat*)

**setQuestionTag**

public void **setQuestionTag** (*String questionTag*)

**setQuestionValue**

public void **setQuestionValue** (*String questionValue*)

**FormSchemaQuestionOptionJson**

public class **FormSchemaQuestionOptionJson** implements *Serializable*  
Represents a single answer for a specific question from a CommCare form.

**Methods****getLabel**

public *String* **getLabel** ()

**getValue**

public *String* **getValue** ()

**setLabel**

public void **setLabel** (*String label*)

**setValue**

public void **setValue** (*String value*)

**FormStubJson**

public class **FormStubJson**

Represents a single form stub. It only contains the date at which it was received, the form ID and the related cases IDs.

**Methods****getCaseIds**

public *List*<*String*> **getCaseIds** ()

**getFormId**

public *String* **getFormId** ()

#### **getReceivedOn**

public [String](#) **getReceivedOn** ()

#### **setCaseIds**

public void **setCaseIds** ([List](#)<[String](#)> *caseIds*)

#### **setFormId**

public void **setFormId** ([String](#) *formId*)

#### **setReceivedOn**

public void **setReceivedOn** ([String](#) *receivedOn*)

### **FormValueAttribute**

public class **FormValueAttribute** implements [FormNode](#)  
Represents a form value attribute.

#### **Constructors**

##### **FormValueAttribute**

public **FormValueAttribute** ([String](#) *value*)  
Creates new form value attribute with value set to *value*.

##### **Parameters**

- **value** – the value of the attribute

#### **Methods**

##### **getValue**

public [String](#) **getValue** ()

### **FormValueElement**

public class **FormValueElement** implements [FormNode](#)  
Object representation of the forwarded form from CommCareHQ.

#### **Methods**

##### **addAttribute**

public void **addAttribute** ([String](#) *key*, [String](#) *value*)

##### **addFormValueElement**

public void **addFormValueElement** ([String](#) *key*, [FormValueElement](#) *element*)

##### **containsAttribute**

public boolean **containsAttribute** ([String](#) *key*)

**getAllElements**

```
public List<FormValueElement> getAllElements (String elementName)
```

Returns a list of all descendant elements with a given element name, in the order in which they are encountered in a preorder traversal of this element tree. Lookup starts with this element and hence this element will be part of the returned list if it matches the given element name.

**Note:** This method will return an empty list if no matching node is found.

**Parameters**

- **elementName** – The name of the element to match on.

**Returns** A list containing matching elements.

**getAllElements**

```
public List<FormValueElement> getAllElements (String elementName, List<String> restrictedElements)
```

Returns a list of all descendant elements with a given element name, in the order in which they are encountered in a preorder traversal of this element tree. Lookup starts with this element and hence this element will be part of the returned list if it matches the given element name.

Lookup within an element and its descendants is skipped if it is part of `restrictedElements` list.

**Note:** This method will return an empty list if no matching node is found.

**Parameters**

- **elementName** – The name of the element to match on.
- **restrictedElements** – The list containing all restricted elements.

**Returns** A list containing matching elements.

**getAttributes**

```
public Map<String, String> getAttributes ()
```

**getChildElement**

```
public FormValueElement getChildElement (String elementName)
```

Returns child element with the given `elementName`.

**Parameters**

- **elementName** – the name of the child element

**Returns** the matching child element, null if element with the given name does not exist

**getChildElements**

```
public List<FormValueElement> getChildElements (String elementName)
```

Returns child element with the given `elementName` and wraps it into a list.

**Parameters**

- **elementName** – the name of the child element

**Returns** the matching child element wrapped into a list

**getElement**

```
public FormValueElement getElement (String elementName)
```

Returns first element found of all descendant elements with a given element name. The lookup happens in a preorder traversal of this element tree. Lookup starts with this element and hence this element will be returned if it matches the given element name.

**Note:** This method will return null if no matching node is found.

**Parameters**

- **elementName** – The name of the element to match on.

**Returns** First element matching the given name.

**getElement**

```
public FormValueElement getElement (String elementName, List<String> restrictedElements)
```

Returns first element found of all descendant elements with a given element name. The lookup happens in a preorder traversal of this element tree. Lookup starts with this element and hence this element will be part of the returned list if it matches the given element name.

Lookup within an element and its descendants is skipped if it is part of `restrictedElements` list.

**Note:** This method will return null if no matching node is found.

**Parameters**

- **elementName** – The name of the element to match on.
- **restrictedElements** – The list containing all restricted elements.

**Returns** First element matching the given name.

**getElementByAttribute**

```
public FormValueElement getElementByAttribute (String attribute, String value)
```

Returns element with the `attribute` attribute set to `value`.

**Parameters**

- **attribute** – the name of the attribute
- **value** – the value of the attribute

**Returns** the matching element, null if no matching element is found

**getElementByAttribute**

```
public FormValueElement getElementByAttribute (String attribute, String value, List<String> restrictedElements)
```

Returns element with the `attribute` attribute set to `value` and is not one of the `restrictedElements`.

**Parameters**

- **attribute** – the name of the attribute
- **value** – the value of the attribute
- **restrictedElements** – the restricted elements

**Returns** the matching element, null if no matching element is found

**getElementName**

```
public String getElementName ()
```



**getElements**

```
public Map<String, FormValueElement> getElements ()
```

Returns a map of form elements names and related elements.

**Returns** map of elements and their names

**getElementsByAttribute**

```
public List<FormValueElement> getElementsByAttribute (String attribute, String value)
```

Returns a list of elements with the `attribute` attribute set to `value`.

**Parameters**

- **attribute** – the name of the attribute
- **value** – the value of the attribute

**Returns** the list of matching elements

**getElementsByAttribute**

```
public List<FormValueElement> getElementsByAttribute (String attribute, String value, List<String>  
restrictedElements)
```

Returns a list of elements with the `attribute` attribute set to `value` that aren't any of the `restrictedElements`.

**Parameters**

- **attribute** – the name of the attribute
- **value** – the value of the attribute
- **restrictedElements** – the restricted elements

**Returns** the list of matching elements

**getElementsByAttribute**

```
public List<FormValueElement> getElementsByAttribute (String attribute, String value, List<String>  
restrictedElements, boolean breakOn-  
First)
```

Returns a list of elements with the `attribute` attribute set to `value` that aren't any of the `restrictedElements`. If the `breakOnFirst` is set to true, the list will contain only a single element.

**Parameters**

- **attribute** – the name of the attribute
- **value** – the value of the attribute
- **restrictedElements** – the restricted elements
- **breakOnFirst** – defines whether method should return a single value list or not

**Returns** the list of matching elements

**getSubElements**

```
public Multimap<String, FormValueElement> getSubElements ()
```

**getValue**

```
public String getValue ()
```

**search**

public [List](#)<[FormNode](#)> **search** ([String](#) path)

Returns a list of all descendant nodes (elements and attributes) found at the given path, in the order in which they are encountered in a preorder traversal of this element tree. Lookup starts with this element and hence this element will be part of the returned list if it matches the given path.

Path should start with “//”.

**Example:** For `FormValueElement` similar to following XML:

```
<this thisAttribute="thisAttribute">
  <foo>
    <bar barAttribute="barAttribute1">
      <child>child1</child>
      <child>child2</child>
    </bar>
    <child>child3</child>
  </foo>
  <foo>
    <bar barAttribute="barAttribute2" otherBarAttribute="otherBarAttribute">
      <child>child4</child>
      <child>child5</child>
      <otherChild>otherChild</otherChild>
    </bar>
  </foo>
</this>
```

Search for `//foo` will return both foo elements Search for `//foo/bar/child` will return elements having values **child1**, **child2**, **child4**, **child5**. Search for `//foo/child` will return one element having value as **child3**. Search for `//foo/bar/@barAttribute` will return two attribute nodes having values as **barAttribute1** and **barAttribute2**. Search for `//foo/bar/@otherBarAttribute` will return one attribute element having value as **otherBarAttribute**. Search for `//` will return **this** element. Search for `//@thisAttribute` will return “**thisAttribute** node on **this** element. Paths `//foo/bar/#anyValue` will return same result as `//foo/bar` or `//foo/bar/#anyOtherValue`.

**Note:** This method will return an empty list if no matching node is found.

**Parameters**

- **path** – Lookup path.

**Returns** A list containing matching nodes.

**searchFirst**

public [FormNode](#) **searchFirst** ([String](#) path)

Returns first node (elements and attributes) found at the given path, in the order in which they are encountered in a preorder traversal of this element tree. Lookup starts with this element and hence this element will be part of the returned list if it matches the given path.

For usage and other information see [search\(String\)](#).

**Note:** This method will return null if no matching node is found.

**Parameters**

- **path** – Lookup path.

**Returns** First node found at lookup path.

**setAttributes**

public void **setAttributes** ([Map](#)<[String](#), [String](#)> attributes)

**setElementName**

public void **setElementName** (*String elementName*)

**setSubElements**

public void **setSubElements** (*Multimap<String, FormValueElement> subElements*)

**setValue**

public void **setValue** (*String value*)

**IndexTask**

public class **IndexTask**

A domain class to include in a case task in order to generate an index element with IndexSubElements as the indices.

**Constructors****IndexTask**

public **IndexTask** (*List<IndexSubElement> indices*)

**Methods****getIndices**

public *List<IndexSubElement>* **getIndices** ()

**setIndices**

public void **setIndices** (*List<IndexSubElement> indices*)

**MetadataValue**

public class **MetadataValue** implements *Serializable*

Represents a value for a metadata entry in a Commcare form. Can have multiple string values.

**Constructors****MetadataValue**

public **MetadataValue** (*String value*)

Creates this object from a single value.

**Parameters**

- **value** – the metadata value

**MetadataValue**

public **MetadataValue** (*Collection<String> values*)

Creates this object from a collection of values.

**Parameters**

- **values** – the metadata values

## Methods

### firstValue

public `String` **firstValue** ()

Returns the first value from the metadata. Can be used to easily retrieve the value if we know that this entry has only a single value.

**Returns** the first value, or null if there are no values

### getValues

public `Collection<String>` **getValues** ()

**Returns** the values collection

### setValues

public void **setValues** (`Collection<String>` *values*)

#### Parameters

- **values** – the values collection

## SettingsDto

public class **SettingsDto**

Represents a single CommCare module configuration. It includes forwarding rules, event strategy and information about related CommCareHQ account.

## Constructors

### SettingsDto

public **SettingsDto** ()

## Methods

### getCommcareBaseUrl

public `String` **getCommcareBaseUrl** ()

### getCommcareDomain

public `String` **getCommcareDomain** ()

### getEventStrategy

public `String` **getEventStrategy** ()

### getPassword

public `String` **getPassword** ()

### getUsername

public `String` **getUsername** ()

**setCommcareBaseUrl**

public void **setCommcareBaseUrl** (*String commcareBaseUrl*)

**setCommcareDomain**

public void **setCommcareDomain** (*String commcareDomain*)

**setEventStrategy**

public void **setEventStrategy** (*String eventStrategy*)

**setForwardAppStructure**

public void **setForwardAppStructure** (boolean *forwardAppStructure*)

**setForwardCases**

public void **setForwardCases** (boolean *forwardCases*)

**setForwardFormStubs**

public void **setForwardFormStubs** (boolean *forwardFormStubs*)

**setForwardForms**

public void **setForwardForms** (boolean *forwardForms*)

**setPassword**

public void **setPassword** (*String password*)

**setUsername**

public void **setUsername** (*String username*)

**shouldForwardAppStructure**

public boolean **shouldForwardAppStructure** ()

**shouldForwardCases**

public boolean **shouldForwardCases** ()

**shouldForwardFormStubs**

public boolean **shouldForwardFormStubs** ()

**shouldForwardForms**

public boolean **shouldForwardForms** ()

**UpdateTask**

public class **UpdateTask**

A domain class to include in a case task in order to generate an upload case block in case xml.

## Methods

### **getCaseName**

public `String` **getCaseName** ()

### **getCaseType**

public `String` **getCaseType** ()

### **getDateOpened**

public `String` **getDateOpened** ()

### **getFieldValues**

public `Map<String, String>` **getFieldValues** ()

### **getOwnerId**

public `String` **getOwnerId** ()

### **setCaseName**

public void **setCaseName** (`String caseName`)

### **setCaseType**

public void **setCaseType** (`String caseType`)

### **setDateOpened**

public void **setDateOpened** (`String dateOpened`)

### **setFieldValues**

public void **setFieldValues** (`Map<String, String> fieldValues`)

### **setOwnerId**

public void **setOwnerId** (`String ownerId`)

## 19.14.20 org.motechproject.commcare.events

### **AppSchemaChangeEventHandler**

public class **AppSchemaChangeEventHandler**

The `AppSchemaChangeEventHandler` class listens to notifications about Schema changes and performs actions when such event is received. The current schema is dropped and a call to CommcareHQ is made to fetch the latest schema version. Once we parse the response, tasks are notified to update triggers.

## Methods

### schemaChange

public synchronized void **schemaChange** (MotechEvent *event*)

Responsible for handling `SCHEMA_CHANGE_EVENT`. This event is fired when the CommCare server sends a notification about schema change to the MOTech forwarding endpoint. All stored applications that originate from the configuration passed in the event will be deleted and new applications will be downloaded from the CommCare server.

#### Parameters

- **event** – the schema change event to be handled

## CaseEvent

public class **CaseEvent**

Wrapper class for convenience when a MotechEvent representing a CaseEvent is received.

## Constructors

### CaseEvent

public **CaseEvent** (String *caseId*)

Creates an instance of the `CaseEvent` class with the given *caseId*.

#### Parameters

- **caseId** – the ID of the case to create

### CaseEvent

public **CaseEvent** (MotechEvent *event*)

Creates an instance of the `CaseEvent` class based on the data passed as the given *event* parameters.

#### Parameters

- **event** – the event that stores data used for creating `CaseEvent`

## Methods

### fromCaseXml

public static `CaseEvent` **fromCaseXml** (`CaseXml` *caseInstance*, String *configName*)

Creates an instance of the `CaseEvent` class based on the given *caseInstance* and *configName*.

#### Parameters

- **caseInstance** – the case instance
- **configName** – the configuration name

**Returns** an instance of the `CaseEvent`

### getAction

public String **getAction** ()

**getApiKey**

public `String` **getApiKey** ()

**getCaseDataXmlns**

public `String` **getCaseDataXmlns** ()

**getCaseId**

public `String` **getCaseId** ()

**getCaseName**

public `String` **getCaseName** ()

**getCaseType**

public `String` **getCaseType** ()

**getConfigName**

public `String` **getConfigName** ()

**getDateModified**

public `String` **getDateModified** ()

**getFieldValues**

public `Map<String, String>` **getFieldValues** ()

**getOwnerId**

public `String` **getOwnerId** ()

**getServerModifiedOn**

public `String` **getServerModifiedOn** ()

**getUserId**

public `String` **getUserId** ()

**setAction**

public void **setAction** (`String` *action*)

**setApiKey**

public void **setApiKey** (`String` *apiKey*)

**setCaseDataXmlns**

public void **setCaseDataXmlns** (`String` *caseDataXmlns*)

**setCaseId**

public void **setCaseId** (`String` *caseId*)



**setCaseName**

public void **setCaseName** ([String caseName](#))

**setCaseType**

public void **setCaseType** ([String caseType](#))

**setConfigName**

public void **setConfigName** ([String configName](#))

**setDateModified**

public void **setDateModified** ([String dateModified](#))

**setFieldValues**

public void **setFieldValues** ([Map<String, String> fieldValues](#))

**setOwnerId**

public void **setOwnerId** ([String ownerId](#))

**setServerModifiedOn**

public void **setServerModifiedOn** ([String serverModifiedOn](#))

**setUserId**

public void **setUserId** ([String userId](#))

**toMotechEventWithData**

public MotechEvent **toMotechEventWithData** ()

Creates an instance of the `MotechEvent` class based on this `CaseEvent`. All data stored by this `CaseEvent` will be included in the event as parameters.

**Returns** an instance of the `MotechEvent`

**toMotechEventWithoutData**

public MotechEvent **toMotechEventWithoutData** ()

Creates an instance of the `MotechEvent` class based on this `CaseEvent`. No data but case ID, custom parser event key and configuration name will be included in the event as parameters.

**Returns** an instance of the `MotechEvent`

**ConfigurationEventHandler**

public class **ConfigurationEventHandler**

Listens to notifications about configurations and performs actions when such event is received. If configuration is deleted, its applications current schemas are dropped. If configuration is created a call to CommcareHQ is made to fetch the latest schemas version. Once we parse the response, tasks are notified to update triggers.

## Methods

### **configCreated**

public synchronized void **configCreated** (MotechEvent *event*)

Responsible for handling CONFIG\_CREATED event. This event is fired when user creates a new configuration. Handling this event will result in adding new configuration and downloading applications related with it.

#### **Parameters**

- **event** – the event to be handled

### **configDeleted**

public synchronized void **configDeleted** (MotechEvent *event*)

Responsible for handling CONFIG\_DELETED event. This event is fired when user deletes an existing configuration. Handling this event will result in removing the configuration itself and all related application stored in the database.

#### **Parameters**

- **event** – the event to be handled

### **configUpdated**

public synchronized void **configUpdated** (MotechEvent *event*)

Responsible for handling CONFIG\_UPDATED event. This event is fired when user updates an existing configuration. Handling this event will result in deleting all stored applications related with the updated configuration and downloading new ones from the CommCare server.

#### **Parameters**

- **event** – the event to be handled

## FullFormEvent

public class **FullFormEvent**

A utility class representing a full form event. This event signals a form received a form or a device log received in Motech, coming from Commcare.

## Constructors

### **FullFormEvent**

public **FullFormEvent** (FormValueElement *formValueElement*, String *receivedOn*, String *configName*)

Constructs this event.

#### **Parameters**

- **formValueElement** – the full form received
- **receivedOn** – the date the form was received - as the string from Commcare
- **configName** – the name of the Motech Commcare configuration for this form

## Methods

**toMotechEvent**

public MotechEvent **toMotechEvent** ()  
Builds a MotechEvent for this form event.  
**Returns** the Motech event to send

### FullFormFailureEvent

public class **FullFormFailureEvent**  
Represents a failure event for a full form received.

#### Constructors

**FullFormFailureEvent**  
public **FullFormFailureEvent** (String configName, String errorMessage)

##### Parameters

- **configName** – name of the Motech Commcare configuration this failure affected
- **errorMessage** – the error message

#### Methods

**toMotechEvent**  
public MotechEvent **toMotechEvent** ()  
Builds a MotechEvent instance to send.  
**Returns** the event to send

### MalformedFormStatusMessageEvent

public class **MalformedFormStatusMessageEvent**  
A wrapper for the event that will trigger a status message signaling a failure to receive a form. The status messages will be displayed by the Admin module in its UI (if the module is present).

#### Constructors

**MalformedFormStatusMessageEvent**  
public **MalformedFormStatusMessageEvent** (String message)

##### Parameters

- **message** – the message to display

#### Methods

**getMessage**  
public String **getMessage** ()  
**Returns** message the message to display

### **toMotechEvent**

public MotechEvent **toMotechEvent** ()

Builds a `MotechEvent` that will trigger the status message in the admin module.

**Returns** the Motech event to send

## **19.14.21 org.motechproject.commcare.events.constants**

### **EventDataKeys**

public final class **EventDataKeys**

Utility class for storing data keys for events.

#### **Fields**

##### **API\_KEY**

public static final `String` **API\_KEY**

##### **ATTRIBUTES**

public static final `String` **ATTRIBUTES**

##### **CASE\_ACTION**

public static final `String` **CASE\_ACTION**

##### **CASE\_DATA\_XMLNS**

public static final `String` **CASE\_DATA\_XMLNS**

##### **CASE\_ID**

public static final `String` **CASE\_ID**

##### **CASE\_IDS**

public static final `String` **CASE\_IDS**

##### **CASE\_NAME**

public static final `String` **CASE\_NAME**

##### **CASE\_TYPE**

public static final `String` **CASE\_TYPE**

##### **CONFIG\_BASE\_URL**

public static final `String` **CONFIG\_BASE\_URL**

##### **CONFIG\_DOMAIN**

public static final `String` **CONFIG\_DOMAIN**

**CONFIG\_NAME**

public static final [String](#) **CONFIG\_NAME**

**DATE\_MODIFIED**

public static final [String](#) **DATE\_MODIFIED**

**ELEMENT\_NAME**

public static final [String](#) **ELEMENT\_NAME**

**FAILED\_FORM\_MESSAGE**

public static final [String](#) **FAILED\_FORM\_MESSAGE**

**FIELD\_VALUES**

public static final [String](#) **FIELD\_VALUES**

**FORM\_ID**

public static final [String](#) **FORM\_ID**

**MESSAGE**

public static final [String](#) **MESSAGE**

**META\_APP\_VERSION**

public static final [String](#) **META\_APP\_VERSION**

**META\_BASE\_URL**

public static final [String](#) **META\_BASE\_URL**

**META\_DEVICE\_ID**

public static final [String](#) **META\_DEVICE\_ID**

**META\_DOMAIN**

public static final [String](#) **META\_DOMAIN**

**META\_INSTANCE\_ID**

public static final [String](#) **META\_INSTANCE\_ID**

**META\_TIME\_END**

public static final [String](#) **META\_TIME\_END**

**META\_TIME\_START**

public static final [String](#) **META\_TIME\_START**

**META\_USERNAME**

public static final [String](#) **META\_USERNAME**

#### **META\_USER\_ID**

public static final [String](#) **META\_USER\_ID**

#### **OWNER\_ID**

public static final [String](#) **OWNER\_ID**

#### **RECEIVED\_ON**

public static final [String](#) **RECEIVED\_ON**

#### **REGISTERED\_CASE\_NAME**

public static final [String](#) **REGISTERED\_CASE\_NAME**

#### **REGISTERED\_CASE\_TYPE**

public static final [String](#) **REGISTERED\_CASE\_TYPE**

#### **SERVER\_MODIFIED\_ON**

public static final [String](#) **SERVER\_MODIFIED\_ON**

#### **SUB\_ELEMENTS**

public static final [String](#) **SUB\_ELEMENTS**

#### **USER\_ID**

public static final [String](#) **USER\_ID**

#### **VALUE**

public static final [String](#) **VALUE**

### **EventSubjects**

public final class **EventSubjects**

Utility class for storing event subjects.

### **Fields**

#### **CASE\_EVENT**

public static final [String](#) **CASE\_EVENT**

#### **CONFIG\_CREATED**

public static final [String](#) **CONFIG\_CREATED**

#### **CONFIG\_DELETED**

public static final [String](#) **CONFIG\_DELETED**

#### **CONFIG\_UPDATED**

public static final [String](#) **CONFIG\_UPDATED**

**DEVICE\_LOG\_EVENT**

public static final [String](#) **DEVICE\_LOG\_EVENT**

**FORMS\_EVENT**

public static final [String](#) **FORMS\_EVENT**

**FORMS\_FAIL\_EVENT**

public static final [String](#) **FORMS\_FAIL\_EVENT**

**FORM\_STUB\_EVENT**

public static final [String](#) **FORM\_STUB\_EVENT**

**FORM\_STUB\_FAIL\_EVENT**

public static final [String](#) **FORM\_STUB\_FAIL\_EVENT**

**MALFORMED\_CASE\_EXCEPTION**

public static final [String](#) **MALFORMED\_CASE\_EXCEPTION**

**SCHEMA\_CHANGE\_EVENT**

public static final [String](#) **SCHEMA\_CHANGE\_EVENT**

## 19.14.22 org.motechproject.commcare.exception

### CaseParserException

public class **CaseParserException** extends [Exception](#)

Thrown when there were problems while parsing case or [org.motechproject.commcare.response.OpenRosaResponse](#).  
CommCare server response.

#### Constructors

**CaseParserException**

public **CaseParserException** ([String](#) *message*)

**CaseParserException**

public **CaseParserException** ([Exception](#) *ex*, [String](#) *message*)

### CommcareAuthenticationException

public class **CommcareAuthenticationException** extends [RuntimeException](#)

Thrown when there were problems while authenticating to the CommCare server. This exception might indicate that the given credentials are invalid.

## Constructors

### CommcareAuthenticationException

public **CommcareAuthenticationException** (*String message*)

### CommcareAuthenticationException

public **CommcareAuthenticationException** (*Exception ex, String message*)

## CommcareConnectionFailureException

public class **CommcareConnectionFailureException** extends *Exception*

Thrown when there were problems while connecting to the Commcare server.

## Constructors

### CommcareConnectionFailureException

public **CommcareConnectionFailureException** (*String message*)

### CommcareConnectionFailureException

public **CommcareConnectionFailureException** (*Exception ex, String message*)

## ConfigurationNotFoundException

public class **ConfigurationNotFoundException** extends *RuntimeException*

Thrown when there were problems with configuration e.g. unsupported configuration name, no default configuration selected.

## Constructors

### ConfigurationNotFoundException

public **ConfigurationNotFoundException** (*String message*)

## EndpointNotSupported

public class **EndpointNotSupported** extends *Exception*

Thrown when data comes for configuration that doesn't support this endpoint.

## Constructors

### EndpointNotSupported

public **EndpointNotSupported** (*String message*)

## FullFormParserException

public class **FullFormParserException** extends *Exception*

Thrown when there were problems while parsing an XML form.



## Constructors

### FullFormParserException

public **FullFormParserException** (*String message*)

### FullFormParserException

public **FullFormParserException** (*Exception ex, String message*)

## MalformedCaseXmlException

public class **MalformedCaseXmlException** extends *RuntimeException*

Thrown when the given XML is not a valid case.

## Constructors

### MalformedCaseXmlException

public **MalformedCaseXmlException** (*String message*)

### MalformedCaseXmlException

public **MalformedCaseXmlException** (*Exception ex, String message*)

## 19.14.23 org.motechproject.commcare.parser

### CaseParser

public class **CaseParser**<*T*>

Class for converting the XML passed in the constructor to the instance of the *CaseXml* class or domain class, which is defined by the class parameter.

#### Parameters

- *<T>* – the domain class

## Constructors

### CaseParser

public **CaseParser** (*Class<T> clazz, String xmlDocument*)

Creates an instance of the *CaseMapper* class. The created object will provide methods for parsing the given *xmlDocument* into objects of the given *clazz* or the *CaseXml* class.

#### Parameters

- *clazz* – the domain class
- *xmlDocument* – the case as a XML string

## Methods

### getCaseAction

public *String* **getCaseAction** ()

**parseCase**

```
public T parseCase ()
```

Parses the XML into an instance of the domain class defined in the class parameter.

**Throws**

- **CaseParserException** – if there were problems while parsing case XML

**Returns** the created instance

**parseCase**

```
public CaseXml parseCase (Document document)
```

Parses the given XML document into an instance of the **CaseXml** class.

**Parameters**

- **document** – the XML document

**Returns** the created instance

**FormAdapter**

```
public final class FormAdapter
```

Class responsible for deserializing forms from JSON.

**Methods****isAttribute**

```
public static boolean isAttribute (String key)
```

Checks whether the given *key* is an attribute.

**Parameters**

- **key** – the key to be checked

**Returns** true if the key is an attribute, false otherwise

**isValue**

```
public static boolean isValue (String key)
```

Checks whether the given *key* is a value.

**Parameters**

- **key** – the key to be checked

**Returns** true if the key is a value, false otherwise

**readJson**

```
public static CommcareForm readJson (String json)
```

Deserializes the given JSON string into an instance of the **CommcareForm** class.

**Parameters**

- **json** – the JSON representation of the form

**Returns** the deserialized form

**readListJson**

```
public static CommcareFormList readListJson (String json)
```

Reads a list of forms with metadata from the json file. The result is represented as a [CommcareFormList](#) object.

**Parameters**

- **json** – the json to read

**Returns** the parsed value

**recursivelyParse**

```
public static FormValueElement recursivelyParse (String key, JsonElement jsonElement)
```

Parses the given `jsonElement` into an instance of the [FormValueElement](#) class. The created object will have its name set to the key.

**Parameters**

- **key** – the name of the element to set
- **jsonElement** – the JSON element

**Returns** the parsed object

**FullFormParser**

```
public class FullFormParser
```

Class for converting the XML passed in as a constructor parameter to an instance of the [FormValueElement](#) class.

**Fields****DEVICE\_LOG**

```
public static final String DEVICE_LOG
```

**DEVICE\_REPORT\_ELEMENT**

```
public static final String DEVICE_REPORT_ELEMENT
```

**FORM**

```
public static final String FORM
```

**FORM\_DATA\_ELEMENT**

```
public static final String FORM_DATA_ELEMENT
```

**XMLNS\_ELEMENT**

```
public static final String XMLNS_ELEMENT
```

## Constructors

### FullFormParser

public **FullFormParser** (*String xmlDoc*)

Creates an instance of the `FullFormParser` which will provide a method for parsing the given `xmlDoc`.

#### Parameters

- **xmlDoc** – the XML document

## Methods

### parse

public `FormValueElement` **parse** ()

Method to parse incoming “full” XML forms from CommCareHQ. Parser assumes the form has an element with the tag “data”. If the form does not, and instead has a device report, a null form is returned. Otherwise an exception is thrown indicating an unknown or faulty form XML.

#### Throws

- **FullFormParserException** – if the form does not parse correctly and is not a device report form

**Returns** the parsed form, null if a device report

## OpenRosaResponseParser

public class **OpenRosaResponseParser**

Class responsible for parsing CommCareHQ server responses into instances of the `OpenRosaResponse` class.

## Methods

### parseResponse

public `OpenRosaResponse` **parseResponse** (*String response*)

Parses the given CommCareHQ server response into an instance of the `OpenRosaResponse` class.

#### Parameters

- **response** – the response to be parsed

#### Throws

- **CaseParserException** – if there where problems while parsing the response

**Returns** the parsed response

## 19.14.24 org.motechproject.commcare.request

### CaseRequest

public class **CaseRequest**

Represents a case XML request that can be sent to the CommCareHQ server.

## Constructors

### CaseRequest

public **CaseRequest** (*String caseId*, *String userId*, *String dateModified*, *String dataXmlns*)

Creates a case request with the caseId attribute set to caseId, the userId attribute to userId, the dateModified attribute set to dateModified and the dataXmlns attribute set to dataXmlns.

#### Parameters

- **caseId** – the value of the caseId attribute, which should be the ID of the case
- **userId** – the value of the userId attribute, which should be the ID of the user
- **dateModified** – the value of the dateModified attribute, which should be the last modification date
- **dataXmlns** – the value of the dataXmlns attribute, which should be the namespace of the XML element

### CaseRequest

public **CaseRequest** ()

## Methods

### getCaseId

public *String* **getCaseId** ()

### getCloseElement

public *CloseElement* **getCloseElement** ()

### getCreateElement

public *CreateElement* **getCreateElement** ()

### getDataXmlns

public *String* **getDataXmlns** ()

### getDateModified

public *String* **getDateModified** ()

### getIndexElement

public *IndexTask* **getIndexElement** ()

### getUpdateElement

public *UpdateTask* **getUpdateElement** ()

### getUserId

public *String* **getUserId** ()

#### **getXmlns**

public [String](#) **getXmlns** ()

#### **setCaseId**

public void **setCaseId** ([String](#) *caseId*)

#### **setCloseElement**

public void **setCloseElement** ([CloseElement](#) *closeElement*)

#### **setCreateElement**

public void **setCreateElement** ([CreateElement](#) *createElement*)

#### **setDataXmlns**

public void **setDataXmlns** ([String](#) *dataXmlns*)

#### **setDateModified**

public void **setDateModified** ([String](#) *dateModified*)

#### **setIndexElement**

public void **setIndexElement** ([IndexTask](#) *indexElement*)

#### **setUpdateElement**

public void **setUpdateElement** ([UpdateTask](#) *updateElement*)

#### **setUserId**

public void **setUserId** ([String](#) *userId*)

#### **setXmlns**

public void **setXmlns** ([String](#) *xmlns*)

### **CloseElement**

public class **CloseElement**

Represents a close element in an XML document.

### **CommcareRequestData**

public class **CommcareRequestData**

Represents a XML case request. Includes both metadata and case XML itself.

## Constructors

### CommcareRequestData

public **CommcareRequestData** (*String xmlns*, *MetaElement meta*, *CaseRequest ccCase*)

Creates a commcare request with the xmlns attribute set to *xmlns*, the meta element set to *meta* and the case element set to *ccCase*.

#### Parameters

- **xmlns** – the value of the xmlns attribute, which should be the namespace of the XML element
- **meta** – the value of the meta element, for more information about meta element, see {[@see MetaElement](#)}
- **ccCase** – the value of the case element, for more information about case element, see {[@see CaseRequest](#)}

## Methods

### getCcCase

public *CaseRequest* **getCcCase** ()

### getMeta

public *MetaElement* **getMeta** ()

### getXmlns

public *String* **getXmlns** ()

## CreateElement

public class **CreateElement**

Represents a create element in an XML document.

## Constructors

### CreateElement

public **CreateElement** (*String caseType*, *String caseName*, *String ownerId*)

Creates a new create XML element with maximum of three sub-elements - caseType, caseName and ownerId. Values of those elements will be equal to the given parameters. If any of the parameters is null the related element won't be added.

#### Parameters

- **caseType** – the value of the caseType element, which should contain the type of the case
- **caseName** – the value of the caseName element, which should contain the name of the case
- **ownerId** – the value of the ownerId element, which should contain the ID of the owner

## Methods

### getCaseName

public [String](#) **getCaseName** ()

### getCaseType

public [String](#) **getCaseType** ()

### getOwnerId

public [String](#) **getOwnerId** ()

### setCaseName

public void **setCaseName** ([String](#) *caseName*)

### setCaseType

public void **setCaseType** ([String](#) *caseType*)

### setOwnerId

public void **setOwnerId** ([String](#) *ownerId*)

## FormListRequest

public class **FormListRequest** implements [Serializable](#)  
Represents a request for a form list to the Commcare HQ.

## Methods

### addQueryParams

public void **addQueryParams** ([URIBuilder](#) *uriBuilder*)

Adds the params for this request to the *provided*. Allows to easily add params for this query to a Commcare url.

#### Parameters

- **uriBuilder** – the uri builder that should be used

### getAppVersion

public [String](#) **getAppVersion** ()

**Returns** the exact version of the CommCare application used to submit the form

### getPageNumber

public [Integer](#) **getPageNumber** ()

**Returns** the number of the page to fetch



**getPageSize**

```
public Integer getPageSize ()
```

**Returns** the size of the page to fetch

**getReceivedOnEnd**

```
public DateTime getReceivedOnEnd ()
```

**Returns** the end boundary for the received on datetime field

**getReceivedOnStart**

```
public DateTime getReceivedOnStart ()
```

**Returns** the starting boundary for the received on datetime field

**getXmlns**

```
public String getXmlns ()
```

Returns the form XML namespace for this query. (optional) You can find this for any given form by going to your form -> Advanced -> View (Source xml) and look for the address in the header of your xform. It is also in the raw xml of any submitted form.

**Returns** the xmlns

**isIncludeArchived**

```
public Boolean isIncludeArchived ()
```

**Returns** true if archived forms should be returned, false otherwise

**nextPage**

```
public int nextPage ()
```

Increments the current page number. Will set the current page to 1 if it was null.

**Returns** the new page number

**previousPage**

```
public int previousPage ()
```

Decrements the current page number.

**Returns** the new page number

**setAppVersion**

```
public void setAppVersion (String appVersion)
```

**Parameters**

- **appVersion** – the exact version of the CommCare application used to submit the form

**setIncludeArchived**

```
public void setIncludeArchived (Boolean includeArchived)
```

**Parameters**

- **includeArchived** – true if archived forms should be returned, false otherwise

### setPageNumber

public void **setPageNumber** (*Integer pageNumber*)

#### Parameters

- **pageNumber** – the number of the page to fetch

### setPageSize

public void **setPageSize** (*Integer pageSize*)

#### Parameters

- **pageSize** – the size of the page to fetch

### setReceivedOnEnd

public void **setReceivedOnEnd** (*DateTime receivedOnEnd*)

#### Parameters

- **receivedOnEnd** – the end boundary for the received on datetime field

### setReceivedOnStart

public void **setReceivedOnStart** (*DateTime receivedOnStart*)

#### Parameters

- **receivedOnStart** – the starting boundary for the received on datetime field

### setXmlns

public void **setXmlns** (*String xmlns*)

Sets the form XML namespace for this query. (optional) You can find this for any given form by going to your form -> Advanced -> View (Source xml) and look for the address in the header of your xform. It is also in the raw xml of any submitted form.

#### Parameters

- **xmlns** – the xmlns

## IndexSubElement

public class **IndexSubElement**

Domain class that is converted to indices in the index element of the case XML.

### Constructors

#### IndexSubElement

public **IndexSubElement** (*String caseId, String caseType, String indexNodeName*)

Creates a new indexSub XML with maximum of three sub-elements - caseId, caseType and indexNodeName. Values of those elements will be equal to the given parameters. If any of the parameters is null the related element won't be added.

#### Parameters

- **caseId** – the value of the caseId element
- **caseType** – the value of the caseType element

- **indexNodeName** – the value of the indexNodeName element

## Methods

### getCaseId

public **String** **getCaseId** ()

### getCaseType

public **String** **getCaseType** ()

### getIndexNodeName

public **String** **getIndexNodeName** ()

### setCaseId

public void **setCaseId** (**String** *caseId*)

### setCaseType

public void **setCaseType** (**String** *caseType*)

### setIndexNodeName

public void **setIndexNodeName** (**String** *indexNodeName*)

## MetaElement

public class **MetaElement**

Represents a meta element in an XML document.

## Constructors

### MetaElement

public **MetaElement** (**String** *xmlns*, **String** *instanceId*, **String** *timeStart*, **String** *timeEnd*, **String** *userId*)

Creates a meta XML element with maximum of five sub-elements - xmlns, instanceId, timeStart, timeEnd, userId. Values of those elements will be equal to the given parameters. If any of the parameters is null the related element won't be added.

#### Parameters

- **xmlns** – the value of the xmlns element, which should contain the namespace of the XML element
- **instanceId** – the value of the instanceId element, which should contain the ID of the instance
- **timeStart** – the value of the timeStart element, which should contain the start of the time range for which cases will be fetched
- **timeEnd** – the value of the timeEnd element, which should contain the end of the time range for which cases will be fetched
- **userId** – the value of the userId element, which should contain the ID of the user

## Methods

### **getInstanceID**

public [String](#) **getInstanceID** ()

### **getTimeEnd**

public [String](#) **getTimeEnd** ()

### **getTimeStart**

public [String](#) **getTimeStart** ()

### **getUserID**

public [String](#) **getUserID** ()

### **getXmlns**

public [String](#) **getXmlns** ()

## 19.14.25 org.motechproject.commcare.response

### OpenRosaResponse

public class **OpenRosaResponse**

An object representing the Open Rosa response from CommCareHQ when posting case XML.

## Methods

### **getMessageNature**

public [String](#) **getMessageNature** ()

### **getMessageText**

public [String](#) **getMessageText** ()

### **getStatus**

public int **getStatus** ()

### **setMessageNature**

public void **setMessageNature** ([String](#) *messageNature*)

### **setMessageText**

public void **setMessageText** ([String](#) *messageText*)

### **setStatus**

public void **setStatus** (int *status*)

## 19.14.26 org.motechproject.commcare.service

### CommcareAppStructureService

public interface **CommcareAppStructureService**

A service to perform queries against CommCareHQ application structure APIs.

#### Methods

##### getAllApplications

List<CommcareApplicationJson> **getAllApplications** (String *configName*)

Query CommCareHQ for structures of all applications.

##### Parameters

- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the list of the CommcareApplicationJson objects representing all applications found on the configured domain of CommCareHQ

##### getAllApplications

List<CommcareApplicationJson> **getAllApplications** ()

Same as `getAllApplications` but uses default Commcare configuration.

### CommcareApplicationDataService

public interface **CommcareApplicationDataService** extends MotechDataService<CommcareApplicationJson>

Data service for the `CommcareApplicationJson` class. Provides methods for managing instances of said class.

#### Methods

##### byApplicationName

CommcareApplicationJson **byApplicationName** (String *applicationName*)

Returns application with the given `applicationName`.

##### Parameters

- **applicationName** – the name of the application

**Returns** the matching application, null if application with the given `applicationName` does not exist

##### bySourceConfiguration

List<CommcareApplicationJson> **bySourceConfiguration** (String *configurationName*)

Returns a list of the applications that originate from the configuration with the given `configurationName`.

##### Parameters

- **configurationName** – the name of the configuration

**Returns** the list of matching applications

## CommcareCaseService

public interface **CommcareCaseService**

This service provides two main features: Interacting with CommCareHQ's programmatic case APIs and uploading case XML wrapped in a form instance to CommCareHQ.

### Methods

#### getCaseById

**CaseInfo** **getCaseById** (*String* caseId, *String* configName)

Query CommCareHQ for a case by its case id.

##### Parameters

- **caseId** – the id of the case on CommCareHQ
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the CaseInfo object representing the state of the case or null if that case does not exist.

#### getCaseById

**CaseInfo** **getCaseById** (*String* caseId)

Same as `getCaseById` but uses default Commcare configuration.

#### getCases

**List<CaseInfo>** **getCases** (*Integer* pageSize, *Integer* pageNumber, *String* configName)

Query CommCareHQ for all cases of a given page.

##### Parameters

- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the list of CaseInfo objects representing cases of the given page found on the given CommcareHQ configuration

#### getCases

**List<CaseInfo>** **getCases** (*Integer* pageSize, *Integer* pageNumber)

Same as `getCases` but uses default Commcare configuration.

#### getCasesByCasesNameAndTimeWithMetadata

**CasesInfo** **getCasesByCasesNameAndTimeWithMetadata** (*String* caseName, *String* dateModifiedStart, *String* dateModifiedEnd, *Integer* pageSize, *Integer* pageNumber, *String* configName)

Query CommCareHQ for all cases of a given case name, date modified range, page and cases metadata.

##### Parameters

- **caseName** – the case name of case on CommCareHQ
- **dateModifiedStart** – the start of the date range

- **dateModifiedEnd** – the end of the date range
- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the CasesInfo wrapper, containing CaseInfo objects and case metadata from CommCareHQ

#### getCasesByCasesNameAndTimeWithMetadata

**CasesInfo** **getCasesByCasesNameAndTimeWithMetadata** (*String caseName, String dateModifiedStart, String dateModifiedEnd, Integer pageSize, Integer pageNumber*)

Same as `getCasesByCasesNameAndTimeWithMetadata` but uses default Commcare configuration.

#### getCasesByCasesNameWithMetadata

**CasesInfo** **getCasesByCasesNameWithMetadata** (*String caseName, Integer pageSize, Integer pageNumber, String configName*)

Query CommCareHQ for all cases of a given case name, page and cases metadata.

##### Parameters

- **caseName** – the case name of case on CommCareHQ
- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the CasesInfo wrapper, containing CaseInfo objects and case metadata from CommCareHQ

#### getCasesByCasesNameWithMetadata

**CasesInfo** **getCasesByCasesNameWithMetadata** (*String caseName, Integer pageSize, Integer pageNumber*)

Same as `getCasesByCasesNameWithMetadata` but uses default Commcare configuration.

#### getCasesByCasesTimeWithMetadata

**CasesInfo** **getCasesByCasesTimeWithMetadata** (*String dateModifiedStart, String dateModifiedEnd, Integer pageSize, Integer pageNumber, String configName*)

Query CommCareHQ for all cases of a given date modified range, page and cases metadata.

##### Parameters

- **dateModifiedStart** – the start of the date range
- **dateModifiedEnd** – the end of the date range
- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the CasesInfo wrapper, containing CaseInfo objects and case metadata from CommCareHQ.

**getCasesByCasesTimeWithMetadata**

**CasesInfo** **getCasesByCasesTimeWithMetadata** (*String* dateModifiedStart, *String* dateModifiedEnd, *Integer* pageSize, *Integer* pageNumber)

Same as `getCasesByCasesTimeWithMetadata` but uses default Commcare configuration.

**getCasesByType**

**List<CaseInfo>** **getCasesByType** (*String* type, *Integer* pageSize, *Integer* pageNumber, *String* configName)

Query CommCareHQ for all cases of a given case type and page.

**Parameters**

- **type** – the type of case on CommCareHQ
- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the list of CaseInfo objects representing cases of the given type and page found on the given CommcareHQ configuration

**getCasesByType**

**List<CaseInfo>** **getCasesByType** (*String* type, *Integer* pageSize, *Integer* pageNumber)

Same as `getCasesByType` but uses default Commcare configuration.

**getCasesByUserId**

**List<CaseInfo>** **getCasesByUserId** (*String* userId, *Integer* pageSize, *Integer* pageNumber, *String* configName)

Query CommCareHQ for all cases under a given user id.

**Parameters**

- **userId** – the user id from CommCareHQ
- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the list of CaseInfo objects representing cases under the given user id and page found on the given CommcareHQ configuration

**getCasesByUserId**

**List<CaseInfo>** **getCasesByUserId** (*String* userId, *Integer* pageSize, *Integer* pageNumber)

Same as `getCasesByUserId` but uses default Commcare configuration.

**getCasesByUserIdAndType**

**List<CaseInfo>** **getCasesByUserIdAndType** (*String* userId, *String* type, *Integer* pageSize, *Integer* pageNumber, *String* configName)

Query CommCareHQ for all cases of a given case type, user id and page.

**Parameters**

- **userId** – the user id from CommCareHQ



- **type** – the type of case on CommCareHQ
- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the list of CaseInfo objects representing cases of the given type, user id and page found on the given CommcareHQ configuration

#### **getCasesByUserIdAndType**

List<CaseInfo> **getCasesByUserIdAndType** (String *userId*, String *type*, Integer *pageSize*, Integer *pageNumber*)

Same as `getCasesByUserIdAndType` but uses default Commcare configuration.

#### **getCasesWithMetadata**

CasesInfo **getCasesWithMetadata** (Integer *pageSize*, Integer *pageNumber*, String *configName*)

Query CommCareHQ for all cases of the given page and cases metadata

##### **Parameters**

- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the CasesInfo wrapper, containing CaseInfo objects and case metadata from CommCareHQ

#### **getCasesWithMetadata**

CasesInfo **getCasesWithMetadata** (Integer *pageSize*, Integer *pageNumber*)

Same as `getCasesWithMetadata` but uses default Commcare configuration.

#### **uploadCase**

OpenRosaResponse **uploadCase** (CaseTask *caseTask*, String *configName*)

Upload case xml wrapped in a minimal xform instance to CommCareHQ.

##### **Parameters**

- **caseTask** – An object representing the case information and case actions to be submitted as case xml
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** an informational object representing the status, nature and message of the response from CommCareHQ when attempting to upload this instance of case xml. Returns null if your case xml was incorrect.

#### **uploadCase**

OpenRosaResponse **uploadCase** (CaseTask *caseTask*)

Same as `uploadCase` but uses default Commcare configuration.

## CommcareConfigService

public interface **CommcareConfigService**

Interface responsible for managing multiple configurations.

### Methods

#### create

**Config** **create** ()

Returns the template for the configuration.

**Returns** the template for the configuration

#### deleteConfig

void **deleteConfig** (*String name*)

Deletes the configuration with the given name.

##### Parameters

- **name** – the name of the configuration

#### exists

boolean **exists** (*String config*)

Checks whether configuration with the given name exists.

**Returns** true if config with given name exists, false otherwise

#### getBaseUrl

**String** **getBaseUrl** ()

Returns the base URL for CommcareHQ forwarding endpoints.

**Returns** the base URL

#### getByName

**Config** **getByName** (*String name*)

Returns the configuration with the given name.

##### Parameters

- **name** – the name of the configuration

**Returns** the configuration with the given name

#### getConfigs

**Configs** **getConfigs** ()

Returns all the stored configurations.

**Returns** the stored configurations

#### getDefault

**Config** **getDefault** ()

Returns the default configuration.

**Returns** the default configuration

**saveConfig**

*Config* **saveConfig** (*Config configuration*)

Creates the given configuration.

**Parameters**

- **configuration** – the configuration to be saved

**Returns** the saved configuration

**setDefault**

void **setDefault** (*String name*)

Sets configuration with the given name as default.

**Parameters**

- **name** – the name of the configuration

**verifyConfig**

boolean **verifyConfig** (*Config config*)

Verifies CommCareHQ connection with the given configuration.

**Parameters**

- **config** – the configuration to be verified

**Throws**

- **IllegalArgumentException** – if one or more parameters are empty

**Returns** true if connection was successful

**CommcareDataForwardingEndpointService**

public interface **CommcareDataForwardingEndpointService**

A service to perform queries and updates against CommCareHQ's data forwarding endpoints.

**Methods****createNewDataForwardingRule**

boolean **createNewDataForwardingRule** (*CommcareDataForwardingEndpoint newForwardingRule*,  
*Config config*)

Creates a new data forwarding rule for the CommCareHQ.

**Returns** true - if data was successfully sent; otherwise - false

**getAllDataForwardingEndpoints**

List<*CommcareDataForwardingEndpoint*> **getAllDataForwardingEndpoints** (*Config config*)

Queries CommCareHQ for a list of all data forwarding rules on the configured domain.

**Parameters**

- **config** – the configuration to be used when connecting to the CommcareHQ server

**Returns** the list of CommcareDataForwardingEndpoints that represent the information about each data forwarding rule from CommCareHQ

**getDataForwardingEndpoints**

List<CommcareDataForwardingEndpoint> **getDataForwardingEndpoints** (*Integer* pageSize, *Integer* pageNumber, *Config* config)

Queries CommCareHQ for a list of data forwarding rules (located on the given page) on the configured domain.

**Parameters**

- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **config** – the configuration to be used when connecting to the CommcareHQ server

**Returns** the list of CommcareDataForwardingEndpoints that represent the information about each data forwarding rule located on the given page from CommCareHQ

**updateDataForwardingRule**

boolean **updateDataForwardingRule** (*CommcareDataForwardingEndpoint* updatedForwardingRule, *Config* config)

Updates specified CommCareHQ data forwarding rule.

**Parameters**

- **updatedForwardingRule** – the rule to be updated
- **config** – the configuration to be used when connecting to the CommcareHQ server

**Returns** true - if the data has been updated, false otherwise

**CommcareFixtureService**

public interface **CommcareFixtureService**

A service to perform queries against CommCareHQ fixture APIs.

**Methods****getCommcareFixtureById**

*CommcareFixture* **getCommcareFixtureById** (*String* id, *String* configName)

Queries CommCareHQ for a specific fixture

**Parameters**

- **id** – the id of the fixture to retrieve.
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the CommcareFixture object representing the information about the fixture from CommCareHQ, or null if that fixture did not exist on that domain.

**getCommcareFixtureById**

*CommcareFixture* **getCommcareFixtureById** (*String* id)

Same as `getCommcareFixtureById` but uses default configuration.

**getFixtures**

List<CommcareFixture> **getFixtures** (Integer pageSize, Integer pageNumber, String configName)

Queries CommCareHQ for a list of fixtures(located on the given page) on the configured domain.

**Parameters**

- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the list of CommcareFixture that represent the information about each fixture located on the given page from CommCareHQ

**getFixtures**

List<CommcareFixture> **getFixtures** (Integer pageSize, Integer pageNumber)

Same as `getFixtures` but uses default configuration.

**CommcareFormService**

public interface **CommcareFormService**

A service to perform queries against CommCareHQ form APIs.

**Methods****retrieveForm**

CommcareForm **retrieveForm** (String id, String configName)

Retrieves form with the given ID from server represented by the given configuration.

**Parameters**

- **id** – the ID of the form
- **configName** – the name of the configuration used for connecting to the CommcareHQ server, null means default configuration

**Returns** the form with the given ID

**retrieveForm**

CommcareForm **retrieveForm** (String id)

Same as `retrieveForm` but uses default Commcare configuration.

**retrieveFormJson**

String **retrieveFormJson** (String id, String configName)

Retrieves form with the given ID from server represented by the given configuration.

**Parameters**

- **id** – the ID of the form
- **configName** – the name of the configuration used for connecting to the CommcareHQ server, null means default configuration

**Returns** the form with the given ID as JSON

### retrieveFormJson

`String retrieveFormJson (String id)`

Same as `retrieveFormJson` but uses default Commcare configuration.

### retrieveFormList

`CommcareFormList retrieveFormList (FormListRequest request)`

Retrieves a list of forms from Commcare HQ along with metadata. Can be used for browsing through forms.

#### Parameters

- **request** – the request specifying the details of the query

**Returns** the form list received from Commcare HQ

### retrieveFormList

`CommcareFormList retrieveFormList (FormListRequest request, String configName)`

Retrieves a list of forms from Commcare HQ along with metadata. Can be used for browsing through forms.

#### Parameters

- **request** – the request specifying the details of the query
- **configName** – the name of the configuration used for connecting to the CommcareHQ server, null means default configuration

**Returns** the form list received from Commcare HQ

## CommcareLocationService

public interface **CommcareLocationService**

A service to perform queries against CommCareHQ location APIs.

### Methods

#### getCommcareLocationById

`CommcareLocation getCommcareLocationById (String id, String configName)`

Retrieves the CommCare location, based on the id and provided configuration name. If name is not set, the default configuration will be used.

#### Parameters

- **id** – id of the location
- **configName** – name of the configuration

**Returns** location, retrieved from the CommCare server

#### getCommcareLocationById

`CommcareLocation getCommcareLocationById (String id)`

Retrieves the CommCare location, based on the id. It uses default configuration.

#### Parameters

- **id** – id of the location

**Returns** location, retrieved from the CommCare server

**getCommcareLocations**

`List<CommcareLocation>` **getCommcareLocations** (*Integer* `pageSize`, *Integer* `pageNumber`, *String* `configName`)

Retrieves a list of locations, based on provided parameters and configuration name.

**Parameters**

- **pageSize** – amount of instances to fetch per page
- **pageNumber** – the page number
- **configName** – name of the configuration

**Returns** a list of loctions, retrieved from the CommCare server

**getCommcareLocations**

`List<CommcareLocation>` **getCommcareLocations** (*Integer* `pageSize`, *Integer* `pageNumber`)

Retrieves a list of locations, based on provided parameters. It uses default configuration.

**Parameters**

- **pageSize** – amount of instances to fetch per page
- **pageNumber** – the page number

**Returns** a list of loctions, retrieved from the CommCare server

**CommcareSchemaService**

public interface **CommcareSchemaService**

The `CommcareSchemaService` is responsible for retrieving CommCare schemas from MOTECH database. The important point is that implementers should NOT query CommCareHQ for data, but rather use current, local copy of the data.

**Methods****getAllCaseTypes**

`Map<String, Set<String>>` **getAllCaseTypes** (*String* `configName`)

Retrieves case types and fields that are assigned to the case type.

**Returns** Map of case types. Key represents a case type, while a value is a collection of fields assigned to the case type.

**getAllCaseTypes**

`Map<String, Set<String>>` **getAllCaseTypes** ()

Same as `getAllCaseTypes` but uses default CommcareHQ configuration.

**getAllFormSchemas**

`List<FormSchemaJson>` **getAllFormSchemas** (*String* `configName`)

Retrieves form schemas for all modules from MOTECH database for given configuration..

**Returns** List of all from schemas

**getAllFormSchemas**

List<FormSchemaJson> **getAllFormSchemas** ()

Same as `getAllFormSchemas` but uses default CommcareHQ configuration.

**CommcareUserService**

public interface **CommcareUserService**

A service to perform queries against CommCareHQ's user APIs. CommCareHQ exposes one user API, therefore getting a user by Id simply filters the list of all users returned by CommCareHQ.

**Methods****getCommcareUserById**

CommcareUser **getCommcareUserById** (String id, String configName)

Queries CommCareHQ for a list of all users on the configured domain and filters the list to return the user that matches the provided id.

**Parameters**

- **id** – the id of the user to retrieve.
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the CommcareUser object representing the information about the user from CommCareHQ, or null if that user did not exist on that domain.

**getCommcareUserById**

CommcareUser **getCommcareUserById** (String id)

Same as `getCommcareUserById` but uses default Commcare configuration.

**getCommcareUsers**

List<CommcareUser> **getCommcareUsers** (Integer pageSize, Integer pageNumber, String configName)

Queries CommCareHQ for a list of users(located on the given page) on the configured domain.

**Parameters**

- **pageSize** – the size of the page
- **pageNumber** – the number of the page
- **configName** – the name of the configuration used for connecting to CommcareHQ, null means default configuration

**Returns** the list of CommcareUsers that represent the information about each user located on the given page from CommCareHQ

**getCommcareUsers**

List<CommcareUser> **getCommcareUsers** (Integer pageSize, Integer pageNumber)

Same as `getCommcareUsers` but uses default Commcare configuration.



## 19.14.27 org.motechproject.commcare.tasks

### CommcareTasksNotifier

public class **CommcareTasksNotifier**

The `CommcareTasksNotifier` class is responsible for communication with the tasks module. It sends `org.motechproject.tasks.contract.ChannelRequests` in order to create or update tasks channel for the Commcare module. If the tasks module is not present, no updates will be sent.

#### Constructors

##### CommcareTasksNotifier

```
public CommcareTasksNotifier (BundleContext bundleContext, CommcareDataProvider dataProvider,  
                               CommcareConfigService configService, CommcareSchemaService  
                               schemaService)
```

#### Methods

##### updateTasksInfo

```
public void updateTasksInfo ()
```

### TasksChannelServiceInstance

public class **TasksChannelServiceInstance**

The `TasksChannelServiceInstance` class, holds Tasks-related beans. The reason this class must exist is because no Spring beans should be initialised if the Tasks module is not present. Therefore, we should first make sure that the Tasks module is present and only then we use Tasks classes, to avoid `NoClassDefFoundError`.

#### Constructors

##### TasksChannelServiceInstance

```
public TasksChannelServiceInstance (Object service, ChannelRequestBuilder builder)
```

#### Methods

##### updateTaskChannel

```
public void updateTaskChannel ()
```

Registers or updates the task channel if it contains any task triggers, if it doesn't the channel will be unregistered.

### VelocityResourceLoader

public class **VelocityResourceLoader** extends `ClasspathResourceLoader`

Loads resources for velocity from the CommCare bundle.

#### Methods

##### getResourceStream

```
public InputStream getResourceStream (String name)
```

## 19.14.28 org.motechproject.csd.constants

### CSDConstants

public final class **CSDConstants**

#### Fields

##### CSD\_CLASSES

public static final `List<Class<?>>` **CSD\_CLASSES**

##### CSD\_SCHEMA

public static final `URL` **CSD\_SCHEMA**

##### HAS\_MANAGE\_CSD\_PERMISSION

public static final `String` **HAS\_MANAGE\_CSD\_PERMISSION**

##### MANAGE\_CSD

public static final `String` **MANAGE\_CSD**

### CSDEventKeys

public final class **CSDEventKeys**

#### Fields

##### CONSUME\_XML\_EVENT\_BASE

public static final `String` **CONSUME\_XML\_EVENT\_BASE**

##### CONSUME\_XML\_EVENT\_WILDCARD

public static final `String` **CONSUME\_XML\_EVENT\_WILDCARD**

##### CSD\_TASK\_REST\_UPDATE

public static final `String` **CSD\_TASK\_REST\_UPDATE**

##### CSD\_TASK\_SOAP\_UPDATE

public static final `String` **CSD\_TASK\_SOAP\_UPDATE**

##### CSD\_UPDATE\_BASE

public static final `String` **CSD\_UPDATE\_BASE**

##### CSD\_UPDATE\_CUSTOM

public static final `String` **CSD\_UPDATE\_CUSTOM**

**CSD\_UPDATE\_SCHEDULED**

public static final [String](#) **CSD\_UPDATE\_SCHEDULED**

**CSD\_UPDATE\_TASK**

public static final [String](#) **CSD\_UPDATE\_TASK**

**LAST\_MODIFIED**

public static final [String](#) **LAST\_MODIFIED**

**UPDATE\_DATE**

public static final [String](#) **UPDATE\_DATE**

**XML\_URL**

public static final [String](#) **XML\_URL**

## 19.14.29 org.motechproject.csd.domain

**AbstractID**

public abstract class **AbstractID**

**Methods****getCreationDate**

public [DateTime](#) **getCreationDate** ()

**getId**

public [Long](#) **getId** ()

**getModificationDate**

public [DateTime](#) **getModificationDate** ()

**setCreationDate**

public void **setCreationDate** ([DateTime](#) *creationDate*)

**setId**

public void **setId** ([Long](#) *id*)

**setModificationDate**

public void **setModificationDate** ([DateTime](#) *modificationDate*)

## AbstractUniqueID

public abstract class **AbstractUniqueID** extends [AbstractID](#)

Java class for uniqueID complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="uniqueID">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <attribute name="entityID" use="required" type="{http://www.w3.org/2001/XMLSchema}string" />
    </restriction>
  </complexContent>
</complexType>
```

## Methods

### equals

public boolean **equals** ([Object o](#))

### getEntityID

public [String](#) **getEntityID** ()

### hashCode

public int **hashCode** ()

### setEntityID

public void **setEntityID** ([String entityID](#))

#### Parameters

- **entityID** – Should be a valid UUID represented as an URN. Example: “urn:uuid:53347B2E-185E-4BC3-BCDA-7FAB5D521FE7”

### toString

public [String](#) **toString** ()

## Address

public class **Address** extends [AbstractID](#)

Java class for address complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="address">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="addressLine" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="<http://www.w3.org/2001/XMLSchema>string">
                <attribute name="component" type="{http://www.w3.org/2001/XMLSchema}string" />
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

```
        </extension>
      </simpleContent>
    </complexType>
  </element>
</sequence>
<attribute name="type" type="{http://www.w3.org/2001/XMLSchema}anySimpleType" />
</restriction>
</complexContent>
</complexType>
```

## Constructors

### Address

public **Address** ()

### Address

public **Address** (*String type*)

### Address

public **Address** (*Set<AddressLine> addressLines, String type*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getAddressLines

public *Set<AddressLine>* **getAddressLines** ()

### getType

public *String* **getType** ()

### hashCode

public int **hashCode** ()

### setAddressLines

public void **setAddressLines** (*Set<AddressLine> addressLines*)

### setType

public void **setType** (*String type*)

### toString

public *String* **toString** ()

## AddressLine

public class **AddressLine** extends [AbstractID](#)

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <simpleContent>
    <extension base="<http://www.w3.org/2001/XMLSchema>string">
      <attribute name="component" type="{http://www.w3.org/2001/XMLSchema}string" />
    </extension>
  </simpleContent>
</complexType>
```

## Constructors

### AddressLine

public **AddressLine** ()

### AddressLine

public **AddressLine** (*String value*)

### AddressLine

public **AddressLine** (*String value*, *String component*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getComponent

public *String* **getComponent** ()

### getValue

public *String* **getValue** ()

### hashCode

public int **hashCode** ()

### setComponent

public void **setComponent** (*String component*)

### setValue

public void **setValue** (*String value*)

### toString

public *String* **toString** ()

## BaseMainEntity

public abstract class **BaseMainEntity** extends [AbstractID](#)

### Methods

#### **equals**

public boolean **equals** ([Object](#) *o*)

#### **getEntityID**

public [String](#) **getEntityID** ()

#### **getRecord**

public [Record](#) **getRecord** ()

#### **getSourceDirectory**

public [String](#) **getSourceDirectory** ()

#### **getStatus**

public [String](#) **getStatus** ()

#### **hashCode**

public int **hashCode** ()

#### **setEntityID**

public void **setEntityID** ([String](#) *entityID*)

#### **setRecord**

public void **setRecord** ([Record](#) *record*)

#### **setSourceDirectory**

public void **setSourceDirectory** ([String](#) *sourceDirectory*)

#### **setStatus**

public void **setStatus** ([String](#) *status*)

#### **toString**

public [String](#) **toString** ()

## CSD

public class **CSD**

Java class for CSD complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="CSD">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="organizationDirectory">
          <complexType>
            <complexContent>
              <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
                <sequence>
                  <element name="organization" type="{urn:ihe:iti:csd:2013}organization" maxOccurs="unbounded"/>
                </sequence>
              </restriction>
            </complexContent>
          </complexType>
        </element>
        <element name="serviceDirectory">
          <complexType>
            <complexContent>
              <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
                <sequence>
                  <element name="service" type="{urn:ihe:iti:csd:2013}service" maxOccurs="unbounded"/>
                </sequence>
              </restriction>
            </complexContent>
          </complexType>
        </element>
        <element name="facilityDirectory">
          <complexType>
            <complexContent>
              <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
                <sequence>
                  <element name="facility" type="{urn:ihe:iti:csd:2013}facility" maxOccurs="unbounded"/>
                </sequence>
              </restriction>
            </complexContent>
          </complexType>
        </element>
        <element name="providerDirectory">
          <complexType>
            <complexContent>
              <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
                <sequence>
                  <element name="provider" type="{urn:ihe:iti:csd:2013}provider" maxOccurs="unbounded"/>
                </sequence>
              </restriction>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

## Constructors

### CSD

public CSD ()



**CSD**

public **CSD** ([OrganizationDirectory](#) *organizationDirectory*, [ServiceDirectory](#) *serviceDirectory*, [FacilityDirectory](#) *facilityDirectory*, [ProviderDirectory](#) *providerDirectory*)

**Methods****equals**

public boolean **equals** ([Object](#) *o*)

**getFacilityDirectory**

public [FacilityDirectory](#) **getFacilityDirectory** ()

**getOrganizationDirectory**

public [OrganizationDirectory](#) **getOrganizationDirectory** ()

**getProviderDirectory**

public [ProviderDirectory](#) **getProviderDirectory** ()

**getServiceDirectory**

public [ServiceDirectory](#) **getServiceDirectory** ()

**hashCode**

public int **hashCode** ()

**setFacilityDirectory**

public void **setFacilityDirectory** ([FacilityDirectory](#) *facilityDirectory*)

**setOrganizationDirectory**

public void **setOrganizationDirectory** ([OrganizationDirectory](#) *organizationDirectory*)

**setProviderDirectory**

public void **setProviderDirectory** ([ProviderDirectory](#) *providerDirectory*)

**setServiceDirectory**

public void **setServiceDirectory** ([ServiceDirectory](#) *serviceDirectory*)

**toString**

public [String](#) **toString** ()

**CodedType**

public class **CodedType** extends [AbstractID](#)

The text content of this element may optionally be used to provide translations to the local language of a coded value.

Java class for codedtype complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="codedtype">
  <simpleContent>
    <extension base="<http://www.w3.org/2001/XMLSchema>string">
      <attribute name="code" use="required" type="{http://www.w3.org/2001/XMLSchema}anySimpleType" />
      <attribute name="codingScheme" use="required" type="{http://www.w3.org/2001/XMLSchema}anyURI" />
    </extension>
  </simpleContent>
</complexType>
```

## Constructors

### CodedType

public **CodedType** ()

### CodedType

public **CodedType** (*String code*, *String codingScheme*)

### CodedType

public **CodedType** (*String value*, *String code*, *String codingScheme*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getCode

public *String* **getCode** ()

### getCodingScheme

public *String* **getCodingScheme** ()

### getValue

public *String* **getValue** ()

### hashCode

public int **hashCode** ()

### setCode

public void **setCode** (*String code*)

### setCodingScheme

public void **setCodingScheme** (*String codingScheme*)

### setValue

public void **setValue** (*String value*)

**toString**

```
public String toString ()
```

**CommunicationProtocol**

```
public enum CommunicationProtocol
```

**Enum Constants****REST**

```
public static final CommunicationProtocol REST
```

**SOAP**

```
public static final CommunicationProtocol SOAP
```

**Config**

```
public class Config
```

**Fields****DATE\_TIME\_PICKER\_FORMAT**

```
public static final String DATE_TIME_PICKER_FORMAT
```

**Constructors****Config**

```
public Config ()
```

**Config**

```
public Config (boolean schedulerEnabled, String timePeriod, String timePeriodMultiplier, String startDate, String xmlUrl)
```

**Methods****equals**

```
public boolean equals (Object o)
```

**getCommunicationProtocol**

```
public CommunicationProtocol getCommunicationProtocol ()
```

**getLastModified**

```
public String getLastModified ()
```

**getPeriodDays**

```
public int getPeriodDays ()
```

**getPeriodHours**

public int **getPeriodHours** ()

**getPeriodMinutes**

public int **getPeriodMinutes** ()

**getStartDate**

public [String](#) **getStartDate** ()

**getTimePeriod**

public [String](#) **getTimePeriod** ()

**getTimePeriodMultiplier**

public [String](#) **getTimePeriodMultiplier** ()

**getXmlUrl**

public [String](#) **getXmlUrl** ()

**hashCode**

public int **hashCode** ()

**isSchedulerEnabled**

public boolean **isSchedulerEnabled** ()

**setCommunicationProtocol**

public void **setCommunicationProtocol** ([CommunicationProtocol](#) *communicationProtocol*)

**setLastModified**

public void **setLastModified** ([String](#) *lastModified*)

**setSchedulerEnabled**

public void **setSchedulerEnabled** (boolean *schedulerEnabled*)

**setStartDate**

public void **setStartDate** ([String](#) *startDate*)

**setTimePeriod**

public void **setTimePeriod** ([String](#) *timePeriod*)

**setTimePeriodMultiplier**

public void **setTimePeriodMultiplier** ([String](#) *timePeriodMultiplier*)

**setXmlUrl**

public void **setXmlUrl** ([String](#) *xmlUrl*)

## ContactPoint

public class **ContactPoint** extends [AbstractID](#)

Java class for contactPoint complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="contactPoint">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="codedType" type="{urn:ihe:iti:csd:2013}codedtype"/>
        <element name="equipment" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        <element name="purpose" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        <element name="certificate" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

## Constructors

### ContactPoint

public **ContactPoint** ()

### ContactPoint

public **ContactPoint** ([CodedType](#) codedType)

### ContactPoint

public **ContactPoint** ([CodedType](#) codedType, [String](#) equipment, [String](#) purpose, [String](#) certificate)

## Methods

### equals

public boolean **equals** ([Object](#) o)

### getCertificate

public [String](#) **getCertificate** ()

### getCodedType

public [CodedType](#) **getCodedType** ()

### getEquipment

public [String](#) **getEquipment** ()

### getPurpose

public [String](#) **getPurpose** ()

**hashCode**

```
public int hashCode ()
```

**setCertificate**

```
public void setCertificate (String certificate)
```

**setCodedType**

```
public void setCodedType (CodedType codedType)
```

**setEquipment**

```
public void setEquipment (String equipment)
```

**setPurpose**

```
public void setPurpose (String purpose)
```

**toString**

```
public String toString ()
```

**Credential**

```
public class Credential extends AbstractID
```

Java class for credential complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="credential">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="codedType" type="{urn:ihe:iti:csd:2013}codedtype"/>
        <element name="number" type="{http://www.w3.org/2001/XMLSchema}string"/>
        <element name="issuingAuthority" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        <element name="credentialIssueDate" type="{http://www.w3.org/2001/XMLSchema}date" minOccurs="0"/>
        <element name="credentialRenewalDate" type="{http://www.w3.org/2001/XMLSchema}date" minOccurs="0"/>
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

**Constructors****Credential**

```
public Credential ()
```

**Credential**

```
public Credential (CodedType codedType, String number)
```

**Credential**

public **Credential** (*CodedType codedType, String number, String issuingAuthority, DateTime credentialIssueDate, DateTime credentialRenewalDate, Set<Extension> extensions*)

**Methods****equals**

public boolean **equals** (*Object o*)

**getCodedType**

public *CodedType* **getCodedType** ()

**getCredentialIssueDate**

public *DateTime* **getCredentialIssueDate** ()

**getCredentialRenewalDate**

public *DateTime* **getCredentialRenewalDate** ()

**getExtensions**

public *Set<Extension>* **getExtensions** ()

**getIssuingAuthority**

public *String* **getIssuingAuthority** ()

**getNumber**

public *String* **getNumber** ()

**hashCode**

public int **hashCode** ()

**setCodedType**

public void **setCodedType** (*CodedType codedType*)

**setCredentialIssueDate**

public void **setCredentialIssueDate** (*DateTime credentialIssueDate*)

**setCredentialRenewalDate**

public void **setCredentialRenewalDate** (*DateTime credentialRenewalDate*)

**setExtensions**

public void **setExtensions** (*Set<Extension> extensions*)

**setIssuingAuthority**

public void **setIssuingAuthority** (*String issuingAuthority*)

**setNumber**

public void **setNumber** (*String number*)

**toString**

public *String* **toString** ()

**DayOfTheWeek**

public enum **DayOfTheWeek**

**Enum Constants****FRIDAY**

public static final *DayOfTheWeek* **FRIDAY**

**MONDAY**

public static final *DayOfTheWeek* **MONDAY**

**SATURDAY**

public static final *DayOfTheWeek* **SATURDAY**

**SUNDAY**

public static final *DayOfTheWeek* **SUNDAY**

**THURSDAY**

public static final *DayOfTheWeek* **THURSDAY**

**TUESDAY**

public static final *DayOfTheWeek* **TUESDAY**

**WEDNESDAY**

public static final *DayOfTheWeek* **WEDNESDAY**

**Extension**

public class **Extension** extends *AbstractID*

Java class for extension complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="extension">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <attribute name="type" use="required" type="{http://www.w3.org/2001/XMLSchema}anySimpleT" />
      <attribute name="urn" use="required" type="{http://www.w3.org/2001/XMLSchema}string" />
    </restriction>
  </complexContent>
</complexType>
```



## Constructors

### Extension

public **Extension** ()

### Extension

public **Extension** (String *type*, String *urn*)

## Methods

### equals

public boolean **equals** (Object *o*)

### getType

public String **getType** ()

### getUrn

public String **getUrn** ()

### hashCode

public int **hashCode** ()

### setType

public void **setType** (String *type*)

### setUrn

public void **setUrn** (String *urn*)

### toString

public String **toString** ()

## Facility

public class **Facility** extends [BaseMainEntity](#)

Java class for facility complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="facility">
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="otherID" type="{urn:ihe:iti:csd:2013}otherID" maxOccurs="unbounded" minOccurs="0"/>
        <element name="codedType" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="0"/>
        <element name="primaryName" type="{http://www.w3.org/2001/XMLSchema}string" maxOccurs="unbounded" minOccurs="0"/>
        <element name="otherName" type="{http://www.w3.org/2001/XMLSchema}string" maxOccurs="unbounded" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
        <attribute ref="{http://www.w3.org/XML/1998/namespace}lang"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="address" type="{urn:ihe:iti:csd:2013}address" maxOccurs="unbounded" minOccurs="1"/>
<element name="contact" maxOccurs="unbounded" minOccurs="0">
  <complexType>
    <complexContent>
      <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
        <choice>
          <element name="provider" type="{urn:ihe:iti:csd:2013}uniqueID"/>
          <element name="person" type="{urn:ihe:iti:csd:2013}person"/>
        </choice>
      </restriction>
    </complexContent>
  </complexType>
</element>
<element name="geocode" type="{urn:ihe:iti:csd:2013}geocode" minOccurs="0"/>
<element name="language" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="1"/>
<element name="contactPoint" type="{urn:ihe:iti:csd:2013}contactPoint" maxOccurs="unbounded" minOccurs="1"/>
<element name="organizations" minOccurs="0">
  <complexType>
    <complexContent>
      <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="organization" maxOccurs="unbounded">
            <complexType>
              <complexContent>
                <extension base="{urn:ihe:iti:csd:2013}uniqueID">
                  <sequence>
                    <element name="service" maxOccurs="unbounded" minOccurs="0">
                      <complexType>
                        <complexContent>
                          <extension base="{urn:ihe:iti:csd:2013}uniqueID">
                            <sequence>
                              <element name="name" type="{urn:ihe:iti:csd:2013}name" maxOccurs="unbounded" minOccurs="1"/>
                              <element name="language" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="1"/>
                              <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="1"/>
                              <element name="freeBusyURI" type="{http://www.w3.org/2001/XMLSchema}anyURI" maxOccurs="unbounded" minOccurs="1"/>
                              <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
                            </sequence>
                          </extension>
                        </complexContent>
                      </complexType>
                    </element>
                    <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
                  </sequence>
                </extension>
              </complexContent>
            </complexType>
          </element>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
</element>
<element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="1"/>
```

```
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" m
        <element name="record" type="{urn:ihe:iti:csd:2013}record"/>
    </sequence>
</extension>
</complexContent>
</complexType>
```

## Constructors

### Facility

public **Facility** ()

### Facility

public **Facility** (String *entityID*, Set<CodedType> *codedTypes*, Record *record*, String *primaryName*)

### Facility

public **Facility** (String *entityID*, Set<CodedType> *codedTypes*, Set<Extension> *extensions*, Record *record*, Set<OtherID> *otherIDs*, String *primaryName*, Set<OtherName> *otherNames*, Set<Address> *addresses*, Set<OrganizationContact> *contacts*, Set<CodedType> *languages*, Set<ContactPoint> *contactPoints*, Geocode *geocode*, Set<FacilityOrganization> *facilityOrganizations*, Set<OperatingHours> *operatingHours*)

## Methods

### equals

public boolean **equals** (Object *o*)

### getAddresses

public Set<Address> **getAddresses** ()

### getCodedTypes

public Set<CodedType> **getCodedTypes** ()

### getContactPoints

public Set<ContactPoint> **getContactPoints** ()

### getContacts

public Set<OrganizationContact> **getContacts** ()

### getExtensions

public Set<Extension> **getExtensions** ()

### getFacilityOrganizationSet

public Set<FacilityOrganization> **getFacilityOrganizationSet** ()

**getFacilityOrganizations**

public [Set](#)<[FacilityOrganization](#)> **getFacilityOrganizations** ()

**getGeocode**

public [Geocode](#) **getGeocode** ()

**getLanguages**

public [Set](#)<[CodedType](#)> **getLanguages** ()

**getOperatingHours**

public [Set](#)<[OperatingHours](#)> **getOperatingHours** ()

**getOtherIDs**

public [Set](#)<[OtherID](#)> **getOtherIDs** ()

**getOtherNames**

public [Set](#)<[OtherName](#)> **getOtherNames** ()

**getPrimaryName**

public [String](#) **getPrimaryName** ()

**hashCode**

public int **hashCode** ()

**setAddresses**

public void **setAddresses** ([Set](#)<[Address](#)> *addresses*)

**setCodedTypes**

public void **setCodedTypes** ([Set](#)<[CodedType](#)> *codedTypes*)

**setContactPoints**

public void **setContactPoints** ([Set](#)<[ContactPoint](#)> *contactPoints*)

**setContacts**

public void **setContacts** ([Set](#)<[OrganizationContact](#)> *contacts*)

**setExtensions**

public void **setExtensions** ([Set](#)<[Extension](#)> *extensions*)

**setFacilityOrganizationSet**

public void **setFacilityOrganizationSet** ([Set](#)<[FacilityOrganization](#)> *facilityOrganizationSet*)

**setFacilityOrganizations**

public void **setFacilityOrganizations** ([Set](#)<[FacilityOrganization](#)> *facilityOrganizations*)

**setGeocode**

public void **setGeocode** ([Geocode](#) *geocode*)

**setLanguages**

public void **setLanguages** ([Set](#)<[CodedType](#)> *languages*)

**setOperatingHours**

public void **setOperatingHours** ([Set](#)<[OperatingHours](#)> *operatingHours*)

**setOtherIDs**

public void **setOtherIDs** ([Set](#)<[OtherID](#)> *otherIDs*)

**setOtherNames**

public void **setOtherNames** ([Set](#)<[OtherName](#)> *otherNames*)

**setPrimaryName**

public void **setPrimaryName** ([String](#) *primaryName*)

**toString**

public [String](#) **toString** ()

**FacilityDirectory**

public class **FacilityDirectory**

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="facility" type="{urn:ihe:iti:csd:2013}facility" maxOccurs="unbounded" minOccurs="1"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

**Constructors****FacilityDirectory**

public **FacilityDirectory** ()

**FacilityDirectory**

public **FacilityDirectory** ([Set](#)<[Facility](#)> *facilities*)

## Methods

### **equals**

public boolean **equals** (*Object o*)

### **getFacilities**

public *Set*<*Facility*> **getFacilities** ()

### **hashCode**

public int **hashCode** ()

### **setFacilities**

public void **setFacilities** (*Set*<*Facility*> *facilities*)

### **toString**

public *String* **toString** ()

## FacilityOrganization

public class **FacilityOrganization** extends *AbstractUniqueID*

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="service" maxOccurs="unbounded" minOccurs="0">
          <complexType>
            <complexContent>
              <extension base="{urn:ihe:iti:csd:2013}uniqueID">
                <sequence>
                  <element name="name" type="{urn:ihe:iti:csd:2013}name" maxOccurs="unbounded" minOccurs="0"/>
                  <element name="language" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="0"/>
                  <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="0"/>
                  <element name="freeBusyURI" type="{http://www.w3.org/2001/XMLSchema}anyURI" minOccurs="0"/>
                  <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
                </sequence>
              </extension>
            </complexContent>
          </complexType>
        </element>
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

## Constructors

### FacilityOrganization

```
public FacilityOrganization ()
```

### **FacilityOrganization**

```
public FacilityOrganization (String entityID)
```

### **FacilityOrganization**

```
public FacilityOrganization (String entityID, Set<FacilityOrganizationService> services,  
                             Set<Extension> extensions)
```

### **Methods**

#### **equals**

```
public boolean equals (Object o)
```

#### **getExtensions**

```
public Set<Extension> getExtensions ()
```

#### **getServices**

```
public Set<FacilityOrganizationService> getServices ()
```

#### **hashCode**

```
public int hashCode ()
```

#### **setExtensions**

```
public void setExtensions (Set<Extension> extensions)
```

#### **setServices**

```
public void setServices (Set<FacilityOrganizationService> services)
```

#### **toString**

```
public String toString ()
```

### **FacilityOrganizationService**

```
public class FacilityOrganizationService extends AbstractUniqueID
```

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>  
  <complexContent>  
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">  
      <sequence>  
        <element name="name" type="{urn:ihe:iti:csd:2013}name" maxOccurs="unbounded" minOccurs="0"/>  
        <element name="language" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="0"/>  
        <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="0"/>  
        <element name="freeBusyURI" type="{http://www.w3.org/2001/XMLSchema}anyURI" minOccurs="0"/>  
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>  
      </sequence>  
    </extension>  
  </complexContent>  
</complexType>
```

```
        </extension>
    </complexContent>
</complexType>
```

## Constructors

### FacilityOrganizationService

```
public FacilityOrganizationService ()
```

### FacilityOrganizationService

```
public FacilityOrganizationService (String entityID)
```

### FacilityOrganizationService

```
public FacilityOrganizationService (String entityID, Set<Name> names, Set<CodedType> languages, Set<OperatingHours> operatingHours, String freeBusyURI, Set<Extension> extensions)
```

## Methods

### equals

```
public boolean equals (Object o)
```

### getExtensions

```
public Set<Extension> getExtensions ()
```

### getFreeBusyURI

```
public String getFreeBusyURI ()
```

### getLanguages

```
public Set<CodedType> getLanguages ()
```

### getNames

```
public Set<Name> getNames ()
```

### getOperatingHours

```
public Set<OperatingHours> getOperatingHours ()
```

### hashCode

```
public int hashCode ()
```

### setExtensions

```
public void setExtensions (Set<Extension> extensions)
```

### setFreeBusyURI

```
public void setFreeBusyURI (String freeBusyURI)
```



**setLanguages**

public void **setLanguages** ([Set](#)<[CodedType](#)> *languages*)

**setNames**

public void **setNames** ([Set](#)<[Name](#)> *names*)

**setOperatingHours**

public void **setOperatingHours** ([Set](#)<[OperatingHours](#)> *operatingHours*)

**toString**

public [String](#) **toString** ()

**Geocode**

public class **Geocode** extends [AbstractID](#)

Java class for geocode complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="geocode">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="latitude" type="{http://www.w3.org/2001/XMLSchema}decimal"/>
        <element name="longitude" type="{http://www.w3.org/2001/XMLSchema}decimal"/>
        <element name="altitude" type="{http://www.w3.org/2001/XMLSchema}decimal" minOccurs="0"/>
        <element name="coordinateSystem" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

**Constructors****Geocode**

public **Geocode** ()

**Geocode**

public **Geocode** (double *latitude*, double *longitude*)

**Geocode**

public **Geocode** (double *latitude*, double *longitude*, double *altitude*, [String](#) *coordinateSystem*)

**Methods****equals**

public boolean **equals** ([Object](#) *o*)

**getAltitude**

public double **getAltitude** ()

**getCoordinateSystem**

public [String](#) **getCoordinateSystem** ()

**getLatitude**

public double **getLatitude** ()

**getLongitude**

public double **getLongitude** ()

**hashCode**

public int **hashCode** ()

**setAltitude**

public void **setAltitude** (double *altitude*)

**setCoordinateSystem**

public void **setCoordinateSystem** ([String](#) *coordinateSystem*)

**setLatitude**

public void **setLatitude** (double *latitude*)

**setLongitude**

public void **setLongitude** (double *longitude*)

**toString**

public [String](#) **toString** ()

**GetModificationsRequest**

public class **GetModificationsRequest**

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="lastModified" type="{http://www.w3.org/2001/XMLSchema}dateTime"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

## Methods

### getLastModified

public [XMLGregorianCalendar](#) **getLastModified**()

Gets the value of the lastModified property.

**Returns** possible object is `javax.xml.datatype.XMLGregorianCalendar`

### setLastModified

public void **setLastModified**([XMLGregorianCalendar](#) *value*)

Sets the value of the lastModified property.

#### Parameters

- **value** – allowed object is `javax.xml.datatype.XMLGregorianCalendar`

## GetModificationsResponse

public class **GetModificationsResponse**

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element ref="{urn:ihe:iti:csd:2013}CSD"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

## Methods

### getCSD

public [CSD](#) **getCSD**()

Gets the value of the csd property.

**Returns** possible object is [CSD](#)

### setCSD

public void **setCSD**([CSD](#) *value*)

Sets the value of the csd property.

#### Parameters

- **value** – allowed object is [CSD](#)

## Name

public class **Name** extends [AbstractID](#)

Java class for name complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="name">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="commonName" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="<http://www.w3.org/2001/XMLSchema>string">
                </extension>
              </simpleContent>
            </complexType>
          </element>
          <element name="honorific" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="forename" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="otherName" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" m
          <element name="surname" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
          <element name="suffix" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
```

## Constructors

### Name

public **Name** ()

### Name

public **Name** (*List<String> commonNames*)

### Name

public **Name** (*List<String> commonNames, String honorific, String forename, Set<CodedType> otherNames, String surname, String suffix*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getCommonNames

public *List<String>* **getCommonNames** ()

### getForename

public *String* **getForename** ()

### getHonorific

public *String* **getHonorific** ()

**getOtherNames**

```
public Set<CodedType> getOtherNames ()
```

**getSuffix**

```
public String getSuffix ()
```

**getSurname**

```
public String getSurname ()
```

**hashCode**

```
public int hashCode ()
```

**setCommonNames**

```
public void setCommonNames (List<String> commonNames)
```

**setForename**

```
public void setForename (String forename)
```

**setHonorific**

```
public void setHonorific (String honorific)
```

**setOtherNames**

```
public void setOtherNames (Set<CodedType> otherNames)
```

**setSuffix**

```
public void setSuffix (String suffix)
```

**setSurname**

```
public void setSurname (String surname)
```

**toString**

```
public String toString ()
```

## OperatingHours

```
public class OperatingHours extends AbstractID
```

Java class for operatingHours complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="operatingHours">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="openFlag" type="{http://www.w3.org/2001/XMLSchema}boolean"/>
        <element name="dayOfTheWeek" type="{http://www.w3.org/2001/XMLSchema}integer" maxOccurs="0"/>
        <element name="beginningHour" type="{http://www.w3.org/2001/XMLSchema}time" minOccurs="0"/>
        <element name="endingHour" type="{http://www.w3.org/2001/XMLSchema}time" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

```
        <element name="beginEffectiveDate" type="{http://www.w3.org/2001/XMLSchema}date"/>
        <element name="endEffectiveDate" type="{http://www.w3.org/2001/XMLSchema}date" minOccurs="1"/>
    </sequence>
</restriction>
</complexContent>
</complexType>
```

## Constructors

### OperatingHours

public **OperatingHours** ()

### OperatingHours

public **OperatingHours** (boolean *openFlag*, [DateTime](#) *beginEffectiveDate*)

### OperatingHours

public **OperatingHours** (boolean *openFlag*, [List](#)<[DayOfTheWeek](#)> *daysOfTheWeek*, [DateTime](#) *beginningHour*, [DateTime](#) *endingHour*, [DateTime](#) *beginEffectiveDate*, [DateTime](#) *endEffectiveDate*)

## Methods

### equals

public boolean **equals** ([Object](#) *o*)

### getBeginEffectiveDate

public [DateTime](#) **getBeginEffectiveDate** ()

### getBeginningHour

public [DateTime](#) **getBeginningHour** ()

### getDaysOfTheWeek

public [List](#)<[DayOfTheWeek](#)> **getDaysOfTheWeek** ()

### getEndEffectiveDate

public [DateTime](#) **getEndEffectiveDate** ()

### getEndingHour

public [DateTime](#) **getEndingHour** ()

### hashCode

public int **hashCode** ()

### isOpenFlag

public boolean **isOpenFlag** ()

**setBeginEffectiveDate**

public void **setBeginEffectiveDate** (*DateTime beginEffectiveDate*)

**setBeginningHour**

public void **setBeginningHour** (*DateTime beginningHour*)

**setDaysOfTheWeek**

public void **setDaysOfTheWeek** (*List<DayOfTheWeek> daysOfTheWeek*)

**setEndEffectiveDate**

public void **setEndEffectiveDate** (*DateTime endEffectiveDate*)

**setEndingHour**

public void **setEndingHour** (*DateTime endingHour*)

**setOpenFlag**

public void **setOpenFlag** (boolean *openFlag*)

**toString**

public *String* **toString** ()

## Organization

public class **Organization** extends *BaseMainEntity*

Java class for organization complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="organization">
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="otherID" type="{urn:ihe:iti:csd:2013}otherID" maxOccurs="unbounded" minOccurs="0"/>
        <element name="codedType" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="0"/>
        <element name="primaryName" type="{http://www.w3.org/2001/XMLSchema}string"/>
        <element name="otherName" maxOccurs="unbounded" minOccurs="0">
          <complexType>
            <simpleContent>
              <extension base="<http://www.w3.org/2001/XMLSchema>string">
                <attribute ref="{http://www.w3.org/XML/1998/namespace}lang"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
        <element name="address" type="{urn:ihe:iti:csd:2013}address" maxOccurs="unbounded" minOccurs="0"/>
        <element name="contact" maxOccurs="unbounded" minOccurs="0">
          <complexType>
            <complexContent>
              <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
                <choice>
                  <element name="provider" type="{urn:ihe:iti:csd:2013}uniqueID"/>
                  <element name="person" type="{urn:ihe:iti:csd:2013}person"/>
                </choice>
              </restriction>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
        </choice>
        </restriction>
    </complexContent>
</complexType>
</element>
<element name="credential" type="{urn:ihe:iti:csd:2013}credential" maxOccurs="unbounded" minOccurs="1"/>
<element name="language" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="1"/>
<element name="specialization" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="1"/>
<element name="contactPoint" type="{urn:ihe:iti:csd:2013}contactPoint" maxOccurs="unbounded" minOccurs="1"/>
<element name="parent" type="{urn:ihe:iti:csd:2013}uniqueID" minOccurs="0"/>
<element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
<element name="record" type="{urn:ihe:iti:csd:2013}record"/>
</sequence>
</extension>
</complexContent>
</complexType>
```

## Constructors

### Organization

public **Organization** ()

### Organization

public **Organization** (*String entityID*, *Set<CodedType> codedTypes*, *String primaryName*, *Record record*)

### Organization

public **Organization** (*String entityID*, *Set<Credential> credentials*, *String parentOrganization*, *Set<CodedType> specializations*, *Set<OtherID> otherIDs*, *String primaryName*, *Set<OtherName> otherNames*, *Set<Address> addresses*, *Set<OrganizationContact> contacts*, *Set<CodedType> languages*, *Set<ContactPoint> contactPoints*, *Set<CodedType> codedTypes*, *Set<Extension> extensions*, *Record record*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getAddresses

public *Set<Address>* **getAddresses** ()

### getCodedTypes

public *Set<CodedType>* **getCodedTypes** ()

### getContactPoints

public *Set<ContactPoint>* **getContactPoints** ()

### getContacts

public *Set<OrganizationContact>* **getContacts** ()



**getCredentials**

```
public Set<Credential> getCredentials ()
```

**getExtensions**

```
public Set<Extension> getExtensions ()
```

**getLanguages**

```
public Set<CodedType> getLanguages ()
```

**getOtherIDs**

```
public Set<OtherID> getOtherIDs ()
```

**getOtherNames**

```
public Set<OtherName> getOtherNames ()
```

**getParent**

```
public UniqueID getParent ()
```

**getParentOrganization**

```
public String getParentOrganization ()
```

**getPrimaryName**

```
public String getPrimaryName ()
```

**getSpecializations**

```
public Set<CodedType> getSpecializations ()
```

**hashCode**

```
public int hashCode ()
```

**setAddresses**

```
public void setAddresses (Set<Address> addresses)
```

**setCodedTypes**

```
public void setCodedTypes (Set<CodedType> codedTypes)
```

**setContactPoints**

```
public void setContactPoints (Set<ContactPoint> contactPoints)
```

**setContacts**

```
public void setContacts (Set<OrganizationContact> contacts)
```

**setCredentials**

```
public void setCredentials (Set<Credential> credentials)
```

**setExtensions**

public void **setExtensions** ([Set<Extension>](#) *extensions*)

**setLanguages**

public void **setLanguages** ([Set<CodedType>](#) *languages*)

**setOtherIDs**

public void **setOtherIDs** ([Set<OtherID>](#) *otherIDs*)

**setOtherNames**

public void **setOtherNames** ([Set<OtherName>](#) *otherNames*)

**setParent**

public void **setParent** ([UniqueID](#) *parent*)

**setParentOrganization**

public void **setParentOrganization** ([String](#) *parentOrganization*)

**setPrimaryName**

public void **setPrimaryName** ([String](#) *primaryName*)

**setSpecializations**

public void **setSpecializations** ([Set<CodedType>](#) *specializations*)

**toString**

public [String](#) **toString** ()

**OrganizationContact**

public class **OrganizationContact** extends [AbstractID](#)

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <choice>
        <element name="provider" type="{urn:ihe:iti:csd:2013}uniqueID"/>
        <element name="person" type="{urn:ihe:iti:csd:2013}person"/>
      </choice>
    </restriction>
  </complexContent>
</complexType>
```

**Constructors****OrganizationContact**

public **OrganizationContact** ()

**OrganizationContact**

public **OrganizationContact** (*String providerEntityID*)

**OrganizationContact**

public **OrganizationContact** (*Person person*)

**Methods****equals**

public boolean **equals** (*Object o*)

**getContact**

public *Object* **getContact** ()

**getPerson**

public *Person* **getPerson** ()

**getProviderEntityID**

public *String* **getProviderEntityID** ()

**hashCode**

public int **hashCode** ()

**setContact**

public void **setContact** (*Object contact*)

**setPerson**

public void **setPerson** (*Person person*)

**setProviderEntityID**

public void **setProviderEntityID** (*String providerEntityID*)

**toString**

public *String* **toString** ()

**OrganizationDirectory**

public class **OrganizationDirectory**

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="organization" type="{urn:ihe:iti:csd:2013}organization" maxOccurs="unbound"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

```
        </sequence>
    </restriction>
</complexContent>
</complexType>
```

## Constructors

### OrganizationDirectory

public **OrganizationDirectory** ()

### OrganizationDirectory

public **OrganizationDirectory** ([Set<Organization>](#) organizations)

## Methods

### equals

public boolean **equals** ([Object o](#))

### getOrganizations

public [Set<Organization>](#) **getOrganizations** ()

### hashCode

public int **hashCode** ()

### setOrganizations

public void **setOrganizations** ([Set<Organization>](#) organizations)

### toString

public [String](#) **toString** ()

## OtherID

public class **OtherID** extends [AbstractID](#)

Java class for otherID complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="otherID">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <attribute name="code" use="required" type="{http://www.w3.org/2001/XMLSchema}string" />
      <attribute name="assigningAuthorityName" use="required" type="{http://www.w3.org/2001/XMLSchema}string" />
    </restriction>
  </complexContent>
</complexType>
```

## Constructors

### OtherID

public **OtherID** ()

### OtherID

public **OtherID** (*String code*)

### OtherID

public **OtherID** (*String code*, *String assigningAuthorityName*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getAssigningAuthorityName

public *String* **getAssigningAuthorityName** ()

### getCode

public *String* **getCode** ()

### hashCode

public int **hashCode** ()

### setAssigningAuthorityName

public void **setAssigningAuthorityName** (*String assigningAuthorityName*)

### setCode

public void **setCode** (*String code*)

### toString

public *String* **toString** ()

## OtherName

public class **OtherName** extends *AbstractID*

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <simpleContent>
    <extension base="<http://www.w3.org/2001/XMLSchema>string">
      <attribute ref="{http://www.w3.org/XML/1998/namespace}lang"/>
    </extension>
  </simpleContent>
</complexType>
```

## Constructors

### OtherName

public **OtherName** ()

### OtherName

public **OtherName** (*String value*)

### OtherName

public **OtherName** (*String value*, *String lang*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getLang

public *String* **getLang** ()

### getValue

public *String* **getValue** ()

### hashCode

public int **hashCode** ()

### setLang

public void **setLang** (*String lang*)

### setValue

public void **setValue** (*String value*)

### toString

public *String* **toString** ()

## Person

public class **Person** extends *AbstractID*

Java class for person complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="person">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="name" maxOccurs="unbounded">
          <complexType>
            <complexContent>
```

```
<extension base="{urn:ihe:iti:csd:2013}name">
  <attribute ref="{http://www.w3.org/XML/1998/namespace}lang"/>
</extension>
</complexContent>
</complexType>
</element>
<element name="contactPoint" type="{urn:ihe:iti:csd:2013}contactPoint" maxOccurs="unbound" minOccurs="1"/>
<element name="address" type="{urn:ihe:iti:csd:2013}address" maxOccurs="unbounded" minOccurs="1"/>
<element name="gender" type="{http://www.w3.org/2001/XMLSchema}string" minOccurs="0"/>
<element name="dateOfBirth" type="{http://www.w3.org/2001/XMLSchema}date" minOccurs="0"/>
</sequence>
</restriction>
</complexContent>
</complexType>
```

## Constructors

### Person

```
public Person ()
```

### Person

```
public Person (Set<PersonName> names)
```

### Person

```
public Person (Set<PersonName> names, Set<ContactPoint> contactPoints, Set<Address> addresses, String
gender, DateTime dateOfBirth)
```

## Methods

### equals

```
public boolean equals (Object o)
```

### getAddresses

```
public Set<Address> getAddresses ()
```

### getContactPoints

```
public Set<ContactPoint> getContactPoints ()
```

### getDateOfBirth

```
public DateTime getDateOfBirth ()
```

### getGender

```
public String getGender ()
```

### getNames

```
public Set<PersonName> getNames ()
```

**hashCode**

```
public int hashCode ()
```

**setAddresses**

```
public void setAddresses (Set<Address> addresses)
```

**setContactPoints**

```
public void setContactPoints (Set<ContactPoint> contactPoints)
```

**setDateOfBirth**

```
public void setDateOfBirth (DateTime dateOfBirth)
```

**setGender**

```
public void setGender (String gender)
```

**setNames**

```
public void setNames (Set<PersonName> names)
```

**toString**

```
public String toString ()
```

**PersonName**

```
public class PersonName extends AbstractID
```

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}name">
      <attribute ref="{http://www.w3.org/XML/1998/namespace}lang"/>
    </extension>
  </complexContent>
</complexType>
```

**Constructors****PersonName**

```
public PersonName ()
```

**PersonName**

```
public PersonName (List<String> commonNames)
```

**PersonName**

```
public PersonName (List<String> commonNames, String honorific, String forename, Set<CodedType> otherNames, String surname, String suffix, String lang)
```



## Methods

### **equals**

public boolean **equals** (*Object o*)

### **getCommonNames**

public *List<String>* **getCommonNames** ()

### **getForename**

public *String* **getForename** ()

### **getHonorific**

public *String* **getHonorific** ()

### **getLang**

public *String* **getLang** ()

### **getOtherNames**

public *Set<CodedType>* **getOtherNames** ()

### **getSuffix**

public *String* **getSuffix** ()

### **getSurname**

public *String* **getSurname** ()

### **hashCode**

public int **hashCode** ()

### **setCommonNames**

public void **setCommonNames** (*List<String> commonNames*)

### **setForename**

public void **setForename** (*String forename*)

### **setHonorific**

public void **setHonorific** (*String honorific*)

### **setLang**

public void **setLang** (*String lang*)

### **setOtherNames**

public void **setOtherNames** (*Set<CodedType> otherNames*)

**setSuffix**

```
public void setSuffix (String suffix)
```

**setSurname**

```
public void setSurname (String surname)
```

**toString**

```
public String toString ()
```

**Provider**

```
public class Provider extends BaseMainEntity
```

Java class for provider complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="provider">
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="otherID" type="{urn:ihe:iti:csd:2013}otherID" maxOccurs="unbounded" minOccurs="0"/>
        <element name="codedType" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="0"/>
        <element name="demographic" type="{urn:ihe:iti:csd:2013}person"/>
        <element name="language" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="0"/>
        <element name="organizations" minOccurs="0">
          <complexType>
            <complexContent>
              <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
                <sequence>
                  <element name="organization" maxOccurs="unbounded">
                    <complexType>
                      <complexContent>
                        <extension base="{urn:ihe:iti:csd:2013}uniqueID">
                          <sequence>
                            <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
                            <element name="address" type="{urn:ihe:iti:csd:2013}address" maxOccurs="unbounded" minOccurs="0"/>
                            <element name="contactPoint" type="{urn:ihe:iti:csd:2013}contactPoint" maxOccurs="unbounded" minOccurs="0"/>
                          </sequence>
                        </extension>
                      </complexContent>
                    </complexType>
                  </element>
                </sequence>
              </restriction>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="facilities" minOccurs="0">
  <complexType>
    <complexContent>
      <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
        <sequence>
          <element name="facility" maxOccurs="unbounded">
            <complexType>
              <complexContent>
                <extension base="{urn:ihe:iti:csd:2013}uniqueID">
```

```

<sequence>
  <element name="service" maxOccurs="unbounded" minOccurs="0">
    <complexType>
      <complexContent>
        <extension base="{urn:ihe:iti:csd:2013}uniqueID">
          <sequence>
            <element name="name" type="{urn:ihe:iti:csd:2013}name" maxOccurs="unbounded" minOccurs="0"/>
            <element name="organization" type="{urn:ihe:iti:csd:2013}organization" maxOccurs="unbounded" minOccurs="0"/>
            <element name="language" type="{urn:ihe:iti:csd:2013}codedType" maxOccurs="unbounded" minOccurs="0"/>
            <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="0"/>
            <element name="freeBusyURI" type="{http://www.w3.org/2001/XMLSchema#string}" maxOccurs="unbounded" minOccurs="0"/>
            <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </element>
  <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="0"/>
  <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
</sequence>
</extension>
</complexContent>
</complexType>
</element>
</restriction>
</complexContent>
</complexType>
</element>
<element name="credential" type="{urn:ihe:iti:csd:2013}credential" maxOccurs="unbounded" minOccurs="0"/>
<element name="specialty" type="{urn:ihe:iti:csd:2013}codedType" maxOccurs="unbounded" minOccurs="0"/>
<element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
<element name="record" type="{urn:ihe:iti:csd:2013}record"/>
</sequence>
</extension>
</complexContent>
</complexType>

```

## Constructors

### Provider

public **Provider** ()

### Provider

public **Provider** (String *entityID*, Set<CodedType> *codedTypes*, Record *record*, Person *demographic*)

### Provider

public **Provider** (String *entityID*, Set<CodedType> *codedTypes*, Set<Extension> *extensions*, Record *record*, Set<OtherID> *otherIDs*, Person *demographic*, Set<CodedType> *languages*, Set<ProviderOrganization> *providerOrganizations*, Set<ProviderFacility> *providerFacilities*, Set<Credential> *credentials*, Set<CodedType> *specialties*)

## Methods

**equals**

public boolean **equals** (*Object o*)

**getCodedTypes**

public *Set*<*CodedType*> **getCodedTypes** ()

**getCredentials**

public *Set*<*Credential*> **getCredentials** ()

**getDemographic**

public *Person* **getDemographic** ()

**getExtensions**

public *Set*<*Extension*> **getExtensions** ()

**getLanguages**

public *Set*<*CodedType*> **getLanguages** ()

**getOtherIDs**

public *Set*<*OtherID*> **getOtherIDs** ()

**getProviderFacilities**

public *Set*<*ProviderFacility*> **getProviderFacilities** ()

**getProviderFacilitySet**

public *Set*<*ProviderFacility*> **getProviderFacilitySet** ()

**getProviderOrganizationSet**

public *Set*<*ProviderOrganization*> **getProviderOrganizationSet** ()

**getProviderOrganizations**

public *Set*<*ProviderOrganization*> **getProviderOrganizations** ()

**getSpecialties**

public *Set*<*CodedType*> **getSpecialties** ()

**hashCode**

public int **hashCode** ()

**setCodedTypes**

public void **setCodedTypes** (*Set*<*CodedType*> *codedTypes*)

**setCredentials**

public void **setCredentials** (*Set*<*Credential*> *credentials*)

**setDemographic**

public void **setDemographic** (*Person demographic*)

**setExtensions**

public void **setExtensions** (*Set<Extension> extensions*)

**setLanguages**

public void **setLanguages** (*Set<CodedType> languages*)

**setOtherIDs**

public void **setOtherIDs** (*Set<OtherID> otherIDs*)

**setProviderFacilities**

public void **setProviderFacilities** (*Set<ProviderFacility> providerFacilities*)

**setProviderFacilitySet**

public void **setProviderFacilitySet** (*Set<ProviderFacility> providerFacilitySet*)

**setProviderOrganizationSet**

public void **setProviderOrganizationSet** (*Set<ProviderOrganization> providerOrganizationSet*)

**setProviderOrganizations**

public void **setProviderOrganizations** (*Set<ProviderOrganization> providerOrganizations*)

**setSpecialties**

public void **setSpecialties** (*Set<CodedType> specialties*)

**toString**

public *String* **toString** ()

**ProviderDirectory**

public class **ProviderDirectory**

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="provider" type="{urn:ihe:iti:csd:2013}provider" maxOccurs="unbounded" minOccurs="1"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

## Constructors

### ProviderDirectory

public **ProviderDirectory** ()

### ProviderDirectory

public **ProviderDirectory** ([Set<Provider>](#) providers)

## Methods

### equals

public boolean **equals** ([Object o](#))

### getProviders

public [Set<Provider>](#) **getProviders** ()

### hashCode

public int **hashCode** ()

### setProviders

public void **setProviders** ([Set<Provider>](#) providers)

### toString

public [String](#) **toString** ()

## ProviderFacility

public class **ProviderFacility** extends [AbstractUniqueID](#)

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="service" maxOccurs="unbounded" minOccurs="0">
          <complexType>
            <complexContent>
              <extension base="{urn:ihe:iti:csd:2013}uniqueID">
                <sequence>
                  <element name="name" type="{urn:ihe:iti:csd:2013}name" maxOccurs="unbounded" minOccurs="0"/>
                  <element name="organization" type="{urn:ihe:iti:csd:2013}uniqueID" minOccurs="0"/>
                  <element name="language" type="{urn:ihe:iti:csd:2013}codedtype" maxOccurs="unbounded" minOccurs="0"/>
                  <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="0"/>
                  <element name="freeBusyURI" type="{http://www.w3.org/2001/XMLSchema}anyURI" minOccurs="0"/>
                  <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
                </sequence>
              </extension>
            </complexContent>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
        </element>
        <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="1"/>
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
    </sequence>
</extension>
</complexContent>
</complexType>
```

## Constructors

### ProviderFacility

public **ProviderFacility** ()

### ProviderFacility

public **ProviderFacility** (*String entityID*)

### ProviderFacility

public **ProviderFacility** (*String entityID*, *Set<ProviderFacilityService> services*, *Set<OperatingHours> operatingHours*, *Set<Extension> extensions*)

## Methods

### equals

public boolean **equals** (*Object o*)

### getExtensions

public *Set<Extension>* **getExtensions** ()

### getOperatingHours

public *Set<OperatingHours>* **getOperatingHours** ()

### getServices

public *Set<ProviderFacilityService>* **getServices** ()

### hashCode

public int **hashCode** ()

### setExtensions

public void **setExtensions** (*Set<Extension> extensions*)

### setOperatingHours

public void **setOperatingHours** (*Set<OperatingHours> operatingHours*)

### setServices

public void **setServices** (*Set<ProviderFacilityService> services*)

**toString**

```
public String toString ()
```

**ProviderFacilityService**

```
public class ProviderFacilityService extends AbstractUniqueID
```

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="name" type="{urn:ihe:iti:csd:2013}name" maxOccurs="unbounded" minOccurs="0"/>
        <element name="organization" type="{urn:ihe:iti:csd:2013}uniqueID" minOccurs="0"/>
        <element name="language" type="{urn:ihe:iti:csd:2013}codedType" maxOccurs="unbounded" minOccurs="0"/>
        <element name="operatingHours" type="{urn:ihe:iti:csd:2013}operatingHours" maxOccurs="unbounded" minOccurs="0"/>
        <element name="freeBusyURI" type="{http://www.w3.org/2001/XMLSchema}anyURI" minOccurs="0"/>
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

**Constructors****ProviderFacilityService**

```
public ProviderFacilityService ()
```

**ProviderFacilityService**

```
public ProviderFacilityService (String entityID)
```

**ProviderFacilityService**

```
public ProviderFacilityService (String entityID, Set<Name> names, String providerOrganization, Set<CodedType> languages, Set<OperatingHours> operatingHours, String freeBusyURI, Set<Extension> extensions)
```

**Methods****equals**

```
public boolean equals (Object o)
```

**getExtensions**

```
public Set<Extension> getExtensions ()
```

**getFreeBusyURI**

```
public String getFreeBusyURI ()
```



**getLanguages**

```
public Set<CodedType> getLanguages ()
```

**getNames**

```
public Set<Name> getNames ()
```

**getOperatingHours**

```
public Set<OperatingHours> getOperatingHours ()
```

**getOrganization**

```
public UniqueID getOrganization ()
```

**getProviderOrganization**

```
public String getProviderOrganization ()
```

**hashCode**

```
public int hashCode ()
```

**setExtensions**

```
public void setExtensions (Set<Extension> extensions)
```

**setFreeBusyURI**

```
public void setFreeBusyURI (String freeBusyURI)
```

**setLanguages**

```
public void setLanguages (Set<CodedType> languages)
```

**setNames**

```
public void setNames (Set<Name> names)
```

**setOperatingHours**

```
public void setOperatingHours (Set<OperatingHours> operatingHours)
```

**setOrganization**

```
public void setOrganization (UniqueID organization)
```

**setProviderOrganization**

```
public void setProviderOrganization (String providerOrganization)
```

**toString**

```
public String toString ()
```

## ProviderOrganization

public class **ProviderOrganization** extends [AbstractUniqueID](#)

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
        <element name="address" type="{urn:ihe:iti:csd:2013}address" maxOccurs="unbounded" minOccurs="0"/>
        <element name="contactPoint" type="{urn:ihe:iti:csd:2013}contactPoint" maxOccurs="unbounded" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

## Constructors

### ProviderOrganization

public **ProviderOrganization** ()

### ProviderOrganization

public **ProviderOrganization** ([String entityID](#))

### ProviderOrganization

public **ProviderOrganization** ([String entityID](#), [Set<Extension>](#) extensions, [Set<Address>](#) addresses, [Set<ContactPoint>](#) contactPoints)

## Methods

### equals

public boolean **equals** ([Object o](#))

### getAddresses

public [Set<Address>](#) **getAddresses** ()

### getContactPoints

public [Set<ContactPoint>](#) **getContactPoints** ()

### getExtensions

public [Set<Extension>](#) **getExtensions** ()

### hashCode

public int **hashCode** ()

**setAddresses**

public void **setAddresses** ([Set<Address>](#) *addresses*)

**setContactPoints**

public void **setContactPoints** ([Set<ContactPoint>](#) *contactPoints*)

**setExtensions**

public void **setExtensions** ([Set<Extension>](#) *extensions*)

**toString**

public [String](#) **toString** ()

**Record**

public class **Record**

Java class for record complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="record">
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <attribute name="created" use="required" type="{http://www.w3.org/2001/XMLSchema}dateTime" />
      <attribute name="updated" use="required" type="{http://www.w3.org/2001/XMLSchema}dateTime" />
      <attribute name="status" use="required" type="{http://www.w3.org/2001/XMLSchema}string" />
      <attribute name="sourceDirectory" type="{http://www.w3.org/2001/XMLSchema}anyURI" />
    </restriction>
  </complexContent>
</complexType>
```

**Constructors****Record**

public **Record** ()

**Record**

public **Record** ([DateTime](#) *created*, [String](#) *status*, [DateTime](#) *updated*)

**Record**

public **Record** ([DateTime](#) *created*, [DateTime](#) *updated*, [String](#) *status*, [String](#) *sourceDirectory*)

**Methods****equals**

public boolean **equals** ([Object](#) *o*)

**getCreated**

public [DateTime](#) **getCreated** ()

**getSourceDirectory**

```
public String getSourceDirectory ()
```

**getStatus**

```
public String getStatus ()
```

**getUpdated**

```
public DateTime getUpdated ()
```

**hashCode**

```
public int hashCode ()
```

**setCreated**

```
public void setCreated (DateTime created)
```

**setSourceDirectory**

```
public void setSourceDirectory (String sourceDirectory)
```

**setStatus**

```
public void setStatus (String status)
```

**setUpdated**

```
public void setUpdated (DateTime updated)
```

**toString**

```
public String toString ()
```

**Service**

```
public class Service extends BaseMainEntity
```

Java class for service complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType name="service">
  <complexContent>
    <extension base="{urn:ihe:iti:csd:2013}uniqueID">
      <sequence>
        <element name="codedType" type="{urn:ihe:iti:csd:2013}codedtype"/>
        <element name="extension" type="{urn:ihe:iti:csd:2013}extension" maxOccurs="unbounded" minOccurs="0"/>
        <element name="record" type="{urn:ihe:iti:csd:2013}record"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

## Constructors

### Service

```
public Service ()
```

### Service

```
public Service (String entityID, CodedType codedType, Record record)
```

### Service

```
public Service (String entityID, CodedType codedType, Set<Extension> extensions, Record record)
```

## Methods

### equals

```
public boolean equals (Object o)
```

### getCodedType

```
public CodedType getCodedType ()
```

### getExtensions

```
public Set<Extension> getExtensions ()
```

### hashCode

```
public int hashCode ()
```

### setCodedType

```
public void setCodedType (CodedType codedType)
```

### setExtensions

```
public void setExtensions (Set<Extension> extensions)
```

### toString

```
public String toString ()
```

## ServiceDirectory

```
public class ServiceDirectory
```

Java class for anonymous complex type.

The following schema fragment specifies the expected content contained within this class.

```
<complexType>
  <complexContent>
    <restriction base="{http://www.w3.org/2001/XMLSchema}anyType">
      <sequence>
        <element name="service" type="{urn:ihe:iti:csd:2013}service" maxOccurs="unbounded" minOccurs="1"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

```
</complexContent>
</complexType>
```

## Constructors

### ServiceDirectory

```
public ServiceDirectory ()
```

### ServiceDirectory

```
public ServiceDirectory (Set<Service> services)
```

## Methods

### equals

```
public boolean equals (Object o)
```

### getServices

```
public Set<Service> getServices ()
```

### hashCode

```
public int hashCode ()
```

### setServices

```
public void setServices (Set<Service> services)
```

### toString

```
public String toString ()
```

## UniqueID

```
public class UniqueID
```

## Constructors

### UniqueID

```
public UniqueID ()
```

### UniqueID

```
public UniqueID (String entityID)
```

## Methods

### getEntityID

```
public String getEntityID ()
```

**setEntityID**

public void **setEntityID** ([String](#) *entityID*)

### 19.14.30 org.motechproject.csd.exception

**ConfigNotFoundException**

public class **ConfigNotFoundException** extends [RuntimeException](#)

The `ConfigNotFoundException` exception signals a situation when an CSD configuration with a given name cannot be found.

**Constructors****ConfigNotFoundException**

public **ConfigNotFoundException** ([String](#) *message*)

**ConfigNotFoundException**

public **ConfigNotFoundException** ([String](#) *message*, [Throwable](#) *cause*)

### 19.14.31 org.motechproject.csd.mds

**AddressDataService**

public interface **AddressDataService** extends [MotechDataService](#)<[Address](#)>

**AddressLineDataService**

public interface **AddressLineDataService** extends [MotechDataService](#)<[AddressLine](#)>

**CodedTypeDataService**

public interface **CodedTypeDataService** extends [MotechDataService](#)<[CodedType](#)>

**ContactPointDataService**

public interface **ContactPointDataService** extends [MotechDataService](#)<[ContactPoint](#)>

**CredentialDataService**

public interface **CredentialDataService** extends [MotechDataService](#)<[Credential](#)>

**ExtensionDataService**

public interface **ExtensionDataService** extends [MotechDataService](#)<[Extension](#)>

## FacilityDataService

public interface **FacilityDataService** extends MotechDataService<Facility>

### Methods

#### findByEntityID

Facility **findByEntityID** (String *entityID*)

## FacilityOrganizationDataService

public interface **FacilityOrganizationDataService** extends MotechDataService<FacilityOrganization>

## FacilityOrganizationServiceDataService

public interface **FacilityOrganizationServiceDataService** extends MotechDataService<FacilityOrganizationService>

## GeocodeDataService

public interface **GeocodeDataService** extends MotechDataService<Geocode>

## NameDataService

public interface **NameDataService** extends MotechDataService<Name>

## OperatingHoursDataService

public interface **OperatingHoursDataService** extends MotechDataService<OperatingHours>

## OrganizationContactDataService

public interface **OrganizationContactDataService** extends MotechDataService<OrganizationContact>

## OrganizationDataService

public interface **OrganizationDataService** extends MotechDataService<Organization>

### Methods

#### findByEntityID

Organization **findByEntityID** (String *entityID*)

## OtherIDDataService

public interface **OtherIDDataService** extends MotechDataService<OtherID>



### OtherNameDataService

public interface **OtherNameDataService** extends MotechDataService<OtherName>

### PersonDataService

public interface **PersonDataService** extends MotechDataService<Person>

### PersonNameDataService

public interface **PersonNameDataService** extends MotechDataService<PersonName>

### ProviderDataService

public interface **ProviderDataService** extends MotechDataService<Provider>

#### Methods

##### findByEntityID

Provider **findByEntityID** (String *entityID*)

### ProviderFacilityDataService

public interface **ProviderFacilityDataService** extends MotechDataService<ProviderFacility>

### ProviderFacilityServiceDataService

public interface **ProviderFacilityServiceDataService** extends MotechDataService<ProviderFacilityService>

### ProviderOrganizationDataService

public interface **ProviderOrganizationDataService** extends MotechDataService<ProviderOrganization>

### ServiceDataService

public interface **ServiceDataService** extends MotechDataService<Service>

#### Methods

##### findByEntityID

Service **findByEntityID** (String *entityID*)

## 19.14.32 org.motechproject.csd.service

### CSDService

public interface **CSDService**

## Methods

### delete

void **delete** ()

### fetchAndUpdate

void **fetchAndUpdate** (String *xmlUrl*)

### fetchAndUpdateUsingREST

void **fetchAndUpdateUsingREST** (String *xmlUrl*)

### fetchAndUpdateUsingSOAP

void **fetchAndUpdateUsingSOAP** (String *xmlUrl*, DateTime *lastModified*)

### getByLastModification

CSD **getByLastModification** (DateTime *lastModified*)

### getCSD

CSD **getCSD** ()

### getXmlContent

String **getXmlContent** ()

### saveFromXml

void **saveFromXml** (String *xml*)

### update

void **update** (CSD *csd*)

## ConfigService

public interface **ConfigService**

## Methods

### getConfig

Config **getConfig** (String *xmlUrl*)

### getConfigs

List<Config> **getConfigs** ()

### updateConfigs

void **updateConfigs** (List<Config> *configs*)

## FacilityService

public interface **FacilityService**

See `org.motechproject.csd.domain.Facility`

### Methods

#### **allFacilities**

`List<Facility> allFacilities ()`

#### **delete**

`void delete (String entityID)`

#### **deleteAll**

`void deleteAll ()`

#### **getFacilityByEntityID**

`Facility getFacilityByEntityID (String entityID)`

#### **getModifiedAfter**

`Set<Facility> getModifiedAfter (DateTime date)`

#### **update**

`void update (Facility facility)`

#### **update**

`void update (Set<Facility> facilities)`

## OrganizationService

public interface **OrganizationService**

### Methods

#### **allOrganizations**

`List<Organization> allOrganizations ()`

#### **delete**

`void delete (String entityID)`

#### **deleteAll**

`void deleteAll ()`

#### **getModifiedAfter**

`Set<Organization> getModifiedAfter (DateTime date)`

### **getOrganizationByEntityID**

**Organization** **getOrganizationByEntityID** (*String entityID*)

### **update**

void **update** (*Organization organization*)

### **update**

void **update** (*Set<Organization> organizations*)

## **ProviderService**

public interface **ProviderService**

### **Methods**

#### **allProviders**

*List<Provider>* **allProviders** ()

#### **delete**

void **delete** (*String entityID*)

#### **deleteAll**

void **deleteAll** ()

#### **getModifiedAfter**

*Set<Provider>* **getModifiedAfter** (*DateTime date*)

#### **getProviderByEntityID**

*Provider* **getProviderByEntityID** (*String entityID*)

#### **update**

void **update** (*Provider provider*)

#### **update**

void **update** (*Set<Provider> providers*)

## **ServiceService**

public interface **ServiceService**

### **Methods**

#### **allServices**

*List<Service>* **allServices** ()

**delete**

void **delete** (*String entityID*)

**deleteAll**

void **deleteAll** ()

**getModifiedAfter**

*Set<Service>* **getModifiedAfter** (*DateTime date*)

**getServiceByEntityID**

*Service* **getServiceByEntityID** (*String entityID*)

**update**

void **update** (*Service service*)

**update**

void **update** (*Set<Service> services*)

## 19.14.33 org.motechproject.csd.util

### CSDReferenceFinder

public class **CSDReferenceFinder**

#### Constructors

**CSDReferenceFinder**

public **CSDReferenceFinder** ()

#### Methods

**findParentEntity**

public *BaseMainEntity* **findParentEntity** (*Object o*)

**findReferences**

public *Set<Object>* **findReferences** (*AbstractID o*)

**findReferences**

public *List<Object>* **findReferences** (*AbstractID o*, *Class c*, *Field f*)

**findReferencingFields**

final *Map<Class, Field>* **findReferencingFields** (*Class clazz*)

**setBundleContext**

public void **setBundleContext** (*BundleContext bundleContext*)

## MarshallUtils

public final class **MarshallUtils**

### Methods

#### marshall

public static **String marshall** (*Object obj*, *URL schemaURL*, *Class... classesToBeBound*)

#### unmarshall

public static *Object* **unmarshall** (*String xml*, *URL schemaURL*, *Class... classesToBeBound*)

## 19.14.34 org.motechproject.csd.web

### CSDController

public class **CSDController**

### Methods

#### consume

public void **consume** (*String xmlUrl*)

#### getXml

public *String* **getXml** ()

#### handleException

public *String* **handleException** (*Exception e*)

### CSEndpoint

public class **CSEndpoint**

### Methods

#### getModifications

public *GetModificationsResponse* **getModifications** (*GetModificationsRequest request*)

### CSEndpointInterceptor

public class **CSEndpointInterceptor** implements *SoapEndpointInterceptor*

### Constructors

#### CSEndpointInterceptor

public **CSEndpointInterceptor** ()

## Methods

### afterCompletion

public void **afterCompletion** (*MessageContext messageContext*, *Object endpoint*, *Exception ex*)

### handleFault

public boolean **handleFault** (*MessageContext messageContext*, *Object endpoint*)

### handleRequest

public boolean **handleRequest** (*MessageContext messageContext*, *Object endpoint*)

### handleResponse

public boolean **handleResponse** (*MessageContext messageContext*, *Object endpoint*)

### understands

public boolean **understands** (*SoapHeaderElement header*)

## ConfigController

public class **ConfigController**

Sends & receives configs to/from the UI.

## Methods

### getConfigs

public *List<Config>* **getConfigs** ()

### handleConfigNotFoundException

public *String* **handleConfigNotFoundException** (*ConfigNotFoundException e*)

### handleException

public *String* **handleException** (*Exception e*)

### setConfigService

public void **setConfigService** (*ConfigService configService*)

### updateConfigs

public *List<Config>* **updateConfigs** (*List<Config> configs*)

## 19.14.35 org.motechproject.dhis2.domain

## DataElement

public class **DataElement**

Represents a DHIS2 Data Element

## Constructors

### DataElement

public **DataElement** ()

### DataElement

public **DataElement** (*String name*, *String uuid*)

## Methods

### getName

public *String* **getName** ()

### getUuid

public *String* **getUuid** ()

### setName

public void **setName** (*String name*)

### setUuid

public void **setUuid** (*String uuid*)

## OrgUnit

public class **OrgUnit**

Represents a DHIS2 Organisation Unit

## Constructors

### OrgUnit

public **OrgUnit** ()

### OrgUnit

public **OrgUnit** (*String name*, *String uuid*)

## Methods

### getName

public *String* **getName** ()

### getUuid

public *String* **getUuid** ()

### setName

public void **setName** (*String name*)



**setUuid**

public void **setUuid** (*String uuid*)

**Program**

public class **Program**

Represents a DHIS2 Program

**Methods****getAttributes**

public *List*<*TrackedEntityAttribute*> **getAttributes** ()

**getName**

public *String* **getName** ()

**getStages**

public *List*<*Stage*> **getStages** ()

**getTrackedEntity**

public *TrackedEntity* **getTrackedEntity** ()

**getUuid**

public *String* **getUuid** ()

**hasRegistration**

public boolean **hasRegistration** ()

**isSingleEvent**

public boolean **isSingleEvent** ()

**setAttributes**

public void **setAttributes** (*List*<*TrackedEntityAttribute*> *attributes*)

**setName**

public void **setName** (*String name*)

**setRegistration**

public void **setRegistration** (boolean *registration*)

**setSingleEvent**

public void **setSingleEvent** (boolean *singleEvent*)

**setStages**

public void **setStages** (*List*<*Stage*> *stages*)

#### **setTrackedEntity**

public void **setTrackedEntity** (*TrackedEntity trackedEntity*)

#### **setUuid**

public void **setUuid** (*String uuid*)

### **Stage**

public class **Stage**

Represents a DHIS2 program stage event

#### **Methods**

#### **getDataElements**

public *List<DataElement>* **getDataElements** ()

#### **getName**

public *String* **getName** ()

#### **getProgram**

public *String* **getProgram** ()

#### **getUuid**

public *String* **getUuid** ()

#### **hasRegistration**

public boolean **hasRegistration** ()

#### **setDataElements**

public void **setDataElements** (*List<DataElement> dataElements*)

#### **setName**

public void **setName** (*String name*)

#### **setProgram**

public void **setProgram** (*String program*)

#### **setRegistration**

public void **setRegistration** (boolean *registration*)

#### **setUuid**

public void **setUuid** (*String uuid*)

## TrackedEntity

public class **TrackedEntity**  
Represents a DHIS2 tracked entity type

### Constructors

**TrackedEntity**  
public **TrackedEntity** ()

**TrackedEntity**  
public **TrackedEntity** (*String name*, *String uuid*)

### Methods

**getName**  
public *String* **getName** ()

**getUuid**  
public *String* **getUuid** ()

**setName**  
public void **setName** (*String name*)

**setUuid**  
public void **setUuid** (*String uuid*)

## TrackedEntityAttribute

public class **TrackedEntityAttribute**  
Represents a DHIS2 tracked entity attribute

### Constructors

**TrackedEntityAttribute**  
public **TrackedEntityAttribute** ()

**TrackedEntityAttribute**  
public **TrackedEntityAttribute** (*String name*, *String uuid*)

### Methods

**getName**  
public *String* **getName** ()

#### **getUuid**

public **String** **getUuid** ()

#### **setName**

public void **setName** (**String** *name*)

#### **setUuid**

public void **setUuid** (**String** *uuid*)

### **TrackedEntityInstanceMapping**

public class **TrackedEntityInstanceMapping**

Maps an External ID to a DHIS2 ID

#### **Constructors**

##### **TrackedEntityInstanceMapping**

public **TrackedEntityInstanceMapping** ()

##### **TrackedEntityInstanceMapping**

public **TrackedEntityInstanceMapping** (**String** *externalName*, **String** *dhis2Uuid*)

#### **Methods**

##### **getDhis2Name**

public **String** **getDhis2Name** ()

##### **getDhis2Uuid**

public **String** **getDhis2Uuid** ()

##### **getExternalName**

public **String** **getExternalName** ()

##### **setDhis2Name**

public void **setDhis2Name** (**String** *dhis2Name*)

##### **setDhis2Uuid**

public void **setDhis2Uuid** (**String** *dhis2Uuid*)

##### **setExternalName**

public void **setExternalName** (**String** *externalName*)

### 19.14.36 org.motechproject.dhis2.event

#### EventHandler

public class **EventHandler**

Event Handler for the DHIS2 module. The public methods listen for the event subjects listed in `org.motechproject.dhis2.event.EventSubjects`

#### Constructors

##### EventHandler

public **EventHandler** (`DhisWebService` *webService*, `TrackedEntityInstanceMappingService` *trackedEntityInstanceMappingService*)

##### EventHandler

public **EventHandler** ()

#### Methods

##### handleCreate

public void **handleCreate** (`MotechEvent` *event*)

Parses the `MotechEvent` and creates a `org.motechproject.dhis2.rest.domain.TrackedEntityInstanceDto` which is then sent to the DHIS2 server via `org.motechproject.dhis2.rest.service.DhisWebService`

##### Parameters

- **event** – `MotechEvent` pertaining to tracked entity instance creation.

##### handleCreateAndEnroll

public void **handleCreateAndEnroll** (`MotechEvent` *event*)

Parses the event and creates a `org.motechproject.dhis2.rest.domain.TrackedEntityInstanceDto` and a `org.motechproject.dhis2.rest.domain.EnrollmentDto` which is then sent to the DHIS2 server via `org.motechproject.dhis2.rest.service.DhisWebService`

##### Parameters

- **event** – pertaining to combined DHIS2 tracked entity instance creation and enrollment.

##### handleDataValue

public void **handleDataValue** (`MotechEvent` *event*)

Parses the event and creates a `org.motechproject.dhis2.rest.domain.DataValueDto` which is then sent to the DHIS2 server via `org.motechproject.dhis2.rest.service.DhisWebService`

##### Parameters

- **event** –

##### handleDataValueSet

public void **handleDataValueSet** (`MotechEvent` *event*)

Parses the event and creates a `org.motechproject.dhis2.rest.domain.DataValueSetDto` which is then sent to the DHIS2 server via `org.motechproject.dhis2.rest.service.DhisWebService`

##### Parameters

- **event** –

#### **handleEnrollment**

public void **handleEnrollment** (MotechEvent *event*)

Parses the MotechEvent and creates a `org.motechproject.dhis2.rest.domain.EnrollmentDto` which is then sent to the DHIS2 server via `org.motechproject.dhis2.rest.service.DhisWebService`

##### **Parameters**

- **event** – MotechEvent pertaining to enrolling a tracked entity instance in a program.

#### **handleStageUpdate**

public void **handleStageUpdate** (MotechEvent *event*)

Parses the MotechEvent and creates a `org.motechproject.dhis2.rest.domain.DhisEventDto` which is then sent to the DHIS2 server via `org.motechproject.dhis2.rest.service.DhisWebService`

##### **Parameters**

- **event** – MotechEvent pertaining to a DHIS2 program stage event.

### **EventParams**

public final class **EventParams**

#### **Fields**

##### **ATTRIBUTE\_OPTION\_COMBO**

public static final `String` **ATTRIBUTE\_OPTION\_COMBO**

##### **CATEGORY\_OPTION\_COMBO**

public static final `String` **CATEGORY\_OPTION\_COMBO**

##### **COMMENT**

public static final `String` **COMMENT**

##### **COMPLETE\_DATE**

public static final `String` **COMPLETE\_DATE**

##### **DATA\_ELEMENT**

public static final `String` **DATA\_ELEMENT**

##### **DATA\_SET**

public static final `String` **DATA\_SET**

##### **DATA\_VALUES**

public static final `String` **DATA\_VALUES**

**DATE**

public static final `String` **DATE**

**ENTITY\_TYPE**

public static final `String` **ENTITY\_TYPE**

**EXTERNAL\_ID**

public static final `String` **EXTERNAL\_ID**

**LOCATION**

public static final `String` **LOCATION**

**PERIOD**

public static final `String` **PERIOD**

**PROGRAM**

public static final `String` **PROGRAM**

**REGISTRATION**

public static final `String` **REGISTRATION**

**STAGE**

public static final `String` **STAGE**

**VALUE**

public static final `String` **VALUE**

**EventSubjects**

public final class **EventSubjects**

**Fields****CREATE\_AND\_ENROLL**

public static final `String` **CREATE\_AND\_ENROLL**

**CREATE\_ENTITY**

public static final `String` **CREATE\_ENTITY**

**ENROLL\_IN\_PROGRAM**

public static final `String` **ENROLL\_IN\_PROGRAM**

**SEND\_DATA\_VALUE**

public static final `String` **SEND\_DATA\_VALUE**

## SEND\_DATA\_VALUE\_SET

public static final `String` **SEND\_DATA\_VALUE\_SET**

## UPDATE\_PROGRAM\_STAGE

public static final `String` **UPDATE\_PROGRAM\_STAGE**

## 19.14.37 org.motechproject.dhis2.repository

### DataElementDataService

public interface **DataElementDataService** extends `MotechDataService<DataElement>`  
MDS data service for `org.motechproject.dhis2.domain.DataElement`

#### Methods

##### findByName

`DataElement` **findByName** (`String` *name*)

##### findByUuid

`DataElement` **findByUuid** (`String` *uuid*)

### OrgUnitDataService

public interface **OrgUnitDataService** extends `MotechDataService<OrgUnit>`  
MDS data service for `org.motechproject.dhis2.domain.OrgUnit`

#### Methods

##### findByName

`OrgUnit` **findByName** (`String` *name*)

##### findByUuid

`OrgUnit` **findByUuid** (`String` *uuid*)

### ProgramDataService

public interface **ProgramDataService** extends `MotechDataService<Program>`  
MDS data service for `org.motechproject.dhis2.domain.Program`

#### Methods

##### findByRegistration

`List<Program>` **findByRegistration** (`Boolean` *registration*)

##### findByUuid

`Program` **findByUuid** (`String` *uuid*)



## StageDataService

public interface **StageDataService** extends MotechDataService<Stage>  
MDS data service for `org.motechproject.dhis2.domain.Stage`

### Methods

**findByUuid**  
`Stage` **findByUuid** (`String uuid`)

## TrackedEntityAttributeDataService

public interface **TrackedEntityAttributeDataService** extends MotechDataService<TrackedEntityAttribute>  
MDS data service for `org.motechproject.dhis2.domain.TrackedEntityAttribute`

### Methods

**findByUuid**  
`TrackedEntityAttribute` **findByUuid** (`String id`)

## TrackedEntityDataService

public interface **TrackedEntityDataService** extends MotechDataService<TrackedEntity>  
MDS data service for `org.motechproject.dhis2.domain.TrackedEntity`

### Methods

**findByUuid**  
`TrackedEntity` **findByUuid** (`String uuid`)

## TrackedEntityInstanceMappingDataService

public interface **TrackedEntityInstanceMappingDataService** extends MotechDataService<TrackedEntityInstanceMapping>  
MDS data service for `TrackedEntityInstanceMapping`

### Methods

**findByExternalName**  
`TrackedEntityInstanceMapping` **findByExternalName** (`String externalName`)

## 19.14.38 org.motechproject.dhis2.rest.domain

### AttributeDto

public class **AttributeDto**  
A class to model the DHIS2 API's attributes resource.

## Methods

### **equals**

public boolean **equals** (*Object obj*)

### **getAttribute**

public *String* **getAttribute** ()

### **getCode**

public *String* **getCode** ()

### **getDisplayName**

public *String* **getDisplayName** ()

### **getType**

public *String* **getType** ()

### **getValue**

public *String* **getValue** ()

### **hashCode**

public int **hashCode** ()

### **setAttribute**

public void **setAttribute** (*String attribute*)

### **setCode**

public void **setCode** (*String code*)

### **setDisplayName**

public void **setDisplayName** (*String displayName*)

### **setType**

public void **setType** (*String type*)

### **setValue**

public void **setValue** (*String value*)

## BaseDto

public abstract class **BaseDto**

## Methods

### **getCode**

public [String](#) **getCode** ()

### **getCreated**

public [String](#) **getCreated** ()

### **getHref**

public [String](#) **getHref** ()

### **getId**

public [String](#) **getId** ()

### **getLastUpdated**

public [String](#) **getLastUpdated** ()

### **getName**

public [String](#) **getName** ()

### **setCode**

public void **setCode** ([String](#) *code*)

### **setCreated**

public void **setCreated** ([String](#) *created*)

### **setHref**

public void **setHref** ([String](#) *href*)

### **setId**

public void **setId** ([String](#) *id*)

### **setLastUpdated**

public void **setLastUpdated** ([String](#) *lastUpdated*)

### **setName**

public void **setName** ([String](#) *name*)

## CoordinateDto

public class **CoordinateDto**

A class to model coordinates for program stage events.

## Methods

### **equals**

public boolean **equals** (*Object obj*)

### **getLatitude**

public *Double* **getLatitude** ()

### **getLongitude**

public *Double* **getLongitude** ()

### **hashCode**

public int **hashCode** ()

### **setLatitude**

public void **setLatitude** (*Double latitude*)

### **setLongitude**

public void **setLongitude** (*Double longitude*)

## DataElementDto

public class **DataElementDto** extends *BaseDto*

A class to model the DHIS2 API's data element resource.

## DataValueDto

public class **DataValueDto**

A class to model data values associated with program stage events in the DHIS2 API.

## Methods

### **equals**

public boolean **equals** (*Object obj*)

### **getCategoryOptionCombo**

public *String* **getCategoryOptionCombo** ()

### **getComment**

public *String* **getComment** ()

### **getDataElement**

public *String* **getDataElement** ()

**getOrgUnit**

```
public String getOrgUnit ()
```

**getPeriod**

```
public String getPeriod ()
```

**getValue**

```
public String getValue ()
```

**hashCode**

```
public int hashCode ()
```

**setCategoryOptionCombo**

```
public void setCategoryOptionCombo (String categoryOptionCombo)
```

**setComment**

```
public void setComment (String comment)
```

**setDataElement**

```
public void setDataElement (String dataElement)
```

**setOrgUnit**

```
public void setOrgUnit (String orgUnit)
```

**setPeriod**

```
public void setPeriod (String period)
```

**setValue**

```
public void setValue (String value)
```

**DataValueSetDto**

```
public class DataValueSetDto
```

A class to model a DHIS2 Data Value Set

**Methods****getAttributeOptionCombo**

```
public String getAttributeOptionCombo ()
```

**getCategoryOptionCombo**

```
public String getCategoryOptionCombo ()
```

**getComment**

```
public String getComment ()
```

**getCompleteDate**

```
public String getCompleteDate ()
```

**getDataSet**

```
public String getDataSet ()
```

**getDataValues**

```
public List<DataValueDto> getDataValues ()
```

**getOrgUnit**

```
public String getOrgUnit ()
```

**getPeriod**

```
public String getPeriod ()
```

**setAttributeOptionCombo**

```
public void setAttributeOptionCombo (String attributeOptionCombo)
```

**setCategoryOptionCombo**

```
public void setCategoryOptionCombo (String categoryOptionCombo)
```

**setComment**

```
public void setComment (String comment)
```

**setCompleteDate**

```
public void setCompleteDate (String completeDate)
```

**setDataSet**

```
public void setDataSet (String dataSet)
```

**setDataValues**

```
public void setDataValues (List<DataValueDto> dataValues)
```

**setOrgUnit**

```
public void setOrgUnit (String orgUnit)
```

**setPeriod**

```
public void setPeriod (String period)
```

**DhisDataValueStatusResponse**

```
public class DhisDataValueStatusResponse
```

A class to model DHIS2 responses that result from importing data values and data value sets.

## Methods

### **getDataValueCount**

public [ImportCountDto](#) **getDataValueCount** ()

### **getDescription**

public [String](#) **getDescription** ()

### **getImportCountDto**

public [ImportCountDto](#) **getImportCountDto** ()

### **getStatus**

public [DhisStatus](#) **getStatus** ()

### **isDataSetComplete**

public boolean **isDataSetComplete** ()

### **setDataSetComplete**

public void **setDataSetComplete** (boolean *dataSetComplete*)

### **setDataValueCount**

public void **setDataValueCount** ([ImportCountDto](#) *dataValueCount*)

### **setDescription**

public void **setDescription** ([String](#) *description*)

### **setImportCountDto**

public void **setImportCountDto** ([ImportCountDto](#) *importCountDto*)

### **setStatus**

public void **setStatus** ([DhisStatus](#) *status*)

## **DhisEventDto**

public class **DhisEventDto**

A class to model program stage events posted to the DHIS2 API.

## Methods

### **equals**

public boolean **equals** ([Object](#) *obj*)

### **getCoordinate**

public [CoordinateDto](#) **getCoordinate** ()

**getDataValues**

public [List](#)<[DataValueDto](#)> **getDataValues** ()

**getDueDate**

public [String](#) **getDueDate** ()

**getEnrollment**

public [String](#) **getEnrollment** ()

**getEnrollmentStatus**

public [String](#) **getEnrollmentStatus** ()

**getEvent**

public [String](#) **getEvent** ()

**getEventDate**

public [String](#) **getEventDate** ()

**getOrgUnit**

public [String](#) **getOrgUnit** ()

**getOrgUnitName**

public [String](#) **getOrgUnitName** ()

**getProgram**

public [String](#) **getProgram** ()

**getProgramStage**

public [String](#) **getProgramStage** ()

**getStatus**

public [String](#) **getStatus** ()

**getStoredBy**

public [String](#) **getStoredBy** ()

**getTrackedEntityInstance**

public [String](#) **getTrackedEntityInstance** ()

**hashCode**

public int **hashCode** ()

**isFollowup**

public boolean **isFollowup** ()



**setCoordinate**

public void **setCoordinate** ([CoordinateDto](#) *coordinate*)

**setDataValues**

public void **setDataValues** ([List](#)<[DataValueDto](#)> *dataValues*)

**setDueDate**

public void **setDueDate** ([String](#) *dueDate*)

**setEnrollment**

public void **setEnrollment** ([String](#) *enrollment*)

**setEnrollmentStatus**

public void **setEnrollmentStatus** ([String](#) *enrollmentStatus*)

**setEvent**

public void **setEvent** ([String](#) *event*)

**setEventDate**

public void **setEventDate** ([String](#) *eventDate*)

**setFollowup**

public void **setFollowup** (boolean *followup*)

**setOrgUnit**

public void **setOrgUnit** ([String](#) *orgUnit*)

**setOrgUnitName**

public void **setOrgUnitName** ([String](#) *orgUnitName*)

**setProgram**

public void **setProgram** ([String](#) *program*)

**setProgramStage**

public void **setProgramStage** ([String](#) *programStage*)

**setStatus**

public void **setStatus** ([String](#) *status*)

**setStoredBy**

public void **setStoredBy** ([String](#) *storedBy*)

**setTrackedEntityInstance**

public void **setTrackedEntityInstance** ([String](#) *trackedEntityInstance*)

## DhisStatus

public enum **DhisStatus**

### Enum Constants

#### ERROR

public static final **DhisStatus** **ERROR**

#### SUCCESS

public static final **DhisStatus** **SUCCESS**

## DhisStatusResponse

public class **DhisStatusResponse**

A class to model DHIS2 responses that result from creating entities.

### Methods

#### equals

public boolean **equals** (*Object obj*)

#### getImportCount

public **ImportCountDto** **getImportCount** ()

#### getReference

public **String** **getReference** ()

#### getStatus

public **DhisStatus** **getStatus** ()

#### hashCode

public int **hashCode** ()

#### setImportCount

public void **setImportCount** (**ImportCountDto** *importCount*)

#### setReference

public void **setReference** (**String** *reference*)

#### setStatus

public void **setStatus** (**DhisStatus** *status*)

## EnrollmentDto

public class **EnrollmentDto**

A class to model enrollments posted to the DHIS2 API.

### Methods

#### **equals**

public boolean **equals** (*Object obj*)

#### **getAttributes**

public *List*<*AttributeDto*> **getAttributes** ()

#### **getDateOfEnrollment**

public *String* **getDateOfEnrollment** ()

#### **getDateOfIncident**

public *String* **getDateOfIncident** ()

#### **getProgram**

public *String* **getProgram** ()

#### **getTrackedEntityInstance**

public *String* **getTrackedEntityInstance** ()

#### **hashCode**

public int **hashCode** ()

#### **isFollowup**

public boolean **isFollowup** ()

#### **setAttributes**

public void **setAttributes** (*List*<*AttributeDto*> *attributes*)

#### **setDateOfEnrollment**

public void **setDateOfEnrollment** (*String dateOfEnrollment*)

#### **setDateOfIncident**

public void **setDateOfIncident** (*String dateOfIncident*)

#### **setFollowup**

public void **setFollowup** (boolean *followup*)

#### **setProgram**

public void **setProgram** (*String program*)

### **setTrackedEntityInstance**

public void **setTrackedEntityInstance** (*String trackedEntityInstance*)

## **ImportCountDto**

public class **ImportCountDto**

A class to model a DHIS2 api response import count

### **Methods**

#### **equals**

public boolean **equals** (*Object obj*)

#### **getDeleted**

public int **getDeleted** ()

#### **getIgnored**

public int **getIgnored** ()

#### **getImported**

public int **getImported** ()

#### **getUpdated**

public int **getUpdated** ()

#### **hashCode**

public int **hashCode** ()

#### **setDeleted**

public void **setDeleted** (int *deleted*)

#### **setIgnored**

public void **setIgnored** (int *ignored*)

#### **setImported**

public void **setImported** (int *imported*)

#### **setUpdated**

public void **setUpdated** (int *updated*)

## **OrganisationUnitDto**

public class **OrganisationUnitDto** extends *BaseDto*

A class to model the DHIS2 API's organisation unit resource.

## PagedResourceDto

```
public class PagedResourceDto<T extends BaseDto>
```

### Methods

#### getPager

```
public PagerDto getPager ()
```

#### getResources

```
public List<T> getResources ()
```

#### setPager

```
public void setPager (PagerDto pager)
```

#### setResources

```
public void setResources (String key, List<T> resources)
```

## PagerDto

```
public class PagerDto
```

### Methods

#### getNextPage

```
public String getNextPage ()
```

#### getPage

```
public int getPage ()
```

#### getPageCount

```
public int getPageCount ()
```

#### getPrevPage

```
public String getPrevPage ()
```

#### getTotal

```
public int getTotal ()
```

#### setNextPage

```
public void setNextPage (String nextPage)
```

#### setPage

```
public void setPage (int page)
```

### **setPageCount**

public void **setPageCount** (int *pageCount*)

### **setPrevPage**

public void **setPrevPage** (String *prevPage*)

### **setTotal**

public void **setTotal** (int *total*)

## **ProgramDto**

public class **ProgramDto** extends [BaseDto](#)

A class to model the DHIS2 API's program resource.

## **Methods**

### **equals**

public boolean **equals** (Object *obj*)

### **getOrganisationUnits**

public List<[OrganisationUnitDto](#)> **getOrganisationUnits** ()

### **getProgramStages**

public List<[ProgramStageDto](#)> **getProgramStages** ()

### **getProgramTrackedEntityAttributes**

public List<[ProgramTrackedEntityAttributeDto](#)> **getProgramTrackedEntityAttributes** ()

### **getRegistration**

public boolean **getRegistration** ()

### **getSingleEvent**

public boolean **getSingleEvent** ()

### **getTrackedEntity**

public [TrackedEntityDto](#) **getTrackedEntity** ()

### **hashCode**

public int **hashCode** ()

### **setOrganisationUnits**

public void **setOrganisationUnits** (List<[OrganisationUnitDto](#)> *organisationUnits*)

### **setProgramStages**

public void **setProgramStages** (List<[ProgramStageDto](#)> *programStages*)

**setProgramTrackedEntityAttributes**

```
public void setProgramTrackedEntityAttributes (List<ProgramTrackedEntityAttributeDto> programTrackedEntityAttributes)
```

**setRegistration**

```
public void setRegistration (boolean registration)
```

**setSingleEvent**

```
public void setSingleEvent (boolean singleEvent)
```

**setTrackedEntity**

```
public void setTrackedEntity (TrackedEntityDto trackedEntity)
```

**ProgramStageDataElementDto**

```
public class ProgramStageDataElementDto
```

A class to model program stage data elements returned by the DHIS2 API.

**Methods****equals**

```
public boolean equals (Object obj)
```

**getDataElement**

```
public DataElementDto getDataElement ()
```

**getProgramStage**

```
public ProgramStageDto getProgramStage ()
```

**hashCode**

```
public int hashCode ()
```

**setDataElement**

```
public void setDataElement (DataElementDto dataElement)
```

**setProgramStage**

```
public void setProgramStage (ProgramStageDto programStage)
```

**ProgramStageDto**

```
public class ProgramStageDto extends BaseDto
```

A class to model program stages returned by the DHIS2 API.

## Methods

### **equals**

public boolean **equals** (*Object obj*)

### **getProgram**

public [ProgramDto](#) **getProgram** ()

### **getProgramStageDataElements**

public [List](#)<[ProgramStageDataElementDto](#)> **getProgramStageDataElements** ()

### **hashCode**

public int **hashCode** ()

### **setProgram**

public void **setProgram** ([ProgramDto](#) *program*)

### **setProgramStageDataElements**

public void **setProgramStageDataElements** ([List](#)<[ProgramStageDataElementDto](#)> *programStageDataElements*)

## **ProgramTrackedEntityAttributeDto**

public class **ProgramTrackedEntityAttributeDto**

A class to model program tracked entity attributes returned by the DHIS2 API.

## Methods

### **getTrackedEntityAttribute**

public [TrackedEntityAttributeDto](#) **getTrackedEntityAttribute** ()

### **setTrackedEntityAttribute**

public void **setTrackedEntityAttribute** ([TrackedEntityAttributeDto](#) *trackedEntityAttribute*)

## **TrackedEntityAttributeDto**

public class **TrackedEntityAttributeDto** extends [BaseDto](#)

A class to model the DHIS2 API's tracked entity attribute resource.

## **TrackedEntityDto**

public class **TrackedEntityDto** extends [BaseDto](#)

A class to model the DHIS2 API's tracked entity resource.



## TrackedEntityInstanceDto

public class **TrackedEntityInstanceDto**

A class to model tracked entity instances posted to the DHIS2 API.

### Methods

#### **equals**

public boolean **equals** (*Object obj*)

#### **getAttributes**

public *List*<*AttributeDto*> **getAttributes** ()

#### **getCreated**

public *String* **getCreated** ()

#### **getOrgUnit**

public *String* **getOrgUnit** ()

#### **getTrackedEntity**

public *String* **getTrackedEntity** ()

#### **getTrackedEntityInstance**

public *String* **getTrackedEntityInstance** ()

#### **hashCode**

public int **hashCode** ()

#### **setAttributes**

public void **setAttributes** (*List*<*AttributeDto*> *attributes*)

#### **setCreated**

public void **setCreated** (*String created*)

#### **setOrgUnit**

public void **setOrgUnit** (*String orgUnit*)

#### **setTrackedEntity**

public void **setTrackedEntity** (*String trackedEntity*)

#### **setTrackedEntityInstance**

public void **setTrackedEntityInstance** (*String trackedEntityInstance*)

## 19.14.39 org.motechproject.dhis2.rest.service

### DhisWebException

public class **DhisWebException** extends [RuntimeException](#)

Thrown if there is an error with an HTTP request or if there is an error marshalling to or from a data transfer object

### Constructors

#### DhisWebException

public **DhisWebException** ([String message](#))

#### DhisWebException

public **DhisWebException** ([String message](#), [Throwable cause](#))

### DhisWebService

public interface **DhisWebService**

Handles HTTP requests and responses to and from DHIS2.

### Methods

#### createEnrollment

[DhisStatusResponse](#) **createEnrollment** ([EnrollmentDto enrollmentDto](#))

Attempts to create an enrollment in DHIS2 via an HTTP post request

#### Parameters

- **enrollmentDto** –

**Returns** a [org.motechproject.dhis2.rest.domain.DhisStatusResponse](#) indicating success or failure

#### createEvent

[DhisStatusResponse](#) **createEvent** ([DhisEventDto event](#))

Attempts to create a DHIS2 event in DHIS2 via an HTTP post request

#### Parameters

- **event** –

**Returns** a [org.motechproject.dhis2.rest.domain.DhisStatusResponse](#) indicating success or failure

#### createTrackedEntityInstance

[DhisStatusResponse](#) **createTrackedEntityInstance** ([TrackedEntityInstanceDto trackedEntity](#))

Attempts to create a tracked entity instance in DHIS2 via an HTTP post request

#### Parameters

- **trackedEntity** –

**getDataElementByHref**

`DataElementDto` **getDataElementByHref** (*String href*)

Gets the Data Element specified in the URL

**Parameters**

- **href** –

**Returns** the `org.motechproject.dhis2.rest.domain.DataElementDto`

**getDataElements**

`List<DataElementDto>` **getDataElements** ()

Gets a list of all the Data Elements from DHIS2.

**Returns** a list of `org.motechproject.dhis2.rest.domain.DataElementDto`

**getOrganisationUnitByHref**

`OrganisationUnitDto` **getOrganisationUnitByHref** (*String href*)

Gets the Organisation Unit specified in the URL

**Parameters**

- **href** –

**Returns** the `org.motechproject.dhis2.rest.domain.OrganisationUnitDto`

**getOrganisationUnits**

`List<OrganisationUnitDto>` **getOrganisationUnits** ()

Gets a list of all the Organisation Units from DHIS2

**Returns** a list of `org.motechproject.dhis2.rest.domain.OrganisationUnitDto`

**getProgramByHref**

`ProgramDto` **getProgramByHref** (*String href*)

Gets the program specified in the URL

**Parameters**

- **href** –

**Returns** the `org.motechproject.dhis2.rest.domain.ProgramDto`

**getProgramStageByHref**

`ProgramStageDto` **getProgramStageByHref** (*String href*)

Gets the program stage specified in the URL

**Parameters**

- **href** –

**Returns** the `org.motechproject.dhis2.rest.domain.ProgramStageDto`

**getProgramStages**

`List<ProgramStageDto>` **getProgramStages** ()

Gets a list of all the program Stages from DHIS2

**Returns** a list of `:java:ref:'org.motechproject.dhis2.rest.domain.ProgramStageDto'`

**getPrograms**`List<ProgramDto> getPrograms ()`

Gets a list of all the programs from DHIS2.

**Returns** a list of `org.motechproject.dhis2.rest.domain.ProgramDto`

**getTrackedEntities**`List<TrackedEntityDto> getTrackedEntities ()`

Gets a list of all the tracked entity types in DHIS2

**Returns** a list of `org.motechproject.dhis2.rest.domain.TrackedEntityDto`

**getTrackedEntityAttributeByHref**`TrackedEntityAttributeDto getTrackedEntityAttributeByHref (String href)`

Gets the tracked entity attribute specified in the URL

**Parameters**

- href –

**Returns** the `org.motechproject.dhis2.rest.domain.TrackedEntityAttributeDto`

**getTrackedEntityAttributes**`List<TrackedEntityAttributeDto> getTrackedEntityAttributes ()`

Gets a list of all the tracked entity attributes from DHIS2

**Returns** a list of `org.motechproject.dhis2.rest.domain.TrackedEntityInstanceDto`

**getTrackedEntityByHref**`TrackedEntityDto getTrackedEntityByHref (String href)`

Gets the tracked entity specified in the URL.

**Parameters**

- href –

**Returns** the `org.motechproject.dhis2.rest.domain.TrackedEntityDto`

**sendDataValueSet**`DhisDataValueStatusResponse sendDataValueSet (DataValueSetDto dataValueSetDto)`

Attempts to send a data value set to DHIS2 via an HTTP post request.

**Parameters**

- dataValueSetDto –

## 19.14.40 org.motechproject.dhis2.service

**DataElementService**

public interface **DataElementService** extends `GenericCrudService<DataElement>`

Manages CRUD operations for a `org.motechproject.dhis2.domain.DataElement`

## Methods

### createFromDetails

`DataElement` **createFromDetails** (`DataElementDto` *details*)

### findById

`DataElement` **findById** (`String` *id*)

### findByName

`DataElement` **findByName** (`String` *name*)

## GenericCrudService

public interface **GenericCrudService**<`T`>

## Methods

### delete

void **delete** (`T` *dataElement*)

### deleteAll

void **deleteAll** ()

### findAll

`List`<`T`> **findAll** ()

### update

void **update** (`T` *dataElement*)

## OrgUnitService

public interface **OrgUnitService** extends `GenericCrudService`<`OrgUnit`>

Manages CRUD operations for a `org.motechproject.dhis2.domain.OrgUnit`

## Methods

### createFromDetails

`OrgUnit` **createFromDetails** (`OrganisationUnitDto` *details*)

### findById

`OrgUnit` **findById** (`String` *id*)

### findByName

`OrgUnit` **findByName** (`String` *name*)

## ProgramService

public interface **ProgramService** extends [GenericCrudService<Program>](#)  
Manages CRUD operations for a `org.motechproject.dhis2.domain.Program`

### Methods

**createFromDetails**  
[Program](#) **createFromDetails** ([ProgramDto](#) details)

**findById**  
[Program](#) **findById** ([String](#) id)

**findByRegistration**  
[List<Program>](#) **findByRegistration** (boolean registration)

## Settings

public class **Settings**  
DHIS2 server configuration details.

### Constructors

**Settings**  
public **Settings** ()

**Settings**  
public **Settings** ([String](#) serverURI, [String](#) username, [String](#) password)

### Methods

**getPassword**  
public [String](#) **getPassword** ()

**getServerURI**  
public [String](#) **getServerURI** ()

**getUsername**  
public [String](#) **getUsername** ()

**setPassword**  
public void **setPassword** ([String](#) password)

**setServerURI**  
public void **setServerURI** ([String](#) serverURI)

**setUsername**

public void **setUsername** ([String](#) *username*)

**SettingsService**

public interface **SettingsService**

Retrieves and updates settings for the module

**See also:** [Settings](#)

**Methods****getSettings**

[Settings](#) **getSettings** ()

**updateSettings**

void **updateSettings** ([Settings](#) *settings*)

**StageService**

public interface **StageService** extends [GenericCrudService](#)<[Stage](#)>

Manages CRUD operations for a [org.motechproject.dhis2.domain.Stage](#)

**Methods****createFromDetails**

[Stage](#) **createFromDetails** ([ProgramStageDto](#) *details*, [String](#) *programId*, boolean *hasRegistration*)

**findById**

[Stage](#) **findById** ([String](#) *id*)

**SyncService**

public interface **SyncService**

Used to synchronize this module with the DHIS2 server schema

**Methods****sync**

boolean **sync** ()

Queries the DHIS2 server to get the current schema and then persists that information in MDS. If the process fails, all of the records pertaining to the DHIS2 schema are deleted.

**Returns** true if successful; false otherwise

## TasksService

public interface **TasksService**

Processes Task channel updates for this module

**See also:** `org.motechproject.tasks.service.ChannelService`,  
`org.motechproject.dhis2.tasks.ChannelRequestBuilder`

## Methods

**updateChannel**

void **updateChannel** ()

Builds a channel request from the information saved in MDS and then updates the channel for this module.

## TrackedEntityAttributeService

public interface **TrackedEntityAttributeService** extends `GenericCrudService<TrackedEntityAttribute>`

Manages CRUD operations for a `org.motechproject.dhis2.domain.TrackedEntityAttribute`

## Methods

**createFromDetails**

`TrackedEntityAttribute` **createFromDetails** (`TrackedEntityAttributeDto details`)

**findById**

`TrackedEntityAttribute` **findById** (`String id`)

## TrackedEntityInstanceMappingException

public class **TrackedEntityInstanceMappingException** extends `RuntimeException`

Thrown if there is no corresponding mapped value when attempting to map from a DHIS2 id to an external id (or vice versa).

## Constructors

**TrackedEntityInstanceMappingException**

public **TrackedEntityInstanceMappingException** (`String message`)

**TrackedEntityInstanceMappingException**

public **TrackedEntityInstanceMappingException** (`String message`, `Throwable cause`)

## TrackedEntityInstanceMappingService

public interface **TrackedEntityInstanceMappingService** extends `GenericCrudService<TrackedEntityInstanceMapping>`

Manages CRUD operations for a `TrackedEntityInstanceMapping`



## Methods

### create

`TrackedEntityInstanceMapping` **create** (`String externalId`, `String dhisId`)

### findByExternalId

`TrackedEntityInstanceMapping` **findByExternalId** (`String externalId`)

### mapFromExternalId

`String` **mapFromExternalId** (`String externalId`)

## TrackedEntityService

public interface **TrackedEntityService** extends `GenericCrudService<TrackedEntity>`

Manages CRUD operations for a `org.motechproject.dhis2.domain.TrackedEntity`

## Methods

### createFromDetails

`TrackedEntity` **createFromDetails** (`TrackedEntityDto details`)

### findById

`TrackedEntity` **findById** (`String id`)

## 19.14.41 org.motechproject.dhis2.tasks

### ChannelRequestBuilder

public class **ChannelRequestBuilder**

Builds a channel request from the records in MDS pertaining to the DHIS2 instance schema.

## Constructors

### ChannelRequestBuilder

public **ChannelRequestBuilder** (`BundleContext bundleContext`, `ProgramService programService`, `StageService stageService`, `TrackedEntityAttributeService trackedEntityAttributeService`, `TrackedEntityService trackedEntityService`)

## Methods

### build

public `ChannelRequest` **build** ()

Creates task action event requests for tracked entity instance creation, program enrollment, and program stage events.

**Returns** the new Channel Request

## CreateInstanceActionBuilder

public class **CreateInstanceActionBuilder**

Builds task action requests for tracked entity instance creation for each entity type in the DHIS2 schema.

### Methods

#### build

public [List](#)<[ActionEventRequest](#)> **build** ([List](#)<[TrackedEntityAttribute](#)> *attributes*, [List](#)<[TrackedEntity](#)> *trackedEntities*)

Takes a list of attributes and tracked entity types and builds a list of action event requests for tracked entity instance creation.

#### Parameters

- **attributes** –
- **trackedEntities** –

**Returns** a list of action event requests for creating instances of each tracked entity type

## DisplayNames

public final class **DisplayNames**

### Fields

#### ATTRIBUTE\_OPTION\_COMBO

public static final [String](#) **ATTRIBUTE\_OPTION\_COMBO**

#### CATEGORY\_OPTION\_COMBO

public static final [String](#) **CATEGORY\_OPTION\_COMBO**

#### COMMENT

public static final [String](#) **COMMENT**

#### COMPLETE\_DATE

public static final [String](#) **COMPLETE\_DATE**

#### CREATE\_TRACKED\_ENTITY\_INSTANCE

public static final [String](#) **CREATE\_TRACKED\_ENTITY\_INSTANCE**

#### DATA\_ELEMENT

public static final [String](#) **DATA\_ELEMENT**

#### DATA\_SET

public static final [String](#) **DATA\_SET**

**DATA\_VALUES**

public static final [String](#) **DATA\_VALUES**

**DHIS2\_DISPLAY\_NAME**

public static final [String](#) **DHIS2\_DISPLAY\_NAME**

**ENROLLMENT\_DATE**

public static final [String](#) **ENROLLMENT\_DATE**

**EVENT\_DATE**

public static final [String](#) **EVENT\_DATE**

**EXTERNAL\_ID**

public static final [String](#) **EXTERNAL\_ID**

**ORG\_UNIT**

public static final [String](#) **ORG\_UNIT**

**PERIOD**

public static final [String](#) **PERIOD**

**PROGRAM\_ENROLLMENT**

public static final [String](#) **PROGRAM\_ENROLLMENT**

**PROGRAM\_NAME**

public static final [String](#) **PROGRAM\_NAME**

**SEND\_DATA\_VALUE**

public static final [String](#) **SEND\_DATA\_VALUE**

**SEND\_DATA\_VALUE\_SET**

public static final [String](#) **SEND\_DATA\_VALUE\_SET**

**STAGE\_EVENT**

public static final [String](#) **STAGE\_EVENT**

**TRACKED\_ENTITY\_TYPE**

public static final [String](#) **TRACKED\_ENTITY\_TYPE**

**VALUE**

public static final [String](#) **VALUE**

## ProgramActionBuilder

public class **ProgramActionBuilder**

Builds task action requests for enrolling tracked entity instances in a program for each entity type in the DHIS2 schema. Also builds create + enroll combination task action requests for each program and the corresponding tracked entity type.

### Methods

#### build

public [List](#)<[ActionEventRequest](#)> **build** ([List](#)<[Program](#)> *programs*)

Builds a list of action event requests for program enrollment and create + enroll task actions.

#### Parameters

- **programs** –

**Returns** A list of action event rests pertaining to program enrollment.

## StageActionBuilder

public class **StageActionBuilder**

Builds task action requests for program stage events for each program stage.

### Methods

#### build

public [List](#)<[ActionEventRequest](#)> **build** ([List](#)<[Stage](#)> *stages*)

Builds a list of action event request from a list of program stages.

#### Parameters

- **stages** –

**Returns** a list of [ActionEventRequests](#)

## 19.14.42 org.motechproject.dhis2.web

### DhisSchemaController

public class **DhisSchemaController**

The controller that handles requests for information about the DHIS2 schema.

### Methods

#### getAttributes

public [List](#)<[TrackedEntityAttribute](#)> **getAttributes** ()

#### getDataElements

public [List](#)<[DataElement](#)> **getDataElements** ()

**getOrgUnits**

```
public List<OrgUnit> getOrgUnits ()
```

**getPrograms**

```
public List<Program> getPrograms ()
```

**getTrackedEntities**

```
public List<TrackedEntity> getTrackedEntities ()
```

**DhisSettingsController**

```
public class DhisSettingsController
```

The controller that handles both changes to module settings and syncing to the DHIS2 server.

**See also:** `org.motechproject.dhis2.service.SyncService`,  
`org.motechproject.dhis2.service.SettingsService`

**Methods****getSettings**

```
public Settings getSettings ()
```

Returns the settings for the module

**Returns** an `Settings` object for the DHIS2 module

**sync**

```
public boolean sync ()
```

Attempts to sync the module to the DHIS2 server. If successful, updates the task channel to reflect any changes to the DHIS2 schema

**Returns** a boolean value indicating success or failure

**updateSettings**

```
public Settings updateSettings (Settings settings)
```

Updates the settings to the configuration specified in the parameters

**Parameters**

- **settings** – the new settings configuration

**Returns** the new settings configuration

**19.14.43 org.motechproject.eventlogging.converter****EventToLogConverter**

```
public interface EventToLogConverter<T>
```

Interface to implement for custom converters. The converter should return an appropriate return type that a logger can use to log the information.

**Parameters**

- **<T>** – The return type that will be used to log.

## Methods

### convertToLog

T **convertToLog** (MotechEvent *event*)

Converts an event to a generic type that should be used to log the event.

#### Parameters

- **event** – The incoming motech event to be converted to a log.

**Returns** An appropriate object type that can be used by a logger to log the event.

## 19.14.44 org.motechproject.eventlogging.converter.impl

### DefaultDbToLogConverter

public class **DefaultDbToLogConverter** implements [EventToLogConverter<EventLog>](#)

Implementation of the [EventToLogConverter](#) interface. This class is responsible for converting MotechEvents to the class persistable in the database.

## Methods

### configuredConvertEventToDbLog

public [EventLog](#) **configuredConvertEventToDbLog** (MotechEvent *eventToLog*, [LoggableEvent](#) *loggableEvent*)

Converts an event to a [EventLog](#) object, which represents a single logged event in database. Uses [DbLoggableEvent](#) implementation of [LoggableEvent](#) to configure the returned object.

#### Parameters

- **eventToLog** – the incoming motech event to be converted to a database event log
- **loggableEvent** – contains data about an event to log, used to configure returned log instance

#### Throws

- **org.motechproject.commons.api.MotechException** – if the passed [LoggableEvent](#) is not an instance of [DbLoggableEvent](#)

**Returns** [EventLog](#) object that is used to log the event to the database

### convertToLog

public [EventLog](#) **convertToLog** (MotechEvent *eventToLog*)

### DefaultFileToLogConverter

public class **DefaultFileToLogConverter** implements [EventToLogConverter<String>](#)

Implementation of the [EventToLogConverter](#) interface. This class is responsible for converting MotechEvents to the state that can be injected in the file.

## Methods

### convertToLog

public [String](#) **convertToLog** (MotechEvent *event*)

## 19.14.45 org.motechproject.eventlogging.domain

### EventLog

public class **EventLog**

The `EventLog` class represents a single, logged event record, that is persisted in the database.

### Constructors

#### EventLog

public **EventLog** ()

Creates a new instance of `EventLog`, with all fields set to null.

#### EventLog

public **EventLog** (*String subject*, *Map<String, Object> parameters*, *DateTime timeStamp*)

Creates a new instance of `EventLog`, with all fields set to the values specified in the parameters.

#### Parameters

- **subject** – the event subject
- **parameters** – map of key-value pairs representing the event parameters
- **timeStamp** – the date/time at which the event occurred

### Methods

#### getParameters

public *Map<String, Object>* **getParameters** ()

Gets the event parameters.

**Returns** map of key-value pairs representing the event parameters

#### getSubject

public *String* **getSubject** ()

Gets the event subject.

**Returns** the event subject

#### getTimeStamp

public *DateTime* **getTimeStamp** ()

Gets the event timestamp.

**Returns** the date/time at which the event occurred

#### setParameters

public void **setParameters** (*Map<String, Object> parameters*)

Sets the event parameters.

#### Parameters

- **parameters** – map of key-value pairs representing the event parameters

**setSubject**

public void **setSubject** (*String subject*)

Sets the event subject.

**Parameters**

- **subject** – the event subject

**setTimeStamp**

public void **setTimeStamp** (*DateTime timeStamp*)

Sets the event timestamp.

**Parameters**

- **timeStamp** – the date/time at which the event occurred

## 19.14.46 org.motechproject.eventlogging.loggers

### EventLogger

public abstract class **EventLogger**

Class that represents a generic event logger. All loggers have a list of loggable events that they should be able to provide logging functionality for.

#### Methods

**addLoggableEvents**

public void **addLoggableEvents** (*List<LoggableEvent> loggableEvents*)

Adds data about events to log to loggableEvents list in the form of *LoggableEvent* objects.

**Parameters**

- **loggableEvents** – list of loggable events, which should be added to list of events to log

**clearLoggableEvents**

public void **clearLoggableEvents** ()

Removes all data about events to log by clearing the loggableEvents list.

**getLoggableEvents**

public *List<LoggableEvent>* **getLoggableEvents** ()

Returns data about all events to log

**Returns** list of *LoggableEvent* objects which contains data about events that should be added to the log

**log**

public abstract void **log** (*MotechEvent eventToLog*)

Method used to store data about events.

**Parameters**

- **eventToLog** – the incoming motech event to log



**removeLoggableEvents**

```
public void removeLoggableEvents (List<LoggableEvent> loggableEvents)
```

Removes data about events to log from the loggableEvents list in the form of `LoggableEvent` objects.

**Parameters**

- **loggableEvents** – list of loggable events, which should be removed from the list of events to log

## 19.14.47 org.motechproject.eventlogging.loggers.impl

**DbEventLogger**

```
public class DbEventLogger extends EventLogger
```

Implementation of the `EventLogger` class. The `DbEventLogger` class is responsible for logging event records in the database.

**Constructors****DbEventLogger**

```
public DbEventLogger (EventLogService eventLogService, DefaultDbToLogConverter eventConverter)
```

Creates an instance of `DbEventLogger` which provides logic for persisting logs in the database.

**Parameters**

- **eventLogService** – Motech Data Service for EventLogs, generated and injected by MDS module
- **eventConverter** – `DefaultDbToLogConverter` responsible for converting incoming events to database persisting state

**Methods****log**

```
public void log (MotechEvent eventToLog)
```

**FileEventLogger**

```
public class FileEventLogger extends EventLogger
```

Implementation of the `EventLogger` class. The `FileEventLogger` class is responsible for logging event records to the file.

**Constructors****FileEventLogger**

```
public FileEventLogger (DefaultFileToLogConverter eventConverter)
```

Creates an instance of `FileEventLogger` which provides logic for injecting logs to file.

**Parameters**

- **eventConverter** – `DefaultFileToLogConverter` responsible for converting incoming events to file injecting state

**FileEventLogger**

public **FileEventLogger** (*List<LoggableEvent> loggableEvents, List<File> loggingFiles, DefaultFileToLogConverter eventConverter*)

Creates an instance of FileEventLogger which provides logic for injecting logs to file.

**Parameters**

- **loggableEvents** – list of events to log
- **loggingFiles** – list of files to which events should be logged
- **eventConverter** – `DefaultFileToLogConverter` responsible for converting incoming events to the format in which they should be written to the file

**Methods****log**

public void **log** (*MotechEvent eventToLog*)

**log**

protected void **log** (*String informationToLog*)

Logs information about the event to every file included in `loggingFiles`.

**Parameters**

- **informationToLog** – information to be logged

## 19.14.48 org.motechproject.eventlogging.matchers

**DbLoggableEvent**

public class **DbLoggableEvent** extends `LoggableEvent`

Version of `LoggableEvent` class used when working with `DbEventLogger`. Additionally to `LoggableEvent` it stores `LogMappings` used for manipulating event parameters before persisting log in the database.

**Constructors****DbLoggableEvent**

public **DbLoggableEvent** (*List<String> eventSubjects, List<? extends EventFlag> flags, LogMappings mappings*)

Creates an instance of `DbLoggableEvent` using passed parameters

**Parameters**

- **eventSubjects** – list of event subjects to log
- **flags** – event flags used for filtering events by parameters
- **mappings** – `LogMappings` object used for replacing, including and excluding parameters before logging to the database

**Methods****getMappings**

public `LogMappings` **getMappings** ()

**setMappings**

public void **setMappings** ([LogMappings](#) *mappings*)

**EventFlag**

public interface **EventFlag**

Interface representing event flags corresponding to event parameters, used to filter events that should be logged.

**Methods****passesFlags**

boolean **passesFlags** ([MotechEvent](#) *event*)

Checks if motech event parameters contains all expected flags.

**Parameters**

- **event** – motech event which we check for passing flags

**Returns** true if motech event contains all expected flags, false otherwise

**KeyValue**

public class **KeyValue**

Class representing mapping from one parameter key-value pair to another. Used in [org.motechproject.eventlogging.converter.impl.DefaultDbToLogConverter](#) to replace those pairs before persisting logs in the database.

**Constructors****KeyValue**

public **KeyValue** ([String](#) *startKey*, [Object](#) *startValue*, [String](#) *endKey*, [Object](#) *endValue*, boolean *isOptional*)

Creates an instance of KeyValue from passed parameters. If an event contains key specified in startKey and value of this key equals startValue, then that key will be replaced by endKey and that value by endValue.

**Parameters**

- **startKey** – key to replace
- **startValue** – value of replacing key
- **endKey** – new key
- **endValue** – new value for new key
- **isOptional** – have no use at the moment

**Methods****getEndKey**

public [String](#) **getEndKey** ()

**getEndValue**

public [Object](#) **getEndValue** ()

**getStartKey**

public `String` **getStartKey** ()

**getStartValue**

public `Object` **getStartValue** ()

**isOptional**

public boolean **isOptional** ()

**setEndKey**

public void **setEndKey** (`String` *endKey*)

**setEndValue**

public void **setEndValue** (`Object` *endValue*)

**setOptional**

public void **setOptional** (boolean *isOptional*)

**setStartKey**

public void **setStartKey** (`String` *startKey*)

**setStartValue**

public void **setStartValue** (`Object` *startValue*)

## LogMappings

public class **LogMappings**

Class used to store data about manipulations on event parameters before persisting logs in the database. LogMapping allows to include, exclude and replace parameters before persisting them in the log.

### Constructors

**LogMappings**

public **LogMappings** (`List`<`KeyValue`> *mappings*, `List`<`String`> *exclusions*, `List`<`String`> *inclusions*)

Create an instance of LogMappings using passed parameters

**Parameters**

- **mappings** – list of `KeyValue` mappings
- **exclusions** – list of parameters to exclude from the log
- **inclusions** – list of parameters to include in the log

### Methods

**getExclusions**

public `List`<`String`> **getExclusions** ()

**getInclusions**

```
public List<String> getInclusions ()
```

**getMappings**

```
public List<KeyValue> getMappings ()
```

**setExclusions**

```
public void setExclusions (List<String> exclusions)
```

Sets exclusions from log - if an event contains a parameter key that is included in the exclusions list, that parameter will not be persisted in the database log.

**Parameters**

- **exclusions** – list of parameters to exclude from log

**setInclusions**

```
public void setInclusions (List<String> inclusions)
```

Sets inclusions to log - if an event contains a parameter key that is included in the inclusions list, that parameter will be persisted in the database log. By default all parameters are included.

**Parameters**

- **inclusions** – list of parameters to include in log

**setMappings**

```
public void setMappings (List<KeyValue> mappings)
```

Sets replacement map for event parameters in the form of a KeyValue list. See [KeyValue](#) for more information.

**Parameters**

- **mappings** – list of KeyValue mappings

**LoggableEvent**

```
public class LoggableEvent
```

Class containing event data. The field `eventSubjects` contains names of subjects for events that should be logged by `org.motechproject.eventlogging.loggers.EventLogger` and flags are used to filter those events.

**Constructors****LoggableEvent**

```
public LoggableEvent (List<String> eventSubjects, List<? extends EventFlag> flags)
```

Create an instance of `LoggableEvent` using passed parameters

**Parameters**

- **eventSubjects** – list of event subjects to log
- **flags** – event flags used for filter events by parameters

## Methods

### getEventSubjects

public [List<String>](#) **getEventSubjects** ()

### getFlags

public [List<? extends EventFlag>](#) **getFlags** ()

### isLoggableEvent

public boolean **isLoggableEvent** (MotechEvent *eventToLog*)

Checks if the event should be logged by `org.motechproject.eventlogging.loggers.EventLogger`.

#### Parameters

- **eventToLog** – incoming motech event to check if should be logged

**Returns** true if event should be logged or false if it should not

### setEventSubjects

public void **setEventSubjects** ([List<String>](#) *eventSubjects*)

### setFlags

public void **setFlags** ([List<? extends EventFlag>](#) *flags*)

## MappingsJson

public class **MappingsJson**

Class representing json configuration file of the logs structure for database logging.

## Methods

### getExcludes

public [List<String>](#) **getExcludes** ()

### getFlags

public [List<ParametersPresentEventFlag>](#) **getFlags** ()

### getIncludes

public [List<String>](#) **getIncludes** ()

### getMappings

public [List<Map<String, String>>](#) **getMappings** ()

### getSubjects

public [List<String>](#) **getSubjects** ()

### setExcludes

public void **setExcludes** ([List<String>](#) *excludes*)

**setFlags**

public void **setFlags** ([List](#)<[ParametersPresentEventFlag](#)> *flags*)

**setIncludes**

public void **setIncludes** ([List](#)<[String](#)> *includes*)

**setMappings**

public void **setMappings** ([List](#)<[Map](#)<[String](#), [String](#)>> *mappings*)

**setSubjects**

public void **setSubjects** ([List](#)<[String](#)> *subjects*)

**ParametersPresentEventFlag**

public class **ParametersPresentEventFlag** implements [EventFlag](#)

Implementation of [org.motechproject.eventlogging.matchers.EventFlag](#) interface. Stores keyValuePairPresent map which is used as a map of flags.

**Constructors****ParametersPresentEventFlag**

public **ParametersPresentEventFlag** ()

Creates an instance of ParametersPresentEventFlag

**ParametersPresentEventFlag**

public **ParametersPresentEventFlag** ([Map](#)<[String](#), [String](#)> *keyValuePairPresent*)

Creates an instance of ParametersPresentEventFlag and sets keyValuePairPresent to the passed parameter or to an empty map if the parameter is null.

**Parameters**

- **keyValuePairPresent** – map representing parameters and their values

**Methods****getKeyValuePairPresent**

public [Map](#)<[String](#), [String](#)> **getKeyValuePairPresent** ()

**passesFlags**

public boolean **passesFlags** ([MotechEvent](#) *eventToLog*)

**setKeyValuePairPresent**

public void **setKeyValuePairPresent** ([Map](#)<[String](#), [String](#)> *keyValuePairPresent*)

**RegexEventFlag**

public class **RegexEventFlag** implements [EventFlag](#)

Implementation of the [org.motechproject.eventlogging.matchers.EventFlag](#) interface.

## Methods

### passesFlags

public boolean **passesFlags** (MotechEvent *event*)

## 19.14.49 org.motechproject.eventlogging.service

### EventLogService

public interface **EventLogService** extends MotechDataService<EventLog>

Motech Data Service interface for `EventLogs`. The implementation is generated and injected by the MDS module.

## Methods

### findBySubject

List<EventLog> **findBySubject** (String *name*)

Finds all recorded events with the given subject.

#### Parameters

- **name** – a subject to filter by

**Returns** a collection of all recorded events with the given subject

### EventLoggingService

public interface **EventLoggingService**

Interface that represents a logging service. A logging service can provide any sort of implementation, but must be able to log events and return a list of logged events.

## Methods

### getLoggedEventSubjects

Set<String> **getLoggedEventSubjects** ()

Returns a set of event subjects this logger is listening on.

**Returns** the list of event subjects

### logEvent

void **logEvent** (MotechEvent *event*)

Logs an event. The ultimate destination of the logged event depends on the installed logging service's `org.motechproject.eventlogging.service.EventLoggingServiceManager`. A default logging service is configured that logs to the database.

#### Parameters

- **event** – the event to log



## EventLoggingServiceManager

public interface **EventLoggingServiceManager**

A logging service manager that allows for registration of logging services. This class is exposed as an OSGi service to support cross-bundle registration.

### Methods

**registerEventLoggingService**

void **registerEventLoggingService** ([EventLoggingService](#) *eventLoggingService*)

Register a logging service in the platform, which will generate listeners for that logging service based on the subjects it listens on.

#### Parameters

- **eventLoggingService** – the logging service to add to the platform

**updateEventLoggingService**

void **updateEventLoggingService** ([EventLoggingService](#) *eventLoggingService*)

Not yet implemented.

#### Parameters

- **eventLoggingService** –

## EventQueryService

public interface **EventQueryService**<T>

Interface to query for stored events.

### Methods

**getAllEventsByParameter**

[List](#)<T> **getAllEventsByParameter** ([String](#) *parameter*, [String](#) *value*)

Queries the database for all events that match a given parameter and key- value pair.

#### Parameters

- **parameter** – the parameter key from the event
- **value** – the parameter value from the event

**Returns** a list of informational objects that match the key-value pair

**getAllEventsBySubject**

[List](#)<T> **getAllEventsBySubject** ([String](#) *subject*)

Queries the database for all events that match a given subject.

#### Parameters

- **subject** – the subject of the event

**Returns** a list of informational objects that match the event subject

**getAllEventsBySubjectAndParameter**

`List<T> getAllEventsBySubjectAndParameter (String subject, String parameter, String value)`

Queries the database for all events that match a given subject and parameter key-value pair.

**Parameters**

- **subject** – the subject of the event.
- **parameter** – the parameter key from the event
- **value** – the parameter value from the event

**Returns** a list of informational objects that match the subject as well as the key-value pair

## 19.14.50 org.motechproject.eventlogging.service.impl

### DbEventLoggingService

public class **DbEventLoggingService** implements `EventLoggingService`

**Constructors****DbEventLoggingService**

public **DbEventLoggingService** ()

**DbEventLoggingService**

public **DbEventLoggingService** (AllEventMappings *allEventMappings*)

**DbEventLoggingService**

public **DbEventLoggingService** (List<`DbEventLogger`> *dbEventLoggers*)

**Methods****getAllEventMappings**

public AllEventMappings **getAllEventMappings** ()

**getDbEventLoggers**

public List<`DbEventLogger`> **getDbEventLoggers** ()

**getDefaultDbEventLogger**

public `DbEventLogger` **getDefaultDbEventLogger** ()

**getDefaultDbToLogConverter**

public `DefaultDbToLogConverter` **getDefaultDbToLogConverter** ()

**getLoggedEventSubjects**

public Set<`String`> **getLoggedEventSubjects** ()

**logEvent**

public void **logEvent** (MotechEvent *event*)

**setAllEventMappings**

public void **setAllEventMappings** (AllEventMappings *allEventMappings*)

**setDbEventLoggers**

public void **setDbEventLoggers** (List<DbEventLogger> *dbEventLoggers*)

**setDefaultDbEventLogger**

public void **setDefaultDbEventLogger** (DbEventLogger *defaultDbEventLogger*)

**setDefaultDbToLogConverter**

public void **setDefaultDbToLogConverter** (DefaultDbToLogConverter *defaultDbToLogConverter*)

**DbEventQueryService**

public class **DbEventQueryService** implements EventQueryService<EventLog>

The DbEventQueryService allows to query for records, given certain criteria.

**Methods****getAllEventsByParameter**

public List<EventLog> **getAllEventsByParameter** (String *parameter*, String *value*)

**getAllEventsBySubject**

public List<EventLog> **getAllEventsBySubject** (String *subject*)

**getAllEventsBySubjectAndParameter**

public List<EventLog> **getAllEventsBySubjectAndParameter** (String *subject*, String *parameter*,  
String *value*)

**EventLoggingServiceManagerImpl**

public class **EventLoggingServiceManagerImpl** implements EventLoggingServiceManager

**Methods****registerDefaultService**

public void **registerDefaultService** ()

**registerEventLoggingService**

public void **registerEventLoggingService** (EventLoggingService *eventLoggingService*)

**updateEventLoggingService**

public void **updateEventLoggingService** (EventLoggingService *eventLoggingService*)

## FileEventLoggingService

public class **FileEventLoggingService** implements [EventLoggingService](#)

### Constructors

#### FileEventLoggingService

public **FileEventLoggingService** ()

#### FileEventLoggingService

public **FileEventLoggingService** ([List](#)<[FileEventLogger](#)> *fileEventLoggers*)

### Methods

#### getLoggedEventSubjects

public [Set](#)<[String](#)> **getLoggedEventSubjects** ()

#### logEvent

public void **logEvent** ([MotechEvent](#) *event*)

## 19.14.51 org.motechproject.http.agent.domain

### Credentials

public class **Credentials**

The `Credentials` class is used to represent user credentials, when a http request requires an authentication

### Constructors

#### Credentials

public **Credentials** ([String](#) *username*, [String](#) *password*)

Constructor.

#### Parameters

- **username** – the username
- **password** – the user password

### Methods

#### getPassword

public [String](#) **getPassword** ()

**Returns** the user password

#### getUsername

public [String](#) **getUsername** ()

**Returns** the username

**setPassword**

public void **setPassword** (*String password*)

**Parameters**

- **password** – the user password

**setUsername**

public void **setUsername** (*String username*)

**Parameters**

- **username** – the username

**EventDataKeys**

public final class **EventDataKeys**

Event data keys.

**Fields****DATA**

public static final *String* **DATA**

**HEADERS**

public static final *String* **HEADERS**

**METHOD**

public static final *String* **METHOD**

**PASSWORD**

public static final *String* **PASSWORD**

**RETRY\_COUNT**

public static final *String* **RETRY\_COUNT**

**RETRY\_INTERVAL**

public static final *String* **RETRY\_INTERVAL**

**URL**

public static final *String* **URL**

**USERNAME**

public static final *String* **USERNAME**

**EventSubjects**

public final class **EventSubjects**

Event subjects.

## Fields

### HTTP\_REQUEST

public static final [String](#) **HTTP\_REQUEST**

### SYNC\_HTTP\_REQUEST\_RET\_TYPE

public static final [String](#) **SYNC\_HTTP\_REQUEST\_RET\_TYPE**

## 19.14.52 org.motechproject.http.agent.service

### HttpAgent

public interface **HttpAgent**

An OSGI Service for sending http requests.

## Methods

### execute

void **execute** ([String](#) *url*, [Object](#) *data*, [Method](#) *method*)

Executes the http request asynchronously.

#### Parameters

- **url** – the url for request
- **data** – the request data
- **method** – the http method

### execute

void **execute** ([String](#) *url*, [Object](#) *data*, [Method](#) *method*, [Map](#)<[String](#), [String](#)> *headers*)

Executes the http request asynchronously.

#### Parameters

- **url** – the url for request
- **data** – the request data
- **method** – the http method
- **headers** – the request headers

### execute

void **execute** ([String](#) *url*, [Object](#) *data*, [Method](#) *method*, [Credentials](#) *credentials*)

Executes the http request asynchronously.

#### Parameters

- **url** – the url for request
- **data** – the request data
- **method** – the http method
- **credentials** – the user credentials

See also: `org.motechproject.http.agent.domain.Credentials`

#### **execute**

void **execute** (`String url`, `Object data`, `Method method`, `Map<String, String> headers`, `Credentials credentials`)

Executes the http request asynchronously.

##### **Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method
- **headers** – the request headers
- **credentials** – the user credentials

See also: `org.motechproject.http.agent.domain.Credentials`

#### **executeSync**

void **executeSync** (`String url`, `Object data`, `Method method`)

Executes the http request synchronously.

##### **Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method

#### **executeSync**

void **executeSync** (`String url`, `Object data`, `Method method`, `Map<String, String> headers`)

Executes the http request synchronously.

##### **Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method
- **headers** – the request headers

#### **executeSync**

void **executeSync** (`String url`, `Object data`, `Method method`, `Credentials credentials`)

Executes the http request synchronously.

##### **Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method
- **credentials** – the user credentials

See also: `org.motechproject.http.agent.domain.Credentials`

**executeSync**

void **executeSync** ([String url](#), [Object data](#), [Method method](#), [Map<String, String> headers](#), [Credentials credentials](#))

Executes the http request synchronously.

**Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method
- **headers** – the request headers
- **credentials** – the user credentials

See also: [org.motechproject.http.agent.domain.Credentials](#)

**executeWithReturnTypeSync**

[ResponseEntity<?>](#) **executeWithReturnTypeSync** ([String url](#), [Object data](#), [Method method](#))

Executes the http request synchronously and returns the response.

**Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method

**Returns** response from posted request

**executeWithReturnTypeSync**

[ResponseEntity<?>](#) **executeWithReturnTypeSync** ([String url](#), [Object data](#), [Method method](#), [Integer retryCount](#))

Executes the http request synchronously and returns the response, takes additional parameter for number of retries.

**Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method
- **retryCount** – the number of retries

**Returns** response from posted request

**executeWithReturnTypeSync**

[ResponseEntity<?>](#) **executeWithReturnTypeSync** ([String url](#), [Object data](#), [Method method](#), [Integer retryCount](#), [Long retryInterval](#))

Executes the http request synchronously and returns the response, takes additional parameters for number of retries and interval between two retries.

**Parameters**

- **url** – the url for request
- **data** – the request data
- **method** – the http method



- **retryCount** – the number of retries
- **retryInterval** – the interval between two retries in milliseconds

**Returns** response from posted request

## Method

public enum **Method**  
Enum which represents http methods.

## Enum Constants

**DELETE**  
public static final [Method](#) **DELETE**  
Represents http DELETE.

**GET**  
public static final [Method](#) **GET**  
Represents http GET.

**POST**  
public static final [Method](#) **POST**  
Represents http POST.

**PUT**  
public static final [Method](#) **PUT**  
Represents http PUT.

## 19.14.53 org.motechproject.hub.exception

### ApplicationErrors

public enum **ApplicationErrors** implements [HubErrors](#)

## Enum Constants

**BAD\_REQUEST**  
public static final [ApplicationErrors](#) **BAD\_REQUEST**

**SUBSCRIPTION\_NOT\_FOUND**  
public static final [ApplicationErrors](#) **SUBSCRIPTION\_NOT\_FOUND**

**TOPIC\_NOT\_FOUND**  
public static final [ApplicationErrors](#) **TOPIC\_NOT\_FOUND**

## HubError

```
public class HubError
```

### Constructors

#### HubError

```
public HubError ()
```

### Methods

#### getApplication

```
public String getApplication ()
```

#### getErrorCode

```
public String getErrorCode ()
```

#### getErrorMessage

```
public String getErrorMessage ()
```

#### getHostName

```
public String getHostName ()
```

#### getTimeStamp

```
public Date getTimeStamp ()
```

#### setApplication

```
public void setApplication (String application)
```

#### setErrorCode

```
public void setErrorCode (String errorCode)
```

#### setErrorMessage

```
public void setErrorMessage (String errorMessage)
```

#### setHostName

```
public void setHostName (String hostName)
```

#### setTimeStamp

```
public void setTimeStamp (Date timeStamp)
```

## HubErrors

```
public interface HubErrors
```

## Methods

### **getCode**

int **getCode** ()

### **getHttpStatus**

HttpStatus **getHttpStatus** ()

### **getMessage**

String **getMessage** ()

## HubException

public class **HubException** extends [Exception](#)

This class is used for custome exception for the Hub Module

**Author** anuranjan

## Constructors

### **HubException**

public **HubException** ([HubErrors](#) *hubErrors*)

### **HubException**

public **HubException** ([HubErrors](#) *hubErrors*, [String](#) *reason*)

### **HubException**

public **HubException** ([HubErrors](#) *hubErrors*, [Throwable](#) *throwable*)

### **HubException**

public **HubException** ([HubErrors](#) *hubErrors*, [Throwable](#) *throwable*, [String](#) *reason*)

## Methods

### **getError**

public [HubErrors](#) **getError** ()

### **getErrorCode**

public int **getErrorCode** ()

### **getErrorMessage**

public [String](#) **getErrorMessage** ()

### **getErrorMessageDetails**

public [String](#) **getErrorMessageDetails** ()

#### **getReason**

public **String** **getReason** ()

#### **setReason**

public void **setReason** (**String** *reason*)

### **RestException**

public class **RestException** extends **RuntimeException**

#### **Constructors**

##### **RestException**

public **RestException** (**HubException** *exception*, **List**<**String**> *errors*)

##### **RestException**

public **RestException** (**HubException** *hubException*, **String** *reason*)

#### **Methods**

##### **getHttpStatus**

public **HttpStatus** **getHttpStatus** ()

##### **getHubException**

public **HubException** **getHubException** ()

##### **getReason**

public **String** **getReason** ()

## **19.14.54 org.motechproject.hub.mds**

### **HubDistributionContent**

public class **HubDistributionContent** implements **java.io.Serializable**

Stores the content and content type received from the publisher when there is an update in a topic

#### **Constructors**

##### **HubDistributionContent**

public **HubDistributionContent** ()

##### **HubDistributionContent**

public **HubDistributionContent** (**String** *content*, **String** *contentType*)

## Methods

### getContent

public [String](#) **getContent** ()

### getContentType

public [String](#) **getContentType** ()

### setContent

public void **setContent** ([String](#) *content*)

### setContentType

public void **setContentType** ([String](#) *contentType*)

## HubDistributionStatus

public class **HubDistributionStatus** implements [java.io.Serializable](#)

This entity stores the status of content distribution from hub to the subscribers

## Constructors

### HubDistributionStatus

public **HubDistributionStatus** ()

### HubDistributionStatus

public **HubDistributionStatus** ([Integer](#) *distributionStatusId*, [String](#) *distributionStatusCode*)

## Methods

### getDistributionStatusCode

public [String](#) **getDistributionStatusCode** ()

### getDistributionStatusId

public [Integer](#) **getDistributionStatusId** ()

### setDistributionStatusCode

public void **setDistributionStatusCode** ([String](#) *distributionStatusCode*)

### setDistributionStatusId

public void **setDistributionStatusId** ([Integer](#) *distributionStatusId*)

## HubPublisherTransaction

public class **HubPublisherTransaction** implements [java.io.Serializable](#)

Stores the details of the transaction in which the publisher notifies the hub about an update in a topic

## Constructors

### HubPublisherTransaction

public **HubPublisherTransaction** ()

### HubPublisherTransaction

public **HubPublisherTransaction** (*Integer hubTopicId*, *Integer contentId*, *DateTime notificationTime*)

## Methods

### getContentId

public *Integer* **getContentId** ()

### getHubTopicId

public *Integer* **getHubTopicId** ()

### getNotificationTime

public *DateTime* **getNotificationTime** ()

### setContentId

public void **setContentId** (*Integer contentId*)

### setHubTopicId

public void **setHubTopicId** (*Integer hubTopicId*)

### setNotificationTime

public void **setNotificationTime** (*DateTime notificationTime*)

## HubSubscriberTransaction

public class **HubSubscriberTransaction** implements *java.io.Serializable*  
Stores the transaction in which the hub distributes the content to its subscribers

## Constructors

### HubSubscriberTransaction

public **HubSubscriberTransaction** ()

### HubSubscriberTransaction

public **HubSubscriberTransaction** (*Integer hubDistributionStatusId*, *Integer hubSubscriptionId*, *Integer contentId*)

## Methods

### getContentId

public *Integer* **getContentId** ()

**getHubDistributionStatusId**

public **Integer** **getHubDistributionStatusId** ()

**getHubSubscriptionId**

public **Integer** **getHubSubscriptionId** ()

**setContentId**

public void **setContentId** (**Integer** *contentId*)

**setHubDistributionStatusId**

public void **setHubDistributionStatusId** (**Integer** *hubDistributionStatusId*)

**setHubSubscriptionId**

public void **setHubSubscriptionId** (**Integer** *hubSubscriptionId*)

**HubSubscription**

public class **HubSubscription** implements **java.io.Serializable**

Stores the subscription details of a topic

**Constructors****HubSubscription**

public **HubSubscription** ()

**HubSubscription**

public **HubSubscription** (**Integer** *hubTopicId*, **Integer** *hubSubscriptionStatusId*, **String** *callbackUrl*)

**HubSubscription**

public **HubSubscription** (**Integer** *hubTopicId*, **Integer** *hubSubscriptionStatusId*, **String** *callbackUrl*, **Integer** *leaseSeconds*, **String** *secret*)

**Methods****getCallbackUrl**

public **String** **getCallbackUrl** ()

**getHubSubscriptionStatusId**

public **Integer** **getHubSubscriptionStatusId** ()

**getHubTopicId**

public **Integer** **getHubTopicId** ()

**getLeaseSeconds**

public **Integer** **getLeaseSeconds** ()

#### **getSecret**

public **String** **getSecret** ()

#### **setCallbackUrl**

public void **setCallbackUrl** (**String** *callbackUrl*)

#### **setHubSubscriptionStatusId**

public void **setHubSubscriptionStatusId** (**Integer** *hubSubscriptionStatusId*)

#### **setHubTopicId**

public void **setHubTopicId** (**Integer** *hubTopicId*)

#### **setLeaseSeconds**

public void **setLeaseSeconds** (**Integer** *leaseSeconds*)

#### **setSecret**

public void **setSecret** (**String** *secret*)

### **HubSubscriptionStatus**

public class **HubSubscriptionStatus** implements **java.io.Serializable**

Stores the status of a subscription

#### **Constructors**

##### **HubSubscriptionStatus**

public **HubSubscriptionStatus** ()

##### **HubSubscriptionStatus**

public **HubSubscriptionStatus** (**Integer** *subscriptionStatusId*, **String** *subscriptionStatusCode*)

#### **Methods**

##### **getSubscriptionStatusCode**

public **String** **getSubscriptionStatusCode** ()

##### **getSubscriptionStatusId**

public **Integer** **getSubscriptionStatusId** ()

##### **setSubscriptionStatusCode**

public void **setSubscriptionStatusCode** (**String** *subscriptionStatusCode*)

##### **setSubscriptionStatusId**

public void **setSubscriptionStatusId** (**Integer** *subscriptionStatusId*)



## HubTopic

public class **HubTopic** implements `java.io.Serializable`  
Stores the topic urls

### Constructors

**HubTopic**  
public **HubTopic** ()

**HubTopic**  
public **HubTopic** (*String* *topicUrl*)

### Methods

**getTopicUrl**  
public *String* **getTopicUrl** ()

**setTopicUrl**  
public void **setTopicUrl** (*String* *topicUrl*)

## 19.14.55 org.motechproject.hub.mds.service

### HubDistributionContentMDSService

public interface **HubDistributionContentMDSService** extends `MotechDataService<HubDistributionContent>`  
This service provides data access for `org.motechproject.hub.domain.HubDistributionContent`.  
The implementation is generated by Motech Data Services and published as an OSGi service.

### HubDistributionStatusMDSService

public interface **HubDistributionStatusMDSService** extends `MotechDataService<HubDistributionStatus>`

### Methods

**findByDistStatus**  
*List<HubDistributionStatus>* **findByDistStatus** (*Integer* *distributionStatusId*)

### HubPublisherTransactionMDSService

public interface **HubPublisherTransactionMDSService** extends `MotechDataService<HubPublisherTransaction>`

### Methods

**findPubTransactionByTopicId**  
*List<HubPublisherTransaction>* **findPubTransactionByTopicId** (*Integer* *hubTopicId*)

### HubSubscriberTransactionMDSService

public interface **HubSubscriberTransactionMDSService** extends MotechDataService<HubSubscriberTransaction>

#### Methods

##### findSubTransBySubId

List<HubSubscriberTransaction> **findSubTransBySubId** (*Integer hubSubscriptionId*)

### HubSubscriptionMDSService

public interface **HubSubscriptionMDSService** extends MotechDataService<HubSubscription>

#### Methods

##### findSubByCallbackUrl

List<HubSubscription> **findSubByCallbackUrl** (*String callbackUrl*)

##### findSubByCallbackUrlAndTopicId

List<HubSubscription> **findSubByCallbackUrlAndTopicId** (*String callbackUrl*, *Integer hubTopicId*)

##### findSubByTopicId

List<HubSubscription> **findSubByTopicId** (*Integer hubTopicId*)

### HubSubscriptionStatusMDSService

public interface **HubSubscriptionStatusMDSService** extends MotechDataService<HubSubscriptionStatus>

#### Methods

##### findBySubStatus

List<HubSubscriptionStatus> **findBySubStatus** (*Integer subscriptionStatusId*)

### HubTopicMDSService

public interface **HubTopicMDSService** extends MotechDataService<HubTopic>

This service provides data access for `org.motechproject.hub.domain.HubTopic`. The implementation is generated by Motech Data Services and published as an OSGi service.

#### Methods

##### findByTopicUrl

List<HubTopic> **findByTopicUrl** (*String topicUrl*)

## 19.14.56 org.motechproject.hub.model

### DistributionStatusLookup

public enum **DistributionStatusLookup**

#### Enum Constants

##### FAILURE

public static final **DistributionStatusLookup FAILURE**

##### SUCCESS

public static final **DistributionStatusLookup SUCCESS**

### HubSettings

public class **HubSettings**

#### Methods

##### canMakeConnection

public boolean **canMakeConnection** ()

##### getHubBaseUrl

public **String** **getHubBaseUrl** ()

##### setHubBaseUrl

public void **setHubBaseUrl** (**String** *hubBaseUrl*)

### Modes

public enum **Modes**

#### Enum Constants

##### PUBLISH

public static final **Modes PUBLISH**

##### SUBSCRIBE

public static final **Modes SUBSCRIBE**

##### UNSUBSCRIBE

public static final **Modes UNSUBSCRIBE**

## SettingsDTO

```
public class SettingsDTO
```

### Constructors

#### SettingsDTO

```
public SettingsDTO ()
```

### Methods

#### getHubBaseUrl

```
public String getHubBaseUrl ()
```

#### setHubBaseUrl

```
public void setHubBaseUrl (String hubBaseUrl)
```

## SubscriptionStatusLookup

```
public enum SubscriptionStatusLookup
```

This enum class defines the valid status codes for subscription status.

**Author** anuranjan

### Enum Constants

#### ACCEPTED

```
public static final SubscriptionStatusLookup ACCEPTED
```

#### INTENT\_FAILED

```
public static final SubscriptionStatusLookup INTENT_FAILED
```

#### INTENT\_VERIFIED

```
public static final SubscriptionStatusLookup INTENT_VERIFIED
```

## 19.14.57 org.motechproject.hub.service

### ContentDistributionService

```
public interface ContentDistributionService
```

### Methods

#### distribute

```
void distribute (String url)
```

## DistributionServiceDelegate

public interface **DistributionServiceDelegate**

This is an interface providing methods to get the Content of an updated topic and distribute it to all the subscribers for that topic

**Author** Anuranjan

### Methods

#### distribute

void **distribute** (*String callbackUrl*, *String content*, *MediaType contentType*, *String topicUrl*)

Distributes the fetched content to all the subscribers subscribed to the particular topic

##### Parameters

- **callbackUrl** –
  - a *String* representing the subscriber's callback URL where notifications should be delivered
- **content** –
  - a *String* representing the full content of the updated topic
- **contentType** –
  - a *String* representing the Content-Type of the topic updated
- **topicUrl** –
  - a *String* representing the URL of the topic which is updated

#### getContent

*ResponseEntity<String>* **getContent** (*String topicUrl*)

Fetches the content from a publisher corresponding to the *topicUrl*

##### Parameters

- **topicUrl** –
  - a *String* representing the topic URL which is updated

## SubscriptionService

public interface **SubscriptionService**

This is an interface which provides business logic for a subscription/unsubscription request

**Author** Anuranjan

### Methods

#### subscribe

void **subscribe** (*String callbackUrl*, *Modes hubMode*, *String topic*, *String leaseSeconds*, *String secret*)

This method executes the business logic for subscribing/unsubscribing to a topic. The subscriber provides the callback URL where the update notification should be sent.

##### Parameters

- **callbackUrl** –
  - a `String` representing the subscriber’s callback URL where notifications should be delivered
- **hubMode** –
  - a `Modes` enum representing the mode “subscribe” or “unsubscribe”, depending on the goal of the request
- **topic** –
  - a `String` representing the topic URL that the subscriber wishes to subscribe to or unsubscribe from
- **leaseSeconds** –
  - a `String` representing the number of seconds for which the subscriber would like to have the subscription active. This is an optional parameter
- **secret** –
  - a `String` provided by the subscriber which will be used to compute an HMAC digest for authorized content distribution. Currently this is not being consumed by the API

**Throws**

- **HubException** –

## 19.14.58 org.motechproject.hub.service.impl

### ContentDistributionServiceImpl

public class **ContentDistributionServiceImpl** implements [ContentDistributionService](#)

#### Constructors

##### ContentDistributionServiceImpl

```
public ContentDistributionServiceImpl (HubTopicMDSService    hubTopicService, HubSubscriptionMDSService    hubSubscriptionMDSService,  
                                     HubPublisherTransactionMDSService hubPublisherTransactionMDSService, HubSubscriberTransactionMDSService hubSubscriberTransactionMDSService,  
                                     HubDistributionContentMDSService hubDistributionContentMDSService)
```

##### ContentDistributionServiceImpl

```
public ContentDistributionServiceImpl ()
```

#### Methods

##### distribute

```
public void distribute (String url)
```

**getDistributionServiceDelegate**

public [DistributionServiceDelegate](#) **getDistributionServiceDelegate** ()

**setDistributionServiceDelegate**

public void **setDistributionServiceDelegate** ([DistributionServiceDelegate](#) *distributionServiceDelegate*)

**[DistributionServiceDelegateImpl](#)**

public class **DistributionServiceDelegateImpl** implements [DistributionServiceDelegate](#)

This class implements the methods in the interface [DistributionServiceDelegate](#)

**Author** Anuranjan

**Constructors****DistributionServiceDelegateImpl**

public **DistributionServiceDelegateImpl** ([HttpAgent](#) *httpAgentImpl*, [SettingsFacade](#) *settingsFacade*)

**Methods****distribute**

public void **distribute** ([String](#) *callbackUrl*, [String](#) *content*, [MediaType](#) *contentType*, [String](#) *topicUrl*)

**getContent**

public [ResponseEntity](#)<[String](#)> **getContent** ([String](#) *topicUrl*)

**getHttpAgentImpl**

public [HttpAgent](#) **getHttpAgentImpl** ()

**setHttpAgentImpl**

public void **setHttpAgentImpl** ([HttpAgent](#) *httpAgentImpl*)

**setRetryCount**

public void **setRetryCount** ([String](#) *retryCount*)

**setRetryInterval**

public void **setRetryInterval** ([String](#) *retryInterval*)

**[SubscriptionServiceImpl](#)**

public class **SubscriptionServiceImpl** implements [SubscriptionService](#)

This is the implementation class of the interface [SubscriptionService](#)

**Author** Anuranjan

## Constructors

### SubscriptionServiceImpl

```
public SubscriptionServiceImpl (HubTopicMDSService hubTopicService, HubSubscriptionMDSService hubSubscriptionMDSService, HttpAgent httpAgentImpl)
```

## Methods

### getHttpAgentImpl

```
public HttpAgent getHttpAgentImpl ()
```

### setHttpAgentImpl

```
public void setHttpAgentImpl (HttpAgent httpAgentImpl)
```

### setRetryCount

```
public void setRetryCount (String retryCount)
```

### setRetryInterval

```
public void setRetryInterval (String retryInterval)
```

### subscribe

```
public void subscribe (String callbackUrl, Modes mode, String topic, String leaseSeconds, String secret)
```

## 19.14.59 org.motechproject.ivr.domain

### CallDetailRecord

```
public class CallDetailRecord
```

The call detail record is the object persisted for every call from and to MOTECH by the IVR module. It represents the details of the call, and is stored in the database for auditing purposes.

## Fields

### CALL\_FAILED

```
public static final String CALL_FAILED
```

### CALL\_INITIATED

```
public static final String CALL_INITIATED
```

## Constructors

### CallDetailRecord

```
public CallDetailRecord ()
```

Creates a new CallDetailRecord and sets the motechTimestamp value to the current datetime.



**CallDetailRecord**

```
public CallDetailRecord(String configName, String providerTimestamp, String from, String to,
    CallDirection callDirection, String callStatus, String templateName, String
    motechCallId, String providerCallId, Map<String, String> providerExtra-
    Data)
```

Creates a new CallDetailRecord and sets the motechTimestamp value to the current datetime.

**Parameters**

- **configName** – the name of the IVR configuration used for the call this record corresponds to
- **providerTimestamp** – the timestamp for this call from the IVR provider
- **from** – the initiator of this call
- **to** – the recipient of this call
- **callDirection** – the direction of this call
- **callStatus** – the status of this call
- **templateName** – the name of the template associated with this call
- **motechCallId** – the id for this call generated by MOTECH
- **providerCallId** – the id for this call generated by the IVR provider
- **providerExtraData** – the extra data coming from the provider

**Methods****equals**

```
public boolean equals (Object o)
```

**getCallDirection**

```
public CallDirection getCallDirection ()
```

**Returns** the direction of this call

**See also:** `org.motechproject.ivr.domain.CallDirection`

**getCallStatus**

```
public String getCallStatus ()
```

Returns the status of this call, which is a string coming from the IVR provider.

**Returns** the status of this call

**getConfigName**

```
public String getConfigName ()
```

**Returns** the name of the IVR configuration used for the call this record corresponds to

**getCurrentTimestamp**

```
public static String getCurrentTimestamp ()
```

Returns the current datetime as a string formatted using the datetime formatter used for timestamps in call detail records. The format is yyyy-MM-dd HH:mm:ss.SSSS.

**Returns** the current time in the call detail record timestamp format

**getFrom**

```
public String getFrom()
```

Returns the initiator of the call represented by this record. This usually will be the number of the phone that initiated the call.

**Returns** the initiator of the call

**getId**

```
public long getId()
```

**Returns** the id of the record (auto-generated)

**getMotechCallId**

```
public String getMotechCallId()
```

Returns the ID generated on the MOTECH side for this call. Motech will insert an UUID here.

**Returns** the call ID generated by MOTECH

**getMotechTimestamp**

```
public String getMotechTimestamp()
```

Returns the timestamp for this record. The format is yyyy-MM-dd HH:mm:ss.SSSS.

**Returns** the timestamp for this record

**getProviderCallId**

```
public String getProviderCallId()
```

**Returns** the ID for this call coming from the provider

**getProviderExtraData**

```
public Map<String, String> getProviderExtraData()
```

Returns the extra data for this call detail record. Everything that comes from the provider and does not match any mapping to the fields of CallDetailRecord, will be put into the the provider extra data map.

**Returns** the extra data from the provider

**getProviderTimestamp**

```
public String getProviderTimestamp()
```

**Returns** the timestamp for this call from the IVR provider

**getTemplateName**

```
public String getTemplateName()
```

**Returns** the name of the template associated with this call

**getTo**

```
public String getTo()
```

Returns the recipient of the call represented by this record. This usually will the phone number that was called.

**Returns** the recipient of this call

**hashCode**

```
public int hashCode ()
```

**setConfigName**

```
public void setConfigName (String configName)
```

**Parameters**

- **configName** – the name of the IVR configuration used for the call this record corresponds to

**setField**

```
public void setField (String key, String val)
```

When receiving call detail information from an IVR provider the specific call details must be mapped from what the provider sends back to MOTECH and a CallDetailRecord object. This method will find which field on the given callDetailRecord matches the given key and set service to the given value. If there is no matching CallDetailRecord field, then the key/value pair is added to the providerExtraData map field.

**Parameters**

- **key** – the name of the field to set
- **val** – the value to set

**setFrom**

```
public void setFrom (String from)
```

Sets the initiator of the call represented by this record. This usually will be the phone number that initiated the call.

**Parameters**

- **from** – the initiator of the call

**setMotechTimestamp**

```
public void setMotechTimestamp (String motechTimestamp)
```

Sets the timestamp for this record. The expected format is `yyyy-MM-dd HH:mm:ss.SSSS`.

**Parameters**

- **motechTimestamp** – the timestamp for this record

**setProviderTimestamp**

```
public void setProviderTimestamp (String providerTimestamp)
```

**Parameters**

- **providerTimestamp** – the timestamp for this call from the IVR provider

**setTemplateName**

```
public void setTemplateName (String templateName)
```

**Parameters**

- **templateName** – the name of the template associated with this call

**setTo**

```
public void setTo (String to)
```

Sets the recipient of the call represented by this record. This usually will be the phone number that was called.

**Parameters**

- **to** – the recipient of this call

**toString**

```
public String toString ()
```

**CallDirection**

```
public enum CallDirection
```

Two types of call directions: INBOUND, MO, Mobile Originated OUTBOUND, MT, Mobile Terminated

**Enum Constants****INBOUND**

```
public static final CallDirection INBOUND
```

**OUTBOUND**

```
public static final CallDirection OUTBOUND
```

**UNKNOWN**

```
public static final CallDirection UNKNOWN
```

**Config**

```
public class Config
```

IVR provider configuration, represents how the IVR module interacts with an IVR provider

**Constructors****Config**

```
public Config (String name, boolean authRequired, String username, String password, List<String> ignored-  
StatusFields, String statusFieldMapString, String servicesMapString, HttpMethod outgoing-  
CallMethod, boolean jsonRequest, String outgoingCallUriTemplate, boolean jsonResponse,  
List<String> jsonExtraParamsList)
```

**Methods****equals**

```
public boolean equals (Object o)
```

**getIgnoredStatusFields**

```
public List<String> getIgnoredStatusFields ()
```

**getJsonExtraParamsList**

```
public List<String> getJsonExtraParamsList ()
```

**getName**

```
public String getName ()
```

**getOutgoingCallMethod**

```
public HttpMethod getOutgoingCallMethod ()
```

**getOutgoingCallUriTemplate**

```
public String getOutgoingCallUriTemplate ()
```

**getPassword**

```
public String getPassword ()
```

**getServicesMap**

```
public Map<String, String> getServicesMap ()
```

**getServicesMapString**

```
public String getServicesMapString ()
```

**getStatusFieldMapString**

```
public String getStatusFieldMapString ()
```

**getUsername**

```
public String getUsername ()
```

**hashCode**

```
public int hashCode ()
```

**isAuthRequired**

```
public boolean isAuthRequired ()
```

**isJsonRequest**

```
public boolean isJsonRequest ()
```

**isJsonResponse**

```
public boolean isJsonResponse ()
```

**mapStatusField**

```
public String mapStatusField (String fieldName)
```

When ping Motech back to provide call status, IVR providers sometimes send fields with different names than those that are used by the system. For example the originating number is sometimes provided as 'callerid' whereas Motech uses the name 'from'. The statusFieldMap config field contains such a mapping of field names. And mapStatusField() returns that mapping or the original field name if no mapping exists.

**Parameters**

- **fieldName** –

**setAuthRequired**

```
public void setAuthRequired (boolean authRequired)
```

**setIgnoredStatusFields**

```
public void setIgnoredStatusFields (List<String> ignoredStatusFields)
```

**setJsonExtraParamsList**

```
public void setJsonExtraParamsList (List<String> jsonExtraParamsList)
```

**setJsonRequest**

```
public void setJsonRequest (boolean jsonRequest)
```

**setJsonResponse**

```
public void setJsonResponse (boolean jsonResponse)
```

**setName**

```
public void setName (String name)
```

**setOutgoingCallMethod**

```
public void setOutgoingCallMethod (HttpMethod outgoingCallMethod)
```

**setOutgoingCallUriTemplate**

```
public void setOutgoingCallUriTemplate (String outgoingCallUriTemplate)
```

**setPassword**

```
public void setPassword (String password)
```

**setServicesMapString**

```
public void setServicesMapString (String servicesMapString)
```

**setStatusFieldMapString**

```
public void setStatusFieldMapString (String statusFieldMapString)
```

**setUsername**

```
public void setUsername (String username)
```

**shouldIgnoreField**

public boolean **shouldIgnoreField** (*String fieldName*)

Quick way of knowing if a field supplied by the IVR provider in the status callback should be ignored

**Returns** true if the given field should be ignored, false otherwise

**toString**

public *String* **toString** ()

**HttpMethod**

public enum **HttpMethod**

The HTTP call to trigger an outgoing call from the IVR provider can be either GET or POST

**Enum Constants****GET**

public static final *HttpMethod* **GET**

**POST**

public static final *HttpMethod* **POST**

**Template**

public class **Template**

Represents a user-defined template, basically a string identified by a name which typically will be CCXML/VXML to be returned to an IVR provider. See <https://velocity.apache.org/> for the template language rules.

**Constructors****Template**

public **Template** (*String name*, *String value*)

**Methods****equals**

public boolean **equals** (*Object o*)

**getName**

public *String* **getName** ()

**getValue**

public *String* **getValue** ()

**hashCode**

public int **hashCode** ()

#### **setName**

public void **setName** (*String name*)

#### **setValue**

public void **setValue** (*String value*)

#### **toString**

public *String* **toString** ()

### 19.14.60 org.motechproject.ivr.exception

#### **ConfigNotFoundException**

public class **ConfigNotFoundException** extends *RuntimeException*

The *ConfigNotFoundException* exception signals a situation when an IVR configuration with a given name cannot be found.

#### **Constructors**

##### **ConfigNotFoundException**

public **ConfigNotFoundException** (*String message*)

##### **ConfigNotFoundException**

public **ConfigNotFoundException** (*String message, Throwable cause*)

#### **IvrTemplateException**

public class **IvrTemplateException** extends *RuntimeException*

The *IvrTemplateException* is a custom exception that can be thrown from templates. Using this exception you can control the error code in HTTP responses.

#### **Constructors**

##### **IvrTemplateException**

public **IvrTemplateException** (*String message*)

##### **IvrTemplateException**

public **IvrTemplateException** (*String message, HttpStatus errorCode*)

#### **Methods**

##### **getErrorCode**

public *HttpStatus* **getErrorCode** ()



### TemplateNotFoundException

public class **TemplateNotFoundException** extends [RuntimeException](#)

The `TemplateNotFoundException` exception signals a situation when a Velocity template with a given name cannot be found.

#### Constructors

##### TemplateNotFoundException

public **TemplateNotFoundException** ([String message](#))

##### TemplateNotFoundException

public **TemplateNotFoundException** ([String message](#), [Throwable cause](#))

## 19.14.61 org.motechproject.ivr.repository

### CallDetailRecordDataService

public interface **CallDetailRecordDataService** extends [MotechDataService](#)<[CallDetailRecord](#)>

MDS generated `CallDetailRecord` database queries

#### Methods

##### findByMotechCallId

[List](#)<[CallDetailRecord](#)> **findByMotechCallId** ([String motechCallId](#))

##### findByProviderCallId

[List](#)<[CallDetailRecord](#)> **findByProviderCallId** ([String providerCallId](#))

##### findByProviderCallIdAndCallStatus

[List](#)<[CallDetailRecord](#)> **findByProviderCallIdAndCallStatus** ([String providerCallId](#), [String call-Status](#))

## 19.14.62 org.motechproject.ivr.service

### CallInitiationException

public class **CallInitiationException** extends [RuntimeException](#)

Thrown when an error occurs while trying to initiate an outgoing call

#### Constructors

##### CallInitiationException

public **CallInitiationException** ([String message](#))

##### CallInitiationException

public **CallInitiationException** ([String message](#), [Throwable cause](#))

## ConfigService

public interface **ConfigService**

Config service, manages IVR configs. A `org.motechproject.ivr.domain.Config` represents the way to interact with an IVR provider. See `org.motechproject.ivr.domain.Config`

### Methods

#### allConfigs

`List<Config> allConfigs ()`

Returns all stored configurations.

**Returns** a list of configurations

#### getConfig

`Config getConfig (String name)`

Retrieves the config with the given name.

##### Parameters

- **name** – the name of the configuration

##### Throws

- **`org.motechproject.ivr.exception.ConfigNotFoundException`** – if a config with the given name does not exist

**Returns** the config found for the given name, never null

#### hasConfig

`boolean hasConfig (String name)`

Checks whether a configuration with the given name exists.

##### Parameters

- **name** – the config name to check

**Returns** true if a configuration with the given name exists, false otherwise

#### updateConfigs

`void updateConfigs (List<Config> configs)`

Updates the configuration. The old configuration collection is overwritten by the new one.

##### Parameters

- **configs** – the collection of configurations to save

## OutboundCallService

public interface **OutboundCallService**

Service Interface used to initiate an outgoing (Mobile Terminated) call

## Methods

### initiateCall

void **initiateCall** (*String configName*, *Map<String, String> params*)

Constructs & executes an HTTP GET request, replacing [xxx] placeholders in the outgoingCallUrl string with their provided values from params and adding the unmatched params to the request.

#### Parameters

- **configName** – the name of the configuration to use when initiating the call
- **params** – the parameters that will be used for building the url, the ones that don't match the placeholders will be sent as params of the request

## TemplateService

public interface **TemplateService**

Template service, manages IVR templates. A `org.motechproject.ivr.domain.Template` usually contains text to be returned to the IVR provider as CCXML/VXML. See `org.motechproject.ivr.domain.Template`

## Methods

### allTemplates

*List<Template>* **allTemplates** ()

Retrieves all templates stored in the configuration system.

**Returns** the list of all templates

### getTemplate

*Template* **getTemplate** (*String name*)

Retrieves the template with the given name.

#### Parameters

- **name** – the name of the template to retrieve

#### Throws

- **`org.motechproject.ivr.exception.TemplateNotFoundException`** – if the template with the given name does not exist

**Returns** the template with the given name, never null

### hasTemplate

boolean **hasTemplate** (*String name*)

Checks whether the template with the given name exists.

#### Parameters

- **name** – the name to check

**Returns** true if the template with the given name exists, false otherwise

**updateTemplates**

void **updateTemplates** (*List<Template> templates*)

Updates the templates. The old templates collection is overwritten by the new one.

**Parameters**

- **templates** – the collection of configurations to save

## 19.14.63 org.motechproject.messagecampaign

**Constants**

public final class **Constants**

**Fields**

**DAILY\_REPEATING\_DAYS\_CRON\_EXPRESSION**

public static final *String* **DAILY\_REPEATING\_DAYS\_CRON\_EXPRESSION**

**REPEATING\_DEFAULT\_START\_OFFSET**

public static final int **REPEATING\_DEFAULT\_START\_OFFSET**

**REPEATING\_PROGRAM\_24HRS\_MESSAGE\_DISPATCH\_STRATEGY**

public static final *String* **REPEATING\_PROGRAM\_24HRS\_MESSAGE\_DISPATCH\_STRATEGY**

**UI\_CONFIG**

public static final *String* **UI\_CONFIG**

**EventKeys**

public final class **EventKeys**

A class grouping constants representing the keys used by the message campaign module.

**Fields**

**BASE\_SUBJECT**

public static final *String* **BASE\_SUBJECT**

**CAMPAIGN\_COMPLETED**

public static final *String* **CAMPAIGN\_COMPLETED**

**CAMPAIGN\_MESSAGE\_RECORD\_UPDATED**

public static final *String* **CAMPAIGN\_MESSAGE\_RECORD\_UPDATED**

**CAMPAIGN\_NAME\_KEY**

public static final *String* **CAMPAIGN\_NAME\_KEY**

**CAMPAIGN\_RECORD\_UPDATED**

public static final [String](#) **CAMPAIGN\_RECORD\_UPDATED**

**ENROLLED\_USER\_SUBJECT**

public static final [String](#) **ENROLLED\_USER\_SUBJECT**

**ENROLL\_USER\_SUBJECT**

public static final [String](#) **ENROLL\_USER\_SUBJECT**

**EXTERNAL\_ID\_KEY**

public static final [String](#) **EXTERNAL\_ID\_KEY**

**GENERATED\_MESSAGE\_KEY**

public static final [String](#) **GENERATED\_MESSAGE\_KEY**

**MESSAGE\_FORMATS**

public static final [String](#) **MESSAGE\_FORMATS**

**MESSAGE\_KEY**

public static final [String](#) **MESSAGE\_KEY**

**MESSAGE\_LANGUAGES**

public static final [String](#) **MESSAGE\_LANGUAGES**

**MESSAGE\_NAME\_KEY**

public static final [String](#) **MESSAGE\_NAME\_KEY**

**REFERENCE\_DATE**

public static final [String](#) **REFERENCE\_DATE**

**SCHEDULE\_JOB\_ID\_KEY**

public static final [String](#) **SCHEDULE\_JOB\_ID\_KEY**

**SEND\_MESSAGE**

public static final [String](#) **SEND\_MESSAGE**

**START\_TIME**

public static final [String](#) **START\_TIME**

**UNENROLLED\_USER\_SUBJECT**

public static final [String](#) **UNENROLLED\_USER\_SUBJECT**

**UNENROLL\_USER\_SUBJECT**

public static final [String](#) **UNENROLL\_USER\_SUBJECT**

## 19.14.64 org.motechproject.messagecampaign.contract

### CampaignRequest

```
public class CampaignRequest  
    ingroup MessageCampaign
```

This is the document to enroll an entity into a campaign

### Constructors

#### CampaignRequest

```
public CampaignRequest ()
```

#### CampaignRequest

```
public CampaignRequest (String externalId, String campaignName, LocalDate referenceDate, Time start-  
                                Time)
```

Creates a request for enrollment into a campaign. Holds all fields required for the enrollment.

#### Parameters

- **externalId** –
  - a client defined id to identify the enrollment
- **campaignName** –
  - the campaign into which the entity should be enrolled
- **referenceDate** –
  - the date the campaign has started for this enrollment. it can be in the past resulting in a delayed enrollment.
- **startTime** –
  - time of the day at which the alert must be raised. This overrides the campaign's deliver-Time.

### Methods

#### campaignName

```
public String campaignName ()
```

#### deliverTime

```
public Time deliverTime ()
```

#### externalId

```
public String externalId ()
```

#### referenceDate

```
public LocalDate referenceDate ()
```

**setCampaignName**

public void **setCampaignName** (*String campaignName*)

**setExternalId**

public void **setExternalId** (*String externalId*)

**setReferenceDate**

public void **setReferenceDate** (*LocalDate referenceDate*)

**setStartTime**

public void **setStartTime** (*Time startTime*)

**toString**

public *String* **toString** ()

## 19.14.65 org.motechproject.messagecampaign.dao

### CampaignEnrollmentDataService

public interface **CampaignEnrollmentDataService** extends *MotechDataService<CampaignEnrollment>*  
Data Service interface for *CampaignEnrollments*. The implementation is generated by the Motech Data Services module.

#### Methods

**findByCampaignName**

*List<CampaignEnrollment>* **findByCampaignName** (*String campaignName*)

Finds all *CampaignEnrollments* of the given campaign name.

**Parameters**

- **campaignName** – name of the campaign

**Returns** all campaign enrollments of the given campaign name

**findByExternalId**

*List<CampaignEnrollment>* **findByExternalId** (*String externalId*)

Finds all *CampaignEnrollments* of the given external ID.

**Parameters**

- **externalId** – the external ID of the campaign enrollment

**Returns** all campaign enrollments with the given external ID

**findByExternalIdAndCampaignName**

*CampaignEnrollment* **findByExternalIdAndCampaignName** (*String externalId*, *String campaign-*  
*Name*)

Finds the *CampaignEnrollment* of the given external ID and campaign name.

**Parameters**

- **externalId** – the externalId of the campaign enrollment
- **campaignName** – name of the campaign

**Returns** the campaign enrollment of the given external ID and campaign name or null, if not found

#### findByStatus

List<CampaignEnrollment> **findByStatus** (CampaignEnrollmentStatus *status*)

Finds all CampaignEnrollments of the given enrollment status.

##### Parameters

- **status** – the CampaignEnrollmentStatus of the campaign enrollment

**Returns** all campaign enrollments of the given enrollment status

### CampaignMessageRecordService

public interface **CampaignMessageRecordService** extends MotechDataService<CampaignMessageRecord>  
Data Service interface for CampaignMessageRecords. The implementation is generated by the Motech Data Services module.

#### Methods

##### findByName

List<CampaignMessageRecord> **findByName** (String *name*)

Finds all CampaignMessageRecords with the given name.

##### Parameters

- **name** – the name of the campaign message record

**Returns** campaign message records with the given name

##### findByNameAndType

List<CampaignMessageRecord> **findByNameAndType** (CampaignType *messageType*, String *name*)

Finds all CampaignMessageRecords with the given name and of the given type.

##### Parameters

- **messageType** – the CampaignType of the campaign message records
- **name** – the name of the campaign message records

**Returns** campaign message records of the given name and type

### CampaignRecordService

public interface **CampaignRecordService** extends MotechDataService<CampaignRecord>  
Data Service interface for CampaignRecords. The implementation is generated by the Motech Data Services module.



## Methods

### findByName

`CampaignRecord` **findByName** (*String name*)  
Finds the `CampaignRecord` by its name.

#### Parameters

- **name** – unique name of the campaign record

**Returns** campaign record of the given name

## 19.14.66 org.motechproject.messagecampaign.domain.campaign

### AbsoluteCampaign

public class **AbsoluteCampaign** extends `Campaign<AbsoluteCampaignMessage>`

#### Constructors

##### AbsoluteCampaign

public **AbsoluteCampaign** ()

##### AbsoluteCampaign

public **AbsoluteCampaign** (*String name*, *List<AbsoluteCampaignMessage> messages*)

## Methods

### getCampaignMessage

public `AbsoluteCampaignMessage` **getCampaignMessage** (*CampaignMessageRecord messageRecord*)

### AbsoluteCampaignMessage

public class **AbsoluteCampaignMessage** extends `CampaignMessage`

#### Constructors

##### AbsoluteCampaignMessage

public **AbsoluteCampaignMessage** (*CampaignMessageRecord messageRecord*)

##### AbsoluteCampaignMessage

public **AbsoluteCampaignMessage** (*Time startTime*, *LocalDate date*)

##### AbsoluteCampaignMessage

public **AbsoluteCampaignMessage** (*String name*, *List<String> formats*, *List<String> languages*, *String messageKey*, *Time startTime*, *LocalDate date*)

## Methods

### **getDate**

public [LocalDate](#) **getDate** ()

### **setDate**

public void **setDate** ([LocalDate](#) *date*)

### **validate**

public void **validate** ()

## Campaign

public abstract class **Campaign**<T extends CampaignMessage>

## Constructors

### **Campaign**

public **Campaign** ()

### **Campaign**

public **Campaign** ([String](#) *name*, [List](#)<T> *messages*)

### **Campaign**

protected **Campaign** ([String](#) *name*, [List](#)<T> *messages*, [Period](#) *maxDuration*)

## Methods

### **getCampaignMessage**

public abstract T **getCampaignMessage** ([CampaignMessageRecord](#) *messageRecord*)

### **getMaxDuration**

public [Period](#) **getMaxDuration** ()

### **getMessages**

public [List](#)<T> **getMessages** ()

### **getName**

public [String](#) **getName** ()

### **setMaxDuration**

public void **setMaxDuration** ([Period](#) *maxDuration*)

### **setMaxDuration**

public void **setMaxDuration** ([String](#) *maxDuration*)

**setMessageRecords**

public void **setMessageRecords** ([List](#)<[CampaignMessageRecord](#)> *messageRecords*)

**setMessage**

public void **setMessage** ([List](#)<[T](#)> *messages*)

**setName**

public void **setName** ([String](#) *name*)

**validate**

public void **validate** ()

**CampaignEnrollment**

public class **CampaignEnrollment**

Represents an enrollment for a campaign

**Constructors****CampaignEnrollment**

public **CampaignEnrollment** ([String](#) *externalId*, [String](#) *campaignName*)

**Methods****copyFrom**

public [CampaignEnrollment](#) **copyFrom** ([CampaignEnrollment](#) *enrollment*)

**getCampaignName**

public [String](#) **getCampaignName** ()

**getDeliverTime**

public [Time](#) **getDeliverTime** ()

**getExternalId**

public [String](#) **getExternalId** ()

**getId**

public [Long](#) **getId** ()

**getReferenceDate**

public [LocalDate](#) **getReferenceDate** ()

**getStatus**

public [CampaignEnrollmentStatus](#) **getStatus** ()

#### **isActive**

public boolean **isActive** ()

#### **setCampaignName**

public void **setCampaignName** (*String campaignName*)

#### **setDeliverTime**

public void **setDeliverTime** (*Time deliverTime*)

#### **setExternalId**

public void **setExternalId** (*String externalId*)

#### **setId**

public void **setId** (*Long id*)

#### **setReferenceDate**

public void **setReferenceDate** (*LocalDate referenceDate*)

#### **setStatus**

public void **setStatus** (*CampaignEnrollmentStatus status*)

### **CampaignEnrollmentStatus**

public enum **CampaignEnrollmentStatus**

#### **Enum Constants**

##### **ACTIVE**

public static final *CampaignEnrollmentStatus* **ACTIVE**

##### **COMPLETED**

public static final *CampaignEnrollmentStatus* **COMPLETED**

##### **INACTIVE**

public static final *CampaignEnrollmentStatus* **INACTIVE**

### **CampaignMessage**

public abstract class **CampaignMessage**

#### **Constructors**

##### **CampaignMessage**

public **CampaignMessage** (*CampaignMessageRecord messageRecord*)

**CampaignMessage**

```
public CampaignMessage (String name, List<String> formats, List<String> languages, String messageKey,  
                        Time startTime)
```

**Methods****getCampaign**

```
public CampaignRecord getCampaign ()
```

**getFormats**

```
public List<String> getFormats ()
```

**getLanguages**

```
public List<String> getLanguages ()
```

**getMessageKey**

```
public String getMessageKey ()
```

**getName**

```
public String getName ()
```

**getStartTime**

```
public Time getStartTime ()
```

**setCampaign**

```
public void setCampaign (CampaignRecord campaign)
```

**setFormats**

```
public void setFormats (List<String> formats)
```

**setLanguages**

```
public void setLanguages (List<String> languages)
```

**setMessageKey**

```
public void setMessageKey (String messageKey)
```

**setName**

```
public void setName (String name)
```

**setStartTime**

```
public void setStartTime (Time startTime)
```

**setStartTime**

```
public void setStartTime (String startTime)
```

### **validate**

public abstract void **validate** ()

## **CampaignMessageRecord**

public class **CampaignMessageRecord**

### **Methods**

#### **equalTimes**

public boolean **equalTimes** (*CampaignMessageRecord other*)

#### **equals**

public boolean **equals** (*Object o*)

#### **getCampaign**

public *CampaignRecord* **getCampaign** ()

#### **getCron**

public *String* **getCron** ()

#### **getDate**

public *LocalDate* **getDate** ()

#### **getFormats**

public *List<String>* **getFormats** ()

#### **getId**

public *Long* **getId** ()

#### **getLanguages**

public *List<String>* **getLanguages** ()

#### **getMessageKey**

public *String* **getMessageKey** ()

#### **getMessageType**

public *CampaignType* **getMessageType** ()

#### **getName**

public *String* **getName** ()

#### **getRepeatEvery**

public *String* **getRepeatEvery** ()

**getRepeatOn**

```
public List<DayOfWeek> getRepeatOn ()
```

**getStartTime**

```
public String getStartTime ()
```

**getTimeOffset**

```
public String getTimeOffset ()
```

**hashCode**

```
public int hashCode ()
```

**setCampaign**

```
public void setCampaign (CampaignRecord campaign)
```

**setCron**

```
public void setCron (String cron)
```

**setDate**

```
public void setDate (LocalDate date)
```

**setFormats**

```
public void setFormats (List<String> formats)
```

**setId**

```
public void setId (Long id)
```

**setLanguages**

```
public void setLanguages (List<String> languages)
```

**setMessageKey**

```
public void setMessageKey (String messageKey)
```

**setMessageType**

```
public void setMessageType (CampaignType messageType)
```

**setName**

```
public void setName (String name)
```

**setRepeatEvery**

```
public void setRepeatEvery (String repeatEvery)
```

**setRepeatOn**

```
public void setRepeatOn (List<DayOfWeek> repeatOn)
```

#### **setStartTime**

public void **setStartTime** (*String startTime*)

#### **setTimeOffset**

public void **setTimeOffset** (*String timeOffset*)

#### **toString**

public *String* **toString** ()

### **CampaignRecord**

public class **CampaignRecord**

#### **Methods**

#### **equals**

public boolean **equals** (*Object o*)

#### **getCampaignType**

public *CampaignType* **getCampaignType** ()

#### **getId**

public *Long* **getId** ()

#### **getMaxDuration**

public *String* **getMaxDuration** ()

#### **getMessages**

public *List*<*CampaignMessageRecord*> **getMessages** ()

#### **getName**

public *String* **getName** ()

#### **hashCode**

public int **hashCode** ()

#### **setCampaignType**

public void **setCampaignType** (*CampaignType type*)

#### **setId**

public void **setId** (*Long id*)

#### **setMaxDuration**

public void **setMaxDuration** (*String maxDuration*)



**setMessages**

```
public void setMessages (List<CampaignMessageRecord> messages)
```

**setName**

```
public void setName (String name)
```

**toCampaign**

```
public Campaign toCampaign ()
```

**toString**

```
public String toString ()
```

**updateFrom**

```
public void updateFrom (CampaignRecord other)
```

**CampaignType**

```
public enum CampaignType
```

**Enum Constants****ABSOLUTE**

```
public static final CampaignType ABSOLUTE
```

**CRON**

```
public static final CampaignType CRON
```

**DAY\_OF\_WEEK**

```
public static final CampaignType DAY_OF_WEEK
```

**OFFSET**

```
public static final CampaignType OFFSET
```

**REPEAT\_INTERVAL**

```
public static final CampaignType REPEAT_INTERVAL
```

**CronBasedCampaign**

```
public class CronBasedCampaign extends Campaign<CronBasedCampaignMessage>
```

**Constructors****CronBasedCampaign**

```
public CronBasedCampaign ()
```

### CronBasedCampaign

public **CronBasedCampaign** (*String name*, *List<CronBasedCampaignMessage> messages*)

### CronBasedCampaign

public **CronBasedCampaign** (*String name*, *List<CronBasedCampaignMessage> messages*, *Period maxDuration*)

### Methods

#### getCampaignMessage

public **CronBasedCampaignMessage** **getCampaignMessage** (*CampaignMessageRecord messageRecord*)

### CronBasedCampaignMessage

public class **CronBasedCampaignMessage** extends *CampaignMessage*

### Constructors

#### CronBasedCampaignMessage

public **CronBasedCampaignMessage** (*CampaignMessageRecord messageRecord*)

#### CronBasedCampaignMessage

public **CronBasedCampaignMessage** (*String cron*)

#### CronBasedCampaignMessage

public **CronBasedCampaignMessage** (*String name*, *List<String> formats*, *List<String> languages*, *String messageKey*, *String cron*)

### Methods

#### getCron

public *String* **getCron** ()

#### setCron

public void **setCron** (*String cron*)

#### validate

public void **validate** ()

### DayOfWeek

public enum **DayOfWeek**

## Enum Constants

### Friday

public static final `DayOfWeek` **Friday**

### Monday

public static final `DayOfWeek` **Monday**

### Saturday

public static final `DayOfWeek` **Saturday**

### Sunday

public static final `DayOfWeek` **Sunday**

### Thursday

public static final `DayOfWeek` **Thursday**

### Tuesday

public static final `DayOfWeek` **Tuesday**

### Wednesday

public static final `DayOfWeek` **Wednesday**

## DayOfWeekCampaign

public class **DayOfWeekCampaign** extends `Campaign<DayOfWeekCampaignMessage>`

## Constructors

### DayOfWeekCampaign

public **DayOfWeekCampaign** ()

### DayOfWeekCampaign

public **DayOfWeekCampaign** (`String` *name*, `List<DayOfWeekCampaignMessage>` *messages*, `Period` *maxDuration*)

## Methods

### getCampaignMessage

public `DayOfWeekCampaignMessage` **getCampaignMessage** (`CampaignMessageRecord` *messageRecord*)

### validate

public void **validate** ()

## DayOfWeekCampaignMessage

public class **DayOfWeekCampaignMessage** extends [CampaignMessage](#)

### Constructors

#### DayOfWeekCampaignMessage

public **DayOfWeekCampaignMessage** ([CampaignMessageRecord](#) *messageRecord*)

#### DayOfWeekCampaignMessage

public **DayOfWeekCampaignMessage** (Time *startTime*, List<[DayOfWeek](#)> *daysOfWeek*)

#### DayOfWeekCampaignMessage

public **DayOfWeekCampaignMessage** (String *name*, List<String> *formats*, List<String> *languages*, String *messageKey*, Time *startTime*, List<[DayOfWeek](#)> *daysOfWeek*)

### Methods

#### getDaysOfWeek

public List<[DayOfWeek](#)> **getDaysOfWeek** ()

#### setDaysOfWeek

public void **setDaysOfWeek** (List<[DayOfWeek](#)> *daysOfWeek*)

#### validate

public void **validate** ()

## OffsetCampaign

public class **OffsetCampaign** extends [Campaign](#)<[OffsetCampaignMessage](#)>

### Constructors

#### OffsetCampaign

public **OffsetCampaign** ()

#### OffsetCampaign

public **OffsetCampaign** (String *name*, List<[OffsetCampaignMessage](#)> *messages*)

#### OffsetCampaign

public **OffsetCampaign** (String *name*, List<[OffsetCampaignMessage](#)> *messages*, Period *maxDuration*)

### Methods

#### getCampaignMessage

public [OffsetCampaignMessage](#) **getCampaignMessage** ([CampaignMessageRecord](#) *messageRecord*)

## OffsetCampaignMessage

public class **OffsetCampaignMessage** extends [CampaignMessage](#)

### Constructors

#### OffsetCampaignMessage

public **OffsetCampaignMessage** ([CampaignMessageRecord](#) *messageRecord*)

#### OffsetCampaignMessage

public **OffsetCampaignMessage** (Time *startTime*)

#### OffsetCampaignMessage

public **OffsetCampaignMessage** (Time *startTime*, [Period](#) *timeOffset*)

#### OffsetCampaignMessage

public **OffsetCampaignMessage** (String *name*, List<String> *formats*, List<String> *languages*, String *messageKey*, Time *startTime*, [Period](#) *timeOffset*)

### Methods

#### getTimeOffset

public [Period](#) **getTimeOffset** ()

#### setTimeOffset

public final void **setTimeOffset** ([Period](#) *timeOffset*)

#### setTimeOffset

public final void **setTimeOffset** (String *timeOffset*)

#### validate

public void **validate** ()

## RepeatIntervalCampaign

public class **RepeatIntervalCampaign** extends [Campaign](#)<[RepeatIntervalCampaignMessage](#)>

### Constructors

#### RepeatIntervalCampaign

public **RepeatIntervalCampaign** ()

#### RepeatIntervalCampaign

public **RepeatIntervalCampaign** (String *name*, List<[RepeatIntervalCampaignMessage](#)> *messages*)

**RepeatIntervalCampaign**

```
public RepeatIntervalCampaign (String name, List<RepeatIntervalCampaignMessage> messages, Period maxDuration)
```

**Methods****getCampaignMessage**

```
public RepeatIntervalCampaignMessage getCampaignMessage (CampaignMessageRecord messageRecord)
```

**RepeatIntervalCampaignMessage**

```
public class RepeatIntervalCampaignMessage extends CampaignMessage
```

**Constructors****RepeatIntervalCampaignMessage**

```
public RepeatIntervalCampaignMessage (CampaignMessageRecord messageRecord)
```

**RepeatIntervalCampaignMessage**

```
public RepeatIntervalCampaignMessage (Time startTime, Period repeatInterval)
```

**RepeatIntervalCampaignMessage**

```
public RepeatIntervalCampaignMessage (String name, List<String> formats, List<String> languages, String messageKey, Time startTime, Period repeatInterval)
```

**Methods****getRepeatInterval**

```
public Period getRepeatInterval ()
```

**getRepeatIntervalInMillis**

```
public long getRepeatIntervalInMillis ()
```

**getRepeatIntervalInSeconds**

```
public Integer getRepeatIntervalInSeconds ()
```

**setRepeatInterval**

```
public final void setRepeatInterval (Period repeatInterval)
```

**setRepeatInterval**

```
public final void setRepeatInterval (String repeatInterval)
```

**validate**

```
public void validate ()
```

## 19.14.67 org.motechproject.messagecampaign.exception

### CampaignAlreadyEndedException

public class **CampaignAlreadyEndedException** extends [MessageCampaignException](#)  
Thrown when trying to enroll a person to campaign that has already ended.

#### Constructors

##### CampaignAlreadyEndedException

public **CampaignAlreadyEndedException** (*String campaignName*, *Throwable e*)

### CampaignEnrollmentException

public class **CampaignEnrollmentException** extends [RuntimeException](#)  
Signals than an exception occurred during campaign enrollment process

#### Constructors

##### CampaignEnrollmentException

public **CampaignEnrollmentException** (*String message*)

### CampaignJsonException

public class **CampaignJsonException** extends [MessageCampaignException](#)  
Thrown when JSON sent to the server is not a valid campaign.

#### Constructors

##### CampaignJsonException

public **CampaignJsonException** (*Throwable e*)

### CampaignMessageValidationException

public class **CampaignMessageValidationException** extends [CampaignValidationException](#)

#### Constructors

##### CampaignMessageValidationException

public **CampaignMessageValidationException** (*String message*)

##### CampaignMessageValidationException

public **CampaignMessageValidationException** (*String message*, *Exception ex*)

### CampaignNotFoundException

public class **CampaignNotFoundException** extends [RuntimeException](#)

## Constructors

### CampaignNotFoundException

public **CampaignNotFoundException** (*String message*)

### CampaignValidationException

public class **CampaignValidationException** extends *RuntimeException*

## Constructors

### CampaignValidationException

public **CampaignValidationException** (*String message*)

### CampaignValidationException

public **CampaignValidationException** (*String message*, *Exception ex*)

### EnrollmentAlreadyExists

public class **EnrollmentAlreadyExists** extends *MessageCampaignException*

Thrown when trying to enroll a person twice to the same campaign.

## Constructors

### EnrollmentAlreadyExists

public **EnrollmentAlreadyExists** (*String externalId*, *String campaignName*)

### EnrollmentAlreadyExists

public **EnrollmentAlreadyExists** (*String externalId*, *String campaignName*, *Throwable e*)

### EnrollmentNotFoundException

public class **EnrollmentNotFoundException** extends *RuntimeException*

## Constructors

### EnrollmentNotFoundException

public **EnrollmentNotFoundException** ()

### EnrollmentNotFoundException

public **EnrollmentNotFoundException** (*String message*)

### EnrollmentNotFoundException

public **EnrollmentNotFoundException** (*String message*, *Throwable cause*)



## MessageCampaignException

public abstract class **MessageCampaignException** extends [RuntimeException](#)

Base class used by exception indicating that there were some problems with person enrollment.

### Constructors

#### MessageCampaignException

protected **MessageCampaignException** ([String](#) message, [String](#) messageKey, [Throwable](#) cause)

#### MessageCampaignException

protected **MessageCampaignException** ([String](#) message, [String](#) messageKey, [List](#)<[String](#)> params, [Throwable](#) cause)

### Methods

#### getMessageKey

public [MessageKey](#) **getMessageKey** ()

## MessageCampaignException.MessageKey

public static class **MessageKey**

Represents a single parametrized message.

### Constructors

#### MessageKey

public **MessageKey** ([String](#) key, [List](#)<[String](#)> params)

### Methods

#### getKey

public [String](#) **getKey** ()

#### getParams

public [List](#)<[String](#)> **getParams** ()

## SchedulingException

public class **SchedulingException** extends [MessageCampaignException](#)

Thrown when there was a problem while scheduling job for enrollee.

### Constructors

#### SchedulingException

public **SchedulingException** ([String](#) externalId, [Throwable](#) e)

## UnknownCampaignTypeException

public class **UnknownCampaignTypeException** extends [RuntimeException](#)

### Constructors

#### UnknownCampaignTypeException

public **UnknownCampaignTypeException** ([String](#) *message*)

## 19.14.68 org.motechproject.messagecampaign.loader

### CampaignJsonLoader

public class **CampaignJsonLoader**

### Constructors

#### CampaignJsonLoader

public **CampaignJsonLoader** ()

#### CampaignJsonLoader

public **CampaignJsonLoader** ([MotechJsonReader](#) *motechJsonReader*)

### Methods

#### getMessageCampaignsJsonFile

public [String](#) **getMessageCampaignsJsonFile** ()

#### loadAfterInit

public void **loadAfterInit** ()

#### loadCampaigns

public [List](#)<[CampaignRecord](#)> **loadCampaigns** ([String](#) *filename*)

#### loadCampaigns

public [List](#)<[CampaignRecord](#)> **loadCampaigns** ([InputStream](#) *in*)

#### loadSingleCampaign

public [CampaignRecord](#) **loadSingleCampaign** ([InputStream](#) *in*)

#### loadSingleCampaign

public [CampaignRecord](#) **loadSingleCampaign** ([String](#) *filename*)

#### setAllMessageCampaigns

public void **setAllMessageCampaigns** ([CampaignRecordService](#) *campaignRecordService*)

**setMessageCampaignsJsonFile**

public void **setMessageCampaignsJsonFile** (*String messageCampaignsJsonFile*)

**setSettings**

public void **setSettings** (SettingsFacade *settings*)

## 19.14.69 org.motechproject.messagecampaign.service

### CampaignEnrollmentRecord

public class **CampaignEnrollmentRecord**

ingroup MessageCampaign

This is the record which will be returned when message campaign service is queried for enrollments It holds the details of an enrollment

#### Constructors

**CampaignEnrollmentRecord**

public **CampaignEnrollmentRecord** (*String externalId*, *String campaignName*, *LocalDate referenceDate*, *CampaignEnrollmentStatus status*)

This is the constructor which is used to create an CampaignEnrollmentRecord

**Parameters**

- **externalId** –
- **campaignName** –
- **referenceDate** –
- **status** –

#### Methods

**getCampaignName**

public *String* **getCampaignName** ()

This returns the Campaign Name of a CampaignEnrollmentRecord

**Returns** *String*

**getExternalId**

public *String* **getExternalId** ()

This returns the External Id of a CampaignEnrollmentRecord

**Returns** *String*

**getReferenceDate**

public *LocalDate* **getReferenceDate** ()

This returns the Start Date of a CampaignEnrollmentRecord

**Returns** *LocalDate*

**getStatus**

```
public CampaignEnrollmentStatus getStatus ()
```

This returns the Status of a CampaignEnrollmentRecord

**Returns** CampaignEnrollmentStatus

**CampaignEnrollmentRecordMapper**

```
public class CampaignEnrollmentRecordMapper
```

**Methods****map**

```
public CampaignEnrollmentRecord map (CampaignEnrollment enrollment)
```

**CampaignEnrollmentsQuery**

```
public class CampaignEnrollmentsQuery
```

```
    ingroup MessageCampaign
```

This is the Query builder for retrieving campaign enrollments Provides methods for different query criteria The order of criteria matters, as the first criterion is used to fetch the result from database and the other criterion are used to filter the results fetched by the first criterion(in memory)

**Methods****getCriteria**

```
public List<Criterion> getCriteria ()
```

This gives all the criterion which are present in the built query

**Returns** List

**getPrimaryCriterion**

```
public Criterion getPrimaryCriterion ()
```

This gives the primary criterion in the built query, which is used to fetch the results from database

**Returns** Criterion

**getSecondaryCriteria**

```
public List<Criterion> getSecondaryCriteria ()
```

This gives all the criterion other than primary criterion in the built query, which are used to filter the results of the primary criterion

**Returns** List

**havingState**

```
public CampaignEnrollmentsQuery havingState (CampaignEnrollmentStatus campaignEnrollmentStatus)
```

This provides the method for the Status Criterion using which campaign enrollments are filtered based on their status

**Parameters**

- `campaignEnrollmentStatus` –

#### **withCampaignName**

public `CampaignEnrollmentsQuery` **withCampaignName** (`String` *campaignName*)

This provides the method for the CampaignName Criterion using which campaign enrollments belongs to a particular campaign are filtered

##### **Parameters**

- `campaignName` –

#### **withExternalId**

public `CampaignEnrollmentsQuery` **withExternalId** (`String` *externalId*)

This provides the method for the ExternalId Criterion using which campaign enrollments for an ExternalId are filtered

##### **Parameters**

- `externalId` –

### **EnrollmentService**

public class **EnrollmentService**

#### **Methods**

##### **delete**

public void **delete** (`CampaignEnrollment` *enrollment*)

##### **register**

public void **register** (`CampaignEnrollment` *enrollment*)

##### **search**

public `List`<`CampaignEnrollment`> **search** (`CampaignEnrollmentsQuery` *query*)

##### **unregister**

public void **unregister** (`String` *externalId*, `String` *campaignName*)

##### **unregister**

public void **unregister** (`CampaignEnrollment` *enrollment*)

### **MessageCampaignService**

public interface **MessageCampaignService**

ingroup `MessageCampaign`

## Fields

### MESSAGE\_CAMPAIGNS\_JSON\_FILENAME

`String MESSAGE_CAMPAIGNS_JSON_FILENAME`

## Methods

### campaignCompleted

`void campaignCompleted (String externalId, String campaignName)`

### deleteCampaign

`void deleteCampaign (String campaignName)`

### enroll

`void enroll (CampaignRequest enrollRequest)`

Enrolls the external id into the campaign as specified in the request. The enrolled entity will have events raised against it according to the campaign definition.

#### Parameters

- `enrollRequest` –

### getAllCampaignRecords

`List<CampaignRecord> getAllCampaignRecords ()`

### getCampaignRecord

`CampaignRecord getCampaignRecord (String campaignName)`

### getCampaignTimings

`Map<String, List<DateTime>> getCampaignTimings (String externalId, String campaignName, DateTime startDate, DateTime endDate)`

### getLatestCampaignMessage

`String getLatestCampaignMessage (String campaignName, String externalId)`

Return last message we attempted to send for the given campaign enrollment

### getNextCampaignMessage

`String getNextCampaignMessage (String campaignName, String externalId)`

Return next message that will be sent for given enrollment.

### loadCampaigns

`void loadCampaigns ()`

### rescheduleMessageJob

`void rescheduleMessageJob (Long campaignMessageRecordId)`

**saveCampaign**

void **saveCampaign** ([CampaignRecord](#) *campaign*)

**scheduleJobsForEnrollment**

void **scheduleJobsForEnrollment** ([CampaignEnrollment](#) *enrollment*)

**search**

[List<CampaignEnrollmentRecord>](#) **search** ([CampaignEnrollmentsQuery](#) *query*)

Searches and returns the Campaign Enrollment Records as per the criteria in the given CampaignEnrollments-Query. The query consists of various criteria based on Status, ExternalId and CampaignName of the CampaignEnrollment.

**Parameters**

- **query** –

**Returns** List

**stopAll**

void **stopAll** ([CampaignEnrollmentsQuery](#) *query*)

Unenrolls all campaigns enrollment which match criteria provided by a CampaignEnrollmentsQuery object.

**Parameters**

- **query** –

**stopAll**

void **stopAll** ([CampaignEnrollmentsQuery](#) *query*, boolean *deleteEnrollments*)

**unenroll**

void **unenroll** ([String](#) *externalId*, [String](#) *campaignName*)

Unenrolls an external from the campaign as specified in the request. The entity will no longer receive events from the campaign.

**Parameters**

- **externalId** –
  - a client defined id to identify the enrollment
- **campaignName** –
  - the campaign into which the entity should be enrolled

**unscheduleJobsForEnrollment**

void **unscheduleJobsForEnrollment** ([CampaignEnrollment](#) *enrollment*)

**unscheduleMessageJob**

void **unscheduleMessageJob** ([CampaignMessageRecord](#) *campaignMessageRecord*)

**updateEnrollment**

void **updateEnrollment** ([CampaignRequest](#) enrollRequest, [Long](#) enrollmentId)

Updates existing campaign enrollment with data specified in the request.

**Parameters**

- **enrollRequest** –
- **enrollmentId** –

**updateEnrollments**

void **updateEnrollments** ([Long](#) campaignId)

**MessageCampaignServiceImpl**

public class **MessageCampaignServiceImpl** implements [MessageCampaignService](#)

Implementation of [MessageCampaignService](#)

**Constructors****MessageCampaignServiceImpl**

public **MessageCampaignServiceImpl** ([EnrollmentService](#) enrollmentService, [CampaignEnrollmentDataService](#) campaignEnrollmentDataService, [CampaignEnrollmentRecordMapper](#) campaignEnrollmentRecordMapper, [CampaignSchedulerFactory](#) campaignSchedulerFactory, [CampaignRecordService](#) campaignRecordService, [CampaignMessageRecordService](#) campaignMessageRecordService, [EventRelay](#) relay, [MotechSchedulerService](#) schedulerService)

**Methods****campaignCompleted**

public void **campaignCompleted** ([String](#) externalId, [String](#) campaignName)

**changeMaxUploadSize**

public void **changeMaxUploadSize** ([MotechEvent](#) event)

**deleteCampaign**

public void **deleteCampaign** ([String](#) campaignName)

**enroll**

public void **enroll** ([CampaignRequest](#) request)

**getAllCampaignRecords**

public [List](#)<[CampaignRecord](#)> **getAllCampaignRecords** ()

**getCampaignRecord**

public [CampaignRecord](#) **getCampaignRecord** ([String](#) campaignName)



**getCampaignTimings**

```
public Map<String, List<DateTime>> getCampaignTimings (String externalId, String campaignName,  
                                                       DateTime startDate, DateTime endDate)
```

**getLatestCampaignMessage**

```
public String getLatestCampaignMessage (String campaignName, String externalId)
```

**getNextCampaignMessage**

```
public String getNextCampaignMessage (String campaignName, String externalId)
```

**loadCampaigns**

```
public void loadCampaigns ()
```

**loadCampaignsJson**

```
public void loadCampaignsJson ()
```

**rescheduleMessageJob**

```
public void rescheduleMessageJob (Long campaignMessageRecordId)
```

**saveCampaign**

```
public void saveCampaign (CampaignRecord campaign)
```

**scheduleJobsForEnrollment**

```
public void scheduleJobsForEnrollment (CampaignEnrollment enrollment)
```

**search**

```
public List<CampaignEnrollmentRecord> search (CampaignEnrollmentsQuery query)
```

**stopAll**

```
public void stopAll (CampaignEnrollmentsQuery query)
```

**stopAll**

```
public void stopAll (CampaignEnrollmentsQuery query, boolean deleteEnrollments)
```

**unenroll**

```
public void unenroll (String externalId, String campaignName)
```

**unscheduleJobsForEnrollment**

```
public void unscheduleJobsForEnrollment (CampaignEnrollment enrollment)
```

**unscheduleMessageJob**

```
public void unscheduleMessageJob (CampaignMessageRecord campaignMessageRecord)
```

**updateEnrollment**

```
public void updateEnrollment (CampaignRequest enrollRequest, Long enrollmentId)
```

**updateEnrollments**

public void **updateEnrollments** (*Long campaignId*)

**19.14.70 org.motechproject.mtraining.domain****ActivityRecord**

public class **ActivityRecord** extends MdsEntity

Log for an instance of course activity for a user identified by externalId This could be used either as a book-marking system or enrollment system to track progress

**Constructors****ActivityRecord**

public **ActivityRecord** ()

**ActivityRecord**

public **ActivityRecord** (*String externalId*, *String courseName*, *String chapterName*, *String lessonName*,  
*DateTime startTime*, *DateTime completionTime*, *ActivityState state*)

**ActivityRecord**

public **ActivityRecord** (*String externalId*, *String courseName*, *String chapterName*, *String lessonName*,  
*String quizName*, *double quizScore*, *DateTime startTime*, *DateTime completionTime*, *ActivityState state*)

**Methods****getChapterName**

public *String* **getChapterName** ()

**getCompletionTime**

public *DateTime* **getCompletionTime** ()

**getCourseName**

public *String* **getCourseName** ()

**getExternalId**

public *String* **getExternalId** ()

**getLessonName**

public *String* **getLessonName** ()

**getQuizName**

public *String* **getQuizName** ()

**getQuizScore**

public double **getQuizScore** ()

**getStartTime**

public [DateTime](#) **getStartTime** ()

**getState**

public [ActivityState](#) **getState** ()

**setChapterName**

public void **setChapterName** ([String](#) *chapterName*)

**setCompletionTime**

public void **setCompletionTime** ([DateTime](#) *completionTime*)

**setCourseName**

public void **setCourseName** ([String](#) *courseName*)

**setExternalId**

public void **setExternalId** ([String](#) *externalId*)

**setLessonName**

public void **setLessonName** ([String](#) *lessonName*)

**setQuizName**

public void **setQuizName** ([String](#) *quizName*)

**setQuizScore**

public void **setQuizScore** (double *quizScore*)

**setStartTime**

public void **setStartTime** ([DateTime](#) *startTime*)

**setState**

public void **setState** ([ActivityState](#) *state*)

**ActivityState**

public enum **ActivityState**

Enum state for activity. This can be use to mark the user as having started a course, in progress or completed a course curriculum

**Enum Constants****COMPLETED**

public static final [ActivityState](#) **COMPLETED**

**INPROGRESS**

public static final *ActivityState* **INPROGRESS**

**NONE**

public static final *ActivityState* **NONE**

**STARTED**

public static final *ActivityState* **STARTED**

**Bookmark**

public class **Bookmark** extends *MdsEntity*

Bookmark object to store the progress for the user. This stores the identifier for the individual course units (like chapters, lessons)

**Constructors****Bookmark**

public **Bookmark** ()

**Bookmark**

public **Bookmark** (*String externalId*, *String courseIdIdentifier*, *String chapterIdentifier*, *String lessonIdentifier*,  
                  *Map<String, Object> progress*)

**Methods****getChapterIdentifier**

public *String* **getChapterIdentifier** ()

**getCourseIdentifier**

public *String* **getCourseIdentifier** ()

**getExternalId**

public *String* **getExternalId** ()

**getLessonIdentifier**

public *String* **getLessonIdentifier** ()

**getProgress**

public *Map<String, Object>* **getProgress** ()

**setChapterIdentifier**

public void **setChapterIdentifier** (*String chapterIdentifier*)

**setCourseIdentifier**

public void **setCourseIdentifier** (*String courseIdIdentifier*)

**setExternalId**

public void **setExternalId** ([String](#) *externalId*)

**setLessonIdentifier**

public void **setLessonIdentifier** ([String](#) *lessonIdentifier*)

**setProgress**

public void **setProgress** ([Map](#)<[String](#), [Object](#)> *progress*)

## Chapter

public class **Chapter** extends [CourseUnitMetadata](#)

Chapter object to store quiz and lesson metadata. A chapter contains a list of possible lessons

### Constructors

**Chapter**

public **Chapter** ()

**Chapter**

public **Chapter** ([String](#) *name*, [CourseUnitState](#) *state*, [String](#) *content*)

**Chapter**

public **Chapter** ([String](#) *name*, [CourseUnitState](#) *state*, [String](#) *content*, [List](#)<[Lesson](#)> *lessons*)

**Chapter**

public **Chapter** ([String](#) *name*, [CourseUnitState](#) *state*, [String](#) *content*, [List](#)<[Lesson](#)> *lessons*, [Quiz](#) *quiz*)

### Methods

**getLessons**

public [List](#)<[Lesson](#)> **getLessons** ()

**getQuiz**

public [Quiz](#) **getQuiz** ()

**setLessons**

public void **setLessons** ([List](#)<[Lesson](#)> *lessons*)

**setQuiz**

public void **setQuiz** ([Quiz](#) *quiz*)

## Course

public class **Course** extends [CourseUnitMetadata](#)

Course metadata containing chapter information Structure - Course is a collection of chapters Chapter is a collection of lessons Every Chapter has a quiz associated with it Quiz is a collection of questions Question contains pointer to question resource and answer resource

## Constructors

### Course

public **Course** ([String name](#), [CourseUnitState state](#), [String content](#))

### Course

public **Course** ([String name](#), [CourseUnitState state](#), [String content](#), [List<Chapter> chapters](#))

## Methods

### getChapters

public [List<Chapter>](#) **getChapters** ()

### setChapters

public void **setChapters** ([List<Chapter> chapters](#))

## CourseUnitMetadata

public class **CourseUnitMetadata** extends [MdsEntity](#)

Common metadata shared by all course sub classes

## Constructors

### CourseUnitMetadata

public **CourseUnitMetadata** ([String name](#), [CourseUnitState state](#), [String content](#))

Constructor with all arguments

#### Parameters

- **name** – Name of the unit
- **state** – Status of the unit
- **content** – Content reference for the unit

## Methods

### getContent

public [String](#) **getContent** ()

### getName

public [String](#) **getName** ()

**getState**

```
public CourseUnitState getState ()
```

**setContent**

```
public void setContent (String content)
```

**setName**

```
public void setName (String name)
```

**setState**

```
public void setState (CourseUnitState state)
```

**CourseUnitState**

```
public enum CourseUnitState
```

Course state enum This is used to mark the course unit (eg - Course, Chapter, Lesson, etc) as being Active, Pending or Inactive. This helps an implementation filter the list of courses/chapter/lesson according to their business logic

**Enum Constants****Active**

```
public static final CourseUnitState Active
```

**Inactive**

```
public static final CourseUnitState Inactive
```

**Pending**

```
public static final CourseUnitState Pending
```

**Lesson**

```
public class Lesson extends CourseUnitMetadata
```

Lesson domain object forms the leaf node in the course structure hierarchy. A lesson is typically the leaf node in the Course structure hierarchy

**Constructors****Lesson**

```
public Lesson (String name, CourseUnitState state, String content)
```

**Question**

```
public class Question
```

Question object with resource identifiers for question and answer

## Constructors

### Question

public **Question** ()

### Question

public **Question** (*String question*, *String answer*)

## Methods

### getAnswer

public *String* **getAnswer** ()

### getQuestion

public *String* **getQuestion** ()

### setAnswer

public void **setAnswer** (*String answer*)

### setQuestion

public void **setQuestion** (*String question*)

## Quiz

public class **Quiz** extends *CourseUnitMetadata*

Quiz object to store questions and answer for a chapter

## Constructors

### Quiz

public **Quiz** (*String name*, *CourseUnitState state*, *String content*)

### Quiz

public **Quiz** (*String name*, *CourseUnitState state*, *String content*, *List<Question> questions*, double *passPercentage*)

## Methods

### getPassPercentage

public double **getPassPercentage** ()

### getQuestions

public *List<Question>* **getQuestions** ()

### setPassPercentage

public void **setPassPercentage** (double *passPercentage*)



**setQuestions**

public void **setQuestions** ([List<Question>](#) *questions*)

## 19.14.71 org.motechproject.mtraining.repository

### ActivityDataService

public interface **ActivityDataService** extends [MotechDataService<ActivityRecord>](#)

Interface for repository that persists simple records and allows CRUD. [MotechDataService](#) base class will provide the implementation of this class as well as methods for adding, deleting, saving and finding all instances. In this class we define and custom lookups we may need.

#### Methods

**findRecordById**

[ActivityRecord](#) **findRecordById** ([Long](#) *id*)

**findRecordsForUser**

[List<ActivityRecord>](#) **findRecordsForUser** ([String](#) *externalId*)

**findRecordsForUserByState**

[List<ActivityRecord>](#) **findRecordsForUserByState** ([String](#) *externalId*, [ActivityState](#) *state*)

### BookmarkDataService

public interface **BookmarkDataService** extends [MotechDataService<Bookmark>](#)

Interface for repository that persists simple records and allows CRUD. [MotechDataService](#) base class will provide the implementation of this class as well as methods for adding, deleting, saving and finding all instances. In this class we define and custom lookups we may need.

#### Methods

**findBookmarkById**

[Bookmark](#) **findBookmarkById** ([Long](#) *id*)

**findBookmarksForUser**

[List<Bookmark>](#) **findBookmarksForUser** ([String](#) *externalId*)

**findBookmarksForUserParams**

[List<Bookmark>](#) **findBookmarksForUserParams** ([String](#) *externalId*, [QueryParams](#) *queryParams*)

### ChapterDataService

public interface **ChapterDataService** extends [MotechDataService<Chapter>](#)

Interface for repository that persists simple records and allows CRUD. [MotechDataService](#) base class will provide the implementation of this class as well as methods for adding, deleting, saving and finding all instances. In this class we define and custom lookups we may need.

## Methods

### findChapterById

`Chapter` **findChapterById** (`Long id`)

### findChapterByName

`List<Chapter>` **findChapterByName** (`String chapterName`)

## CourseDataService

public interface **CourseDataService** extends `MotechDataService<Course>`

Interface for repository that persists simple records and allows CRUD. `MotechDataService` base class will provide the implementation of this class as well as methods for adding, deleting, saving and finding all instances. In this class we define and custom lookups we may need.

## Methods

### findCourseById

`Course` **findCourseById** (`Long id`)

### findCourseByName

`List<Course>` **findCourseByName** (`String courseName`)

## LessonDataService

public interface **LessonDataService** extends `MotechDataService<Lesson>`

Interface for repository that persists simple records and allows CRUD. `MotechDataService` base class will provide the implementation of this class as well as methods for adding, deleting, saving and finding all instances. In this class we define and custom lookups we may need.

## Methods

### findLessonById

`Lesson` **findLessonById** (`Long id`)

### findLessonByName

`List<Lesson>` **findLessonByName** (`String lessonName`)

## QuizDataService

public interface **QuizDataService** extends `MotechDataService<Quiz>`

Interface for repository that persists simple records and allows CRUD. `MotechDataService` base class will provide the implementation of this class as well as methods for adding, deleting, saving and finding all instances. In this class we define and custom lookups we may need.

## Methods

### findQuizById

Quiz **findQuizById** (Long *id*)

### findQuizByName

List<Quiz> **findQuizByName** (String *quizName*)

## 19.14.72 org.motechproject.mtraining.service

### ActivityService

public interface **ActivityService**

Service interface for managing activity records. These are used for managing activities for single user/patient and serve as enrollment records and quiz score records. These can be used to report activity for a user or course as well as track progress in the course curriculum

## Methods

### createActivity

ActivityRecord **createActivity** (ActivityRecord *activityRecord*)

Add activity for a user (identified by externalId field in the record)

#### Parameters

- **activityRecord** – activity to record

**Returns** activity record from the operation

### deleteActivity

void **deleteActivity** (long *activityRecordId*)

Delete the activity by record id

#### Parameters

- **activityRecordId** – activity record id to delete

### deleteAllActivityForUser

void **deleteAllActivityForUser** (String *externalId*)

Delete all activity for a user (intended for data cleanups)

#### Parameters

- **externalId** – Id of the user

### getActivityById

ActivityRecord **getActivityById** (long *activityId*)

Get an activity by Id

#### Parameters

- **activityId** – Id of the user

**Returns** activity record with id

**getAllActivityForUser**

List<ActivityRecord> **getAllActivityForUser** (String *externalId*)

Get a list of all activity for a user

**Parameters**

- **externalId** – Id of the user

**Returns** list of activity records

**getCompletedActivityForUser**

List<ActivityRecord> **getCompletedActivityForUser** (String *externalId*)

Get all completed activity for user identified by external id (decided by whether completion time has been set on the record)

**Parameters**

- **externalId** – external id of the user

**Returns** list of activity records

**updateActivity**

ActivityRecord **updateActivity** (ActivityRecord *activityRecord*)

Update the activity for a user (set completion time, etc)

**Parameters**

- **activityRecord** – activity record to update

**Returns** updated activity record

**BookmarkService**

public interface **BookmarkService**

Service interface for management of course bookmarks for a user. This is used to maintain the progress of a user in the curriculum

**Methods****createBookmark**

Bookmark **createBookmark** (Bookmark *bookmark*)

Create a bookmark for a user

**Parameters**

- **bookmark** – bookmark object to save

**Returns** bookmark object created in the store

**deleteAllBookmarksForUser**

void **deleteAllBookmarksForUser** (String *externalId*)

delete all bookmarks for a given user

**Parameters**

- **externalId** – external tracking id for the user

**deleteBookmark**

void **deleteBookmark** (long *bookmarkId*)

Delete a bookmark with the given id

**Parameters**

- **bookmarkId** – id of the bookmark

**getAllBookmarksForUser**

List<Bookmark> **getAllBookmarksForUser** (String *externalId*)

Get all the bookmarks for a user

**Parameters**

- **externalId** – external tracking id for the user

**Returns** list of bookmarks for user

**getBookmarkById**

Bookmark **getBookmarkById** (long *bookmarkId*)

get bookmark by bookmark id

**Parameters**

- **bookmarkId** – id of the bookmark

**Returns** bookmark object with the id

**getLatestBookmarkByUserId**

Bookmark **getLatestBookmarkByUserId** (String *externalId*)

Get the latest bookmark for the user identified by externalId

**Parameters**

- **externalId** – external tracking id for the user

**Returns** bookmark object for the user or null if no bookmarks exist for user

**updateBookmark**

Bookmark **updateBookmark** (Bookmark *bookmark*)

Update the given bookmark for the user

**Parameters**

- **bookmark** – bookmark to update

**Returns** updated bookmark object

**MTrainingService**

public interface **MTrainingService**

Service interface for mTraining. Contains APIs to perform CRUD operations on Course and related objects like chapters, lessons, quiz, etc

## Methods

### createChapter

[Chapter](#) **createChapter** ([Chapter](#) *chapter*)

Create a chapter

#### Parameters

- **chapter** – chapter to create

**Returns** chapter created in the store

### createCourse

[Course](#) **createCourse** ([Course](#) *course*)

Create a course with the given structure

#### Parameters

- **course** – course object to store

**Returns** Course object created in the store

### createLesson

[Lesson](#) **createLesson** ([Lesson](#) *lesson*)

Create a lesson

#### Parameters

- **lesson** – lesson to create in store

**Returns** stored lesson from store

### createQuiz

[Quiz](#) **createQuiz** ([Quiz](#) *quiz*)

Create a quiz

#### Parameters

- **quiz** – New quiz object to store

**Returns** quiz object from store

### deleteChapter

void **deleteChapter** (long *chapterId*)

Delete a chapter by id

#### Parameters

- **chapterId** – id of the chapter to delete

### deleteCourse

void **deleteCourse** (long *courseId*)

Delete the course with the given id

#### Parameters

- **courseId** – ID of the course to delete

**deleteLesson**

void **deleteLesson** (long *lessonId*)  
delete a lesson with a given id

**Parameters**

- **lessonId** – id of the lesson to delete

**deleteQuiz**

void **deleteQuiz** (long *quizId*)  
delete a quiz with id

**Parameters**

- **quizId** – id of the quiz to delete

**getAllChapters**

List<Chapter> **getAllChapters** ()  
Get all chapters

**Returns** list of chapters

**getAllCourses**

List<Course> **getAllCourses** ()  
Get all courses

**Returns** list of courses

**getAllLessons**

List<Lesson> **getAllLessons** ()  
Get all lessons

**Returns** list of lessons

**getAllQuizzes**

List<Quiz> **getAllQuizzes** ()  
Get all quizzes

**Returns** list of quizzes

**getChapterById**

Chapter **getChapterById** (long *chapterId*)  
Get chapter from store by ID

**Parameters**

- **chapterId** – the chapter ID

**Returns** Chapter with id

### getChapterByName

List<Chapter> **getChapterByName** (String *chapterName*)

Get chapter by name

#### Parameters

- **chapterName** – name of the chapter

**Returns** list of chapters with matching name

### getCourseById

Course **getCourseById** (long *courseId*)

Retrieve a course with the given course id

#### Parameters

- **courseId** – id of the course to retrieve

**Returns** the course with the given id, or null if it does not exist

### getCourseByName

List<Course> **getCourseByName** (String *courseName*)

Get courses that match the name

#### Parameters

- **courseName** – name of the course

**Returns** list of courses that match the course name

### getLessonById

Lesson **getLessonById** (long *id*)

get lesson by id

#### Parameters

- **id** – id of the lesson

**Returns** lesson with id

### getLessonByName

List<Lesson> **getLessonByName** (String *lessonName*)

get lesson by name

#### Parameters

- **lessonName** – name of the lesson

**Returns** list of lessons with matching name

### getQuizById

Quiz **getQuizById** (long *id*)

Get a quiz by id

#### Parameters

- **id** – Id of the quiz

**Returns** quiz object with id



**getQuizByName**

List<Quiz> **getQuizByName** (String *quizName*)

Get a quiz by a name

**Parameters**

- **quizName** – name of the quiz

**Returns** list of quiz objects with the given name

**getQuizForChapter**

Quiz **getQuizForChapter** (long *chapterId*)

get the quiz for a given chapter

**Parameters**

- **chapterId** – chapter id to retrieve quiz for

**Returns** Quiz object for the chapter

**updateChapter**

Chapter **updateChapter** (Chapter *chapter*)

Update a given chapter

**Parameters**

- **chapter** – chapter to update

**Returns** updated chapter from the store

**updateCourse**

Course **updateCourse** (Course *course*)

Update a course with the given structure

**Parameters**

- **course** – Course structure to update

**Returns** the newly updated course

**updateLesson**

Lesson **updateLesson** (Lesson *lesson*)

update a lesson in the store

**Parameters**

- **lesson** – lesson to update

**Returns** updated lesson from store

**updateQuiz**

Quiz **updateQuiz** (Quiz *quiz*)

update a quiz object in store

**Parameters**

- **quiz** – quiz object to update

**Returns** updated quiz object from store

## 19.14.73 org.motechproject.openmrs19.domain

### OpenMRSAttribute

public class **OpenMRSAttribute**

Represents a single attribute of a user. It's a part of the MOTECH model.

#### Constructors

##### OpenMRSAttribute

public **OpenMRSAttribute** (*String name*, *String value*)

Creates an attribute with the given *name* and *value*.

##### Parameters

- **name** – the name of the attribute
- **value** – the value the attribute

#### Methods

##### equals

public boolean **equals** (*Object o*)

##### getName

public *String* **getName** ()

##### getValue

public *String* **getValue** ()

##### hashCode

public int **hashCode** ()

##### name

public *String* **name** ()

##### setName

public void **setName** (*String name*)

##### setValue

public void **setValue** (*String value*)

##### value

public *String* **value** ()

## OpenMRSConcept

public class **OpenMRSConcept**

Represents a single concept. Concepts are individual data points collected from a population of patients. It's a part of the MOTech model.

### Constructors

#### OpenMRSConcept

public **OpenMRSConcept** ()

Default constructor.

#### OpenMRSConcept

public **OpenMRSConcept** ([OpenMRSConceptName](#) *name*)

Creates a concept with the given name.

#### Parameters

- **name** – the name of the concept

### Methods

#### equals

public boolean **equals** ([Object](#) *o*)

#### getConceptClass

public [String](#) **getConceptClass** ()

#### getDataType

public [String](#) **getDataType** ()

#### getDisplay

public [String](#) **getDisplay** ()

#### getId

public [String](#) **getId** ()

#### getName

public [OpenMRSConceptName](#) **getName** ()

#### getNames

public [List](#)<[OpenMRSConceptName](#)> **getNames** ()

#### getUuid

public [String](#) **getUuid** ()

**hashCode**

```
public int hashCode ()
```

**setConceptClass**

```
public void setConceptClass (String conceptClass)
```

**setDataType**

```
public void setDataType (String dataType)
```

**setDisplay**

```
public void setDisplay (String display)
```

**setId**

```
public void setId (String id)
```

**setName**

```
public void setName (OpenMRSCONCEPT_NAME name)
```

**setNames**

```
public void setNames (List<OpenMRSCONCEPT_NAME> names)
```

**setUuid**

```
public void setUuid (String uuid)
```

**OpenMRSCONCEPT\_NAME**

```
public class OpenMRSCONCEPT_NAME
```

Represents a single concept name. Also stores information about its locale and type.

**Fields****DEFAULT\_CONCEPT\_NAME\_TYPE**

```
public static final String DEFAULT_CONCEPT_NAME_TYPE
```

**DEFAULT\_LOCALE**

```
public static final String DEFAULT_LOCALE
```

**Constructors****OpenMRSCONCEPT\_NAME**

```
public OpenMRSCONCEPT_NAME (String name)
```

Creates a concept name from the given name. Uses default locale and concept type.

**Parameters**

- **name** – the name of the concept

**OpenMRSConceptName**

public **OpenMRSConceptName** (*String name, String locale, String conceptNameType*)

Creates a concept name from the given name. Uses given locale and conceptNameType.

**Parameters**

- **name** – the name of the concept
- **locale** – the locale of the concept name
- **conceptNameType** – the type of the concept name

**OpenMRSConceptName**

public **OpenMRSConceptName** (*Concept.ConceptName conceptName*)

Copying constructor.

**Parameters**

- **conceptName** – the concept name to copy

**Methods****equals**

public boolean **equals** (*Object o*)

**getConceptNameType**

public *String* **getConceptNameType** ()

**getLocale**

public *String* **getLocale** ()

**getName**

public *String* **getName** ()

**hashCode**

public int **hashCode** ()

**setConceptNameType**

public void **setConceptNameType** (*String conceptNameType*)

**setLocale**

public void **setLocale** (*String locale*)

**setName**

public void **setName** (*String name*)

## OpenMRSEncounter

public class **OpenMRSEncounter**

Represents a single OpenMRS encounter. An encounter is a single, specific interaction between the patient and a provider. It can be any interaction and includes doctor visits, laboratory tests, food distribution, home visits, counselor appointments, etc. It's a part of the MOTeCH model.

### Constructors

#### OpenMRSEncounter

public **OpenMRSEncounter** ()

Default constructor.

#### OpenMRSEncounter

public **OpenMRSEncounter** (*OpenMRSPROvider provider*, *OpenMRSUser creator*, *OpenMRSFacility facility*, *Date date*, *OpenMRSPatient patient*, *Set<? extends OpenMRSObservation> observations*, *String encounterType*)

Creates an encounter of type *encounterType* that took place in the *facility* on the *date*. The encounter was held by the *provider* and its subject was the given *patient*. During the encounter the information about the given *observations* were recorder and all the details were entered by the user *creator*.

#### Parameters

- **provider** – the staff who provides information
- **creator** – the staff who enters the details into the OpenMRS system
- **facility** – the location of the encounter
- **date** – the date of the encounter
- **patient** – the patient involved in the encounter
- **observations** – the observations collected during the encounter
- **encounterType** – the type of the encounter.

### Methods

#### getCreator

public *OpenMRSUser* **getCreator** ()

#### getDate

public *DateTime* **getDate** ()

#### getEncounterId

public *String* **getEncounterId** ()

#### getEncounterType

public *String* **getEncounterType** ()

#### getFacility

public *OpenMRSFacility* **getFacility** ()

**getId**

```
public String getId ()
```

**getObservations**

```
public Set<? extends OpenMRSObservation> getObservations ()
```

**getPatient**

```
public OpenMRSPatient getPatient ()
```

**getProvider**

```
public OpenMRSPProvider getProvider ()
```

**setCreator**

```
public void setCreator (OpenMRSUser creator)
```

**setDate**

```
public void setDate (DateTime date)
```

**setEncounterId**

```
public void setEncounterId (String encounterId)
```

**setEncounterType**

```
public void setEncounterType (String encounterType)
```

**setFacility**

```
public void setFacility (OpenMRSFacility facility)
```

**setObservations**

```
public void setObservations (Set<? extends OpenMRSObservation> observations)
```

**setPatient**

```
public void setPatient (OpenMRSPatient patient)
```

**setProvider**

```
public void setProvider (OpenMRSPProvider provider)
```

**updateWithoutObs**

```
public OpenMRSEncounter updateWithoutObs (OpenMRSEncounter fromEncounter)
```

**OpenMRSEncounter.OpenMRSEncounterBuilder**

```
public static class OpenMRSEncounterBuilder
```

## Methods

### build

public `OpenMRSEncounter` **build** ()

### withCreator

public `OpenMRSEncounterBuilder` **withCreator** (`OpenMRSUser` *creator*)

### withCreatorId

public `OpenMRSEncounterBuilder` **withCreatorId** (`String` *creatorId*)

### withDate

public `OpenMRSEncounterBuilder` **withDate** (`Date` *date*)

### withEncounterType

public `OpenMRSEncounterBuilder` **withEncounterType** (`String` *encounterType*)

### withFacility

public `OpenMRSEncounterBuilder` **withFacility** (`OpenMRSFacility` *facility*)

### withFacilityId

public `OpenMRSEncounterBuilder` **withFacilityId** (`String` *facilityId*)

### withId

public `OpenMRSEncounterBuilder` **withId** (`String` *id*)

### withObservations

public `OpenMRSEncounterBuilder` **withObservations** (`Set`<? extends `OpenMRSObservation`> *observations*)

### withPatient

public `OpenMRSEncounterBuilder` **withPatient** (`OpenMRSPatient` *patient*)

### withPatientId

public `OpenMRSEncounterBuilder` **withPatientId** (`String` *patientId*)

### withProvider

public `OpenMRSEncounterBuilder` **withProvider** (`OpenMRSPProvider` *provider*)

### withProviderId

public `OpenMRSEncounterBuilder` **withProviderId** (`String` *providerId*)

## OpenMRSEncounterType

public class **OpenMRSEncounterType**

Class representing a type of the encounter.



## Methods

### getDescription

public `String` `getDescription()`

### getName

public `String` `getName()`

### getUuid

public `String` `getUuid()`

### setDescription

public void `setDescription(String description)`

### setName

public void `setName(String name)`

### setUuid

public void `setUuid(String uuid)`

## OpenMRSFacility

public class **OpenMRSFacility**

Represents a single facility. A facility is a physical place where patients can be seen. It's a part of the MOTECH model.

## Constructors

### OpenMRSFacility

public **OpenMRSFacility**(`String id`)

Creates a facility with the given id.

#### Parameters

- **id** – the ID of the facility

### OpenMRSFacility

public **OpenMRSFacility**(`String name`, `String country`, `String region`, `String countyDistrict`, `String stateProvince`)

Creates a facility with the given name that is placed in the given country, region, countyDistrict and stateProvince.

#### Parameters

- **name** – the name of the facility
- **country** – the name of the country where the facility is placed
- **region** – the name of the region where the facility is placed
- **countyDistrict** – the name of the county/district where the facility is placed
- **stateProvince** – the name of the state/province where the facility is placed

**OpenMRSFacility**

public **OpenMRSFacility** (*String id, String name, String country, String region, String countyDistrict, String stateProvince*)

Creates a facility with the given name and id that is placed in the given country, region, countyDistrict and stateProvince.

**Parameters**

- **id** – the ID of the facility
- **name** – the name of the facility
- **country** – the name of the country where the facility is placed
- **region** – the name of the region where the facility is placed
- **countyDistrict** – the name of the county/district where the facility is placed
- **stateProvince** – the name of the state/province where the facility is placed

**Methods****equals**

public boolean **equals** (*Object o*)

**getCountry**

public *String* **getCountry** ()

**getCountyDistrict**

public *String* **getCountyDistrict** ()

**getFacilityId**

public *String* **getFacilityId** ()

**getId**

public *String* **getId** ()

**getName**

public *String* **getName** ()

**getRegion**

public *String* **getRegion** ()

**getStateProvince**

public *String* **getStateProvince** ()

**hashCode**

public int **hashCode** ()

**setCountry**

public void **setCountry** (*String country*)

**setCountyDistrict**

public void **setCountyDistrict** (*String countyDistrict*)

**setFacilityId**

public void **setFacilityId** (*String id*)

**setId**

public void **setId** (*String id*)

**setName**

public void **setName** (*String name*)

**setRegion**

public void **setRegion** (*String region*)

**setStateProvince**

public void **setStateProvince** (*String stateProvince*)

**OpenMRSObservation**

public class **OpenMRSObservation**<T>

Represents a single observation. An observation is a single piece of information that is recorded about a patient at a moment in time.

**Parameters**

- <T> – the type of the observation’s value

**Constructors****OpenMRSObservation**

public **OpenMRSObservation** (*Date date*, *String conceptName*, *T value*)

Creates an observation that took place on the *date* and recorded a *value* for the concept with the given *conceptName*.

**Parameters**

- **date** – the date when the observation took place
- **conceptName** – the name of the observed concept
- **value** – the value of the observation

**OpenMRSObservation**

public **OpenMRSObservation** (*Date date*, *String conceptName*, *String patientId*, *T value*)

Creates an observation that took place on the *date* and recorded a *value* for the concept with the given *conceptName*. A patient with the given *patientId* was a subject of the observation.

**Parameters**

- **date** – the date when the observation took place
- **conceptName** – the name of the observed concept

- **patientId** – the ID of the patient
- **value** – the value of the observation

### OpenMRSObservation

public **OpenMRSObservation** (*String id*, *Date date*, *String conceptName*, T *value*)

Creates an observation with the given ID that took place on the *date* and recorded a *value* for the concept with the given *conceptName*.

#### Parameters

- **id** – the ID of the observation
- **date** – the date when the observation took place
- **conceptName** – the name of the observed concept
- **value** – the value of the observation

### OpenMRSObservation

public **OpenMRSObservation** (*String id*, *Date date*, *String conceptName*, *String patientId*, T *value*)

Creates an observation with the given ID that took place on the *date* and recorded a *value* for the concept with the given *conceptName*. A patient with the given *patientId* was a subject of the observation.

#### Parameters

- **id** – the ID of the observation
- **date** – the date when the observation took place
- **conceptName** – the name of the observed concept
- **patientId** – the ID of the patient
- **value** – the value of the observation

## Methods

### addDependentObservation

public void **addDependentObservation** (*OpenMRSObservation mrsObservation*)

### equals

public boolean **equals** (*Object o*)

### getConceptName

public *String* **getConceptName** ()

### getDate

public *DateTime* **getDate** ()

### getDependentObservations

public *Set*<? extends *OpenMRSObservation*> **getDependentObservations** ()

**getId**

```
public String getId ()
```

**getObservationId**

```
public String getObservationId ()
```

**getPatientId**

```
public String getPatientId ()
```

**getValue**

```
public T getValue ()
```

**hashCode**

```
public int hashCode ()
```

**setConceptName**

```
public void setConceptName (String conceptName)
```

**setDate**

```
public void setDate (DateTime date)
```

**setDependentObservations**

```
public void setDependentObservations (Set<OpenMRSObservation> dependentObservations)
```

**setObservationId**

```
public void setObservationId (String observationId)
```

**setPatientId**

```
public void setPatientId (String patientId)
```

**setValue**

```
public void setValue (T value)
```

**toString**

```
public String toString ()
```

**OpenMRSPatient**

```
public class OpenMRSPatient
```

Represents a single patient. It's a part of MOTECH model.

**Constructors****OpenMRSPatient**

```
public OpenMRSPatient ()
```

**OpenMRSPatient**

public **OpenMRSPatient** (*String id*)

Creates a patient with the given OpenMRS id.

**Parameters**

- **id** – the OpenMRS ID of the patient

**OpenMRSPatient**

public **OpenMRSPatient** (*String motechId*, *OpenMRSPerson person*, *OpenMRSFacility facility*)

Creates a patient with the given motechId based on the given person details and assigns it to the given facility.

**Parameters**

- **motechId** – the MOTECH ID of the patient
- **person** – the personal details about the patient
- **facility** – the facility by which the patient is treated

**OpenMRSPatient**

public **OpenMRSPatient** (*String id*, *String motechId*, *OpenMRSPerson person*, *OpenMRSFacility facility*)

Creates a patient with the given motechId and OpenMRS id based on the given person details and assigns it to the given facility.

**Parameters**

- **id** – the OpenMRS ID of the patient
- **motechId** – the MOTECH ID of the patient
- **person** – the personal details about the patient
- **facility** – the facility by which the patient is treated

**Methods****equals**

public boolean **equals** (*Object o*)

**getFacility**

public *OpenMRSFacility* **getFacility** ()

**getMotechId**

public *String* **getMotechId** ()

**getPatientId**

public *String* **getPatientId** ()

**getPerson**

public *OpenMRSPerson* **getPerson** ()

**hashCode**

```
public int hashCode ()
```

**setFacility**

```
public void setFacility (OpenMRSFacility facility)
```

**setMotechId**

```
public void setMotechId (String motechId)
```

**setPatientId**

```
public void setPatientId (String id)
```

**setPerson**

```
public void setPerson (OpenMRSPerson person)
```

**OpenMRSPerson**

```
public class OpenMRSPerson
```

Represents a single person. This class stores personal information about the person. It's used by the `OpenMRSPatient`. It's part of the MOTECH model.

**Constructors****OpenMRSPerson**

```
public OpenMRSPerson ()
```

Default constructor.

**OpenMRSPerson**

```
public OpenMRSPerson (String id)
```

Creates a person with the given OpenMRS id.

**Parameters**

- **id** – the OpenMRS ID of the person

**Methods****addAttribute**

```
public void addAttribute (OpenMRSAttribute attribute)
```

**attrValue**

```
public String attrValue (String key)
```

**equalAgeAndBirthDates**

```
public boolean equalAgeAndBirthDates (OpenMRSPerson other)
```

**equalNameData**

public boolean **equalNameData** (*OpenMRSPerson other*)

**equals**

public boolean **equals** (*Object o*)

**getAddress**

public *String* **getAddress** ()

**getAge**

public *Integer* **getAge** ()

**getAttributes**

public *List*<*OpenMRSAttribute*> **getAttributes** ()

**getBirthDateEstimated**

public *Boolean* **getBirthDateEstimated** ()

**getDateOfBirth**

public *DateTime* **getDateOfBirth** ()

**getDeathDate**

public *DateTime* **getDeathDate** ()

**getFirstName**

public *String* **getFirstName** ()

**getFullName**

public *String* **getFullName** ()

**getGender**

public *String* **getGender** ()

**getId**

public *String* **getId** ()

**getLastName**

public *String* **getLastName** ()

**getMiddleName**

public *String* **getMiddleName** ()

**getPersonId**

public *String* **getPersonId** ()



**getPreferredName**

```
public String getPreferredName ()
```

**hashCode**

```
public int hashCode ()
```

**isDead**

```
public Boolean isDead ()
```

**setAddress**

```
public void setAddress (String address)
```

**setAge**

```
public void setAge (Integer age)
```

**setAttributes**

```
public void setAttributes (List<OpenMRSAttribute> attributes)
```

**setBirthDateEstimated**

```
public void setBirthDateEstimated (Boolean birthDateEstimated)
```

**setDateOfBirth**

```
public void setDateOfBirth (DateTime dateOfBirth)
```

**setDeathDate**

```
public void setDeathDate (DateTime deathDate)
```

**setFirstName**

```
public void setFirstName (String firstName)
```

**setGender**

```
public void setGender (String gender)
```

**setId**

```
public void setId (String id)
```

**setIsDead**

```
public void setIsDead (Boolean isDead)
```

**setLastName**

```
public void setLastName (String lastName)
```

**setMiddleName**

```
public void setMiddleName (String middleName)
```

#### **setPersonId**

public void **setPersonId** (*String id*)

#### **setPreferredName**

public void **setPreferredName** (*String preferredName*)

### **OpenMRSPProvider**

public class **OpenMRSPProvider**

Represents a single provider. Provider is a clinician responsible for providing care to a patient. It's part of the MOTECH model.

#### **Constructors**

##### **OpenMRSPProvider**

public **OpenMRSPProvider** ()

Default constructor.

##### **OpenMRSPProvider**

public **OpenMRSPProvider** (*OpenMRSPerson person*)

Creates a provider based on the given *person* details.

##### **Parameters**

- **person** – the personal information about the provider

#### **Methods**

##### **getIdentifier**

public *String* **getIdentifier** ()

##### **getPerson**

public *OpenMRSPerson* **getPerson** ()

##### **getProviderId**

public *String* **getProviderId** ()

##### **setIdentifier**

public void **setIdentifier** (*String identifier*)

##### **setPerson**

public void **setPerson** (*OpenMRSPerson person*)

##### **setProviderId**

public void **setProviderId** (*String providerId*)

## OpenMRSUser

public class **OpenMRSUser**

Represents a single OpenMRS user. It's a part of the MOTECH model.

### Methods

#### **equals**

public boolean **equals** (*Object o*)

#### **getPerson**

public *OpenMRSPerson* **getPerson** ()

#### **getSecurityRole**

public *String* **getSecurityRole** ()

#### **getSystemId**

public *String* **getSystemId** ()

#### **getUserId**

public *String* **getUserId** ()

#### **getUserName**

public *String* **getUserName** ()

#### **hashCode**

public int **hashCode** ()

#### **id**

public *OpenMRSUser* **id** (*String id*)

Sets the ID of this user to *id*.

##### **Parameters**

- **id** – the ID to be set

**Returns** the reference to this object

#### **person**

public *OpenMRSUser* **person** (*OpenMRSPerson person*)

Sets the person of this user to *person*.

##### **Parameters**

- **person** – the person to be set

**Returns** the reference to this object

**securityRole**

public [OpenMRSUser](#) **securityRole** ([String](#) *securityRole*)  
Sets the security role of this user to *securityRole*.

**Parameters**

- **securityRole** – the security role to be set

**Returns** the reference to this object

**setPerson**

public void **setPerson** ([OpenMRSPerson](#) *person*)

**setSecurityRole**

public void **setSecurityRole** ([String](#) *securityRole*)

**setSystemId**

public void **setSystemId** ([String](#) *systemId*)

**setUserId**

public void **setUserId** ([String](#) *userId*)

**setUserName**

public void **setUserName** ([String](#) *userName*)

**systemId**

public [OpenMRSUser](#) **systemId** ([String](#) *systemId*)  
Sets the system ID of this user to *systemId*.

**Parameters**

- **systemId** – the system ID to be set

**Returns** the reference to this object

**userName**

public [OpenMRSUser](#) **userName** ([String](#) *userName*)  
Sets the name of this user to *userName*.

**Parameters**

- **userName** – the user name to be set

**Returns** the reference to this object

**Password**

public class **Password**

Random password generator. It generates a random passwords of length specified in the constructor, built from letters (lower and upper case) and numbers. It's used for generating passwords for new users and users that were modified.

## Constructors

### Password

public **Password** (*Integer length*)

Creates a generator that generates passwords of length *length*.

#### Parameters

- **length** – the length of the password to generate

## Methods

### generate

public *String* **generate** ()

Generates a random password of length specified in the constructor.

**Returns** the generated password

## 19.14.74 org.motechproject.openmrs19.exception

### ConceptNameAlreadyInUseException

public class **ConceptNameAlreadyInUseException** extends *Exception*

Thrown when concept with given name already exists.

## Constructors

### ConceptNameAlreadyInUseException

public **ConceptNameAlreadyInUseException** (*String message*)

### HttpException

public class **HttpException** extends *Exception*

Thrown when HTTP request failed for any reason. Stores the HTTP response status.

## Constructors

### HttpException

public **HttpException** (*String message*, *Throwable cause*)

### HttpException

public **HttpException** (*String message*, *HttpStatus status*)

## Methods

### getStatusCode

public *HttpStatus* **getStatusCode** ()

#### **setStatusCode**

public void **setStatusCode** ([HttpStatus](#) *statusCode*)

### **ObservationNotFoundException**

public class **ObservationNotFoundException** extends [Exception](#)

Thrown when a [OpenMRSObservation](#) is null while performing operations on it. E.g. in voiding an [OpenMR-  
SObservation](#), if observation is not found this exception is thrown.

#### **Constructors**

##### **ObservationNotFoundException**

public **ObservationNotFoundException** ([String](#) *message*)

##### **ObservationNotFoundException**

public **ObservationNotFoundException** ([String](#) *message*, [Throwable](#) *cause*)

### **OpenMRSException**

public class **OpenMRSException** extends [RuntimeException](#)

Thrown when there were problems while fetching data from OpenMRS server.

#### **Constructors**

##### **OpenMRSException**

public **OpenMRSException** ([Throwable](#) *e*)

##### **OpenMRSException**

public **OpenMRSException** ([String](#) *message*)

##### **OpenMRSException**

public **OpenMRSException** ([String](#) *message*, [Throwable](#) *cause*)

### **PatientNotFoundException**

public class **PatientNotFoundException** extends [Exception](#)

Thrown when a patient can't be found on the OpenMRS server while performing operations on it. E.g. in setting the patient as deceased, when Patient is not found this exception is thrown.

#### **Constructors**

##### **PatientNotFoundException**

public **PatientNotFoundException** ([String](#) *message*)

##### **PatientNotFoundException**

public **PatientNotFoundException** ([Exception](#) *e*)

### UserAlreadyExistsException

public class **UserAlreadyExistsException** extends [Exception](#)  
Thrown when trying to create an existing user.

#### Constructors

**UserAlreadyExistsException**  
public **UserAlreadyExistsException** ([String message](#))

### UserDeleteException

public class **UserDeleteException** extends [Exception](#)  
Thrown when there was a problem while deleting user.

#### Constructors

**UserDeleteException**  
public **UserDeleteException** ([String message](#), [Throwable e](#))

## 19.14.75 org.motechproject.openmrs19.service

### EventKeys

public final class **EventKeys**  
Utility class for storing event keys.

#### Fields

**BASE\_SUBJECT**  
public static final [String](#) **BASE\_SUBJECT**

**COMMENT**  
public static final [String](#) **COMMENT**

**CONCEPT\_CONCEPT\_CLASS**  
public static final [String](#) **CONCEPT\_CONCEPT\_CLASS**

**CONCEPT\_DATA\_TYPE**  
public static final [String](#) **CONCEPT\_DATA\_TYPE**

**CONCEPT\_NAME**  
public static final [String](#) **CONCEPT\_NAME**

**CONCEPT\_UUID**  
public static final [String](#) **CONCEPT\_UUID**

**CREATED\_NEW\_CONCEPT\_SUBJECT**

public static final `String` **CREATED\_NEW\_CONCEPT\_SUBJECT**

**CREATED\_NEW\_ENCOUNTER\_SUBJECT**

public static final `String` **CREATED\_NEW\_ENCOUNTER\_SUBJECT**

**CREATED\_NEW\_FACILITY\_SUBJECT**

public static final `String` **CREATED\_NEW\_FACILITY\_SUBJECT**

**CREATED\_NEW\_OBSERVATION\_SUBJECT**

public static final `String` **CREATED\_NEW\_OBSERVATION\_SUBJECT**

**CREATED\_NEW\_PATIENT\_SUBJECT**

public static final `String` **CREATED\_NEW\_PATIENT\_SUBJECT**

**CREATED\_NEW\_PERSON\_SUBJECT**

public static final `String` **CREATED\_NEW\_PERSON\_SUBJECT**

**CREATED\_NEW\_PROVIDER\_SUBJECT**

public static final `String` **CREATED\_NEW\_PROVIDER\_SUBJECT**

**CREATED\_UPDATED\_OBSERVATION\_SUBJECT**

public static final `String` **CREATED\_UPDATED\_OBSERVATION\_SUBJECT**

**DATE\_OF\_DEATH**

public static final `String` **DATE\_OF\_DEATH**

**DELETED\_CONCEPT\_SUBJECT**

public static final `String` **DELETED\_CONCEPT\_SUBJECT**

**DELETED\_ENCOUNTER\_SUBJECT**

public static final `String` **DELETED\_ENCOUNTER\_SUBJECT**

**DELETED\_FACILITY\_SUBJECT**

public static final `String` **DELETED\_FACILITY\_SUBJECT**

**DELETED\_OBSERVATION\_SUBJECT**

public static final `String` **DELETED\_OBSERVATION\_SUBJECT**

**DELETED\_PATIENT\_SUBJECT**

public static final `String` **DELETED\_PATIENT\_SUBJECT**

**DELETED\_PERSON\_SUBJECT**

public static final `String` **DELETED\_PERSON\_SUBJECT**



**DELETED\_PROVIDER\_SUBJECT**

public static final [String](#) **DELETED\_PROVIDER\_SUBJECT**

**ENCOUNTER\_DATE**

public static final [String](#) **ENCOUNTER\_DATE**

**ENCOUNTER\_ID**

public static final [String](#) **ENCOUNTER\_ID**

**ENCOUNTER\_TYPE**

public static final [String](#) **ENCOUNTER\_TYPE**

**FACILITY\_COUNTRY**

public static final [String](#) **FACILITY\_COUNTRY**

**FACILITY\_COUNTY\_DISTRICT**

public static final [String](#) **FACILITY\_COUNTY\_DISTRICT**

**FACILITY\_ID**

public static final [String](#) **FACILITY\_ID**

**FACILITY\_NAME**

public static final [String](#) **FACILITY\_NAME**

**FACILITY\_REGION**

public static final [String](#) **FACILITY\_REGION**

**FACILITY\_STATE\_PROVINCE**

public static final [String](#) **FACILITY\_STATE\_PROVINCE**

**MOTECH\_ID**

public static final [String](#) **MOTECH\_ID**

**OBSERVATION\_CONCEPT\_NAME**

public static final [String](#) **OBSERVATION\_CONCEPT\_NAME**

**OBSERVATION\_DATE**

public static final [String](#) **OBSERVATION\_DATE**

**OBSERVATION\_VALUE**

public static final [String](#) **OBSERVATION\_VALUE**

**PATIENT\_DECEASED\_SUBJECT**

public static final [String](#) **PATIENT\_DECEASED\_SUBJECT**

**PATIENT\_ID**

public static final `String` **PATIENT\_ID**

**PERSON\_ADDRESS**

public static final `String` **PERSON\_ADDRESS**

**PERSON\_AGE**

public static final `String` **PERSON\_AGE**

**PERSON\_BIRTH\_DATE\_ESTIMATED**

public static final `String` **PERSON\_BIRTH\_DATE\_ESTIMATED**

**PERSON\_DATE\_OF\_BIRTH**

public static final `String` **PERSON\_DATE\_OF\_BIRTH**

**PERSON\_DEAD**

public static final `String` **PERSON\_DEAD**

**PERSON\_DEATH\_DATE**

public static final `String` **PERSON\_DEATH\_DATE**

**PERSON\_FIRST\_NAME**

public static final `String` **PERSON\_FIRST\_NAME**

**PERSON\_GENDER**

public static final `String` **PERSON\_GENDER**

**PERSON\_ID**

public static final `String` **PERSON\_ID**

**PERSON\_LAST\_NAME**

public static final `String` **PERSON\_LAST\_NAME**

**PERSON\_MIDDLE\_NAME**

public static final `String` **PERSON\_MIDDLE\_NAME**

**PERSON\_PREFERRED\_NAME**

public static final `String` **PERSON\_PREFERRED\_NAME**

**PROVIDER\_ID**

public static final `String` **PROVIDER\_ID**

**UPDATED\_CONCEPT\_SUBJECT**

public static final `String` **UPDATED\_CONCEPT\_SUBJECT**

**UPDATED\_ENCOUNTER\_SUBJECT**

public static final [String](#) **UPDATED\_ENCOUNTER\_SUBJECT**

**UPDATED\_FACILITY\_SUBJECT**

public static final [String](#) **UPDATED\_FACILITY\_SUBJECT**

**UPDATED\_PATIENT\_SUBJECT**

public static final [String](#) **UPDATED\_PATIENT\_SUBJECT**

**UPDATED\_PERSON\_SUBJECT**

public static final [String](#) **UPDATED\_PERSON\_SUBJECT**

**UPDATED\_PROVIDER\_SUBJECT**

public static final [String](#) **UPDATED\_PROVIDER\_SUBJECT**

**USER\_ID**

public static final [String](#) **USER\_ID**

**VOIDED\_OBSERVATION\_SUBJECT**

public static final [String](#) **VOIDED\_OBSERVATION\_SUBJECT**

**OpenMRSService**

public interface **OpenMRSService**

Interface for handling concepts on the OpenMRS server.

**Methods****createConcept**

[OpenMRSService](#) **createConcept** ([OpenMRSService](#) *concept*)

Creates the given concept on the OpenMRS server.

**Parameters**

- **concept** – the concept to be created

**Throws**

- [ConceptNameAlreadyInUseException](#) – if the concept with the given name already exists

**Returns** the created concept

**deleteConcept**

void **deleteConcept** ([String](#) *uuid*)

Deletes the concept with the given *uuid* from the OpenMRS server.

**Parameters**

- **uuid** – the UUID of the concept

**getAllConcepts**`List<OpenMRSCConcept> getAllConcepts ()`

Returns the basic information (UUID and name) about all the concepts stored on the OpenMRS server.

**Returns** the list of all concepts

**getConceptByUuid**`OpenMRSCConcept getConceptByUuid (String uuid)`

Returns the concept with the given `uuid`.

**Parameters**

- **uuid** – the UUID of the concept

**Returns** the concept with the given UUID

**getConcepts**`List<OpenMRSCConcept> getConcepts (int page, int pageSize)`

Returns a list of the concepts. The returned list will contain maximum of `pageSize` concepts fetched from the page with the given `page` number. The list might contain less entries if the given `page` is the last one and there is less than `pageSize` concepts on it.

**Parameters**

- **page** – the number of the page
- **pageSize** – the size of the page

**Returns** the list of concepts on the given page

**resolveConceptUuidFromConceptName**`String resolveConceptUuidFromConceptName (String name)`

Returns the UUID of the concept with the given `name`.

**Parameters**

- **name** – the name of the concept

**Returns** the UUID of the concept

**search**`List<OpenMRSCConcept> search (String phrase)`

Returns a list of the concepts with the given `phrase` in their names.

**Parameters**

- **phrase** – the phrase to search for

**Returns** the list of concept with names containing the given phrase

**updateConcept**`OpenMRSCConcept updateConcept (OpenMRSCConcept concept)`

Updates the concept with the information stored in the given concept. Fields `name`, `names` and `display name` will be ignored as changing them is not allowed by the OpenMRS system.

**Parameters**

- **concept** – the concept used as an update source

**Returns** the updated concept

## OpenMRSEncounterService

public interface **OpenMRSEncounterService**

Interface for handling encounters on the OpenMRS server.

### Methods

#### createEncounter

`OpenMRSEncounter` **createEncounter** (`OpenMRSEncounter` *encounter*)

Creates the given *encounter* on the OpenMRS server.

##### Parameters

- **encounter** – the encounter to be created

**Returns** the created encounter

#### createEncounterType

`OpenMRSEncounterType` **createEncounterType** (`OpenMRSEncounterType` *encounterType*)

Creates the given *encounterType* on the OpenMRS server.

##### Parameters

- **encounterType** – the encounter type to be created

**Returns** the created encounter type

#### deleteEncounter

void **deleteEncounter** (`String` *uuid*)

Deletes the encounter with the given *uuid* from the OpenMRS server.

##### Parameters

- **uuid** – the UUID of the encounter

#### deleteEncounterType

void **deleteEncounterType** (`String` *uuid*)

Deletes the encounter type with the given *uuid*.

##### Parameters

- **uuid** – the UUID of the encounter type

#### getEncounterByUuid

`OpenMRSEncounter` **getEncounterByUuid** (`String` *uuid*)

Returns the encounter with the given *uuid*.

##### Parameters

- **uuid** – the UUID of the encounter

**Returns** the encounter with the given UUID

**getEncounterTypeByUuid**

`OpenMRSEncounterType` **getEncounterTypeByUuid** (`String uuid`)

Returns the encounter type with the given `uuid`.

**Parameters**

- **uuid** – the UUID of the encounter type

**Returns** the encounter type with given UUID

**getEncountersByEncounterType**

`List<OpenMRSEncounter>` **getEncountersByEncounterType** (`String motechId`, `String encounterType`)

Returns a list of encounters of the given `encounterType` for a patient with the given `motechId`.

**Parameters**

- **motechId** – the MOTech ID of the patient
- **encounterType** – the type of the encounter

**Returns** the list of encounters with the given encounter type for a patient with the given MOTech ID

**getLatestEncounterByPatientMotechId**

`OpenMRSEncounter` **getLatestEncounterByPatientMotechId** (`String motechId`, `String encounterType`)

Returns the latest encounter of type `encounterType` for a patient with the given `motechId`.

**Parameters**

- **motechId** – the MOTech ID of the patient
- **encounterType** – the type of the encounter

**Returns** the latest encounter of the given type for the patient with the given MOTech ID

**OpenMRSFacilityService**

public interface **OpenMRSFacilityService**

Interface for handling facilities on the OpenMRS server.

**Methods****createFacility**

`OpenMRSFacility` **createFacility** (`OpenMRSFacility facility`)

Creates the given `facility` on the OpenMRS server.

**Parameters**

- **facility** – the facility to be created

**Returns** the created facility

**deleteFacility**

void **deleteFacility** (*String uuid*)

Deletes the facility with the given `uuid` from the OpenMRS server.

**Parameters**

- **uuid** – the UUID of the facility

**getAllFacilities**

List<? extends [OpenMRSFacility](#)> **getAllFacilities** ()

Returns a list of all the facilities on the OpenMRS server.

**Returns** the list of all facilities

**getFacilities**

List<? extends [OpenMRSFacility](#)> **getFacilities** (int *page*, int *pageSize*)

Returns a list of the facilities. The returned list will contain maximum of `pageSize` facilities fetched from the page with the given `page` number. The list might contain less entries if the given `page` is the last one and there is less than `pageSize` facilities on it.

**Parameters**

- **page** – the number of the page
- **pageSize** – the size of the page

**Returns** the list of facilities on the given page

**getFacilities**

List<? extends [OpenMRSFacility](#)> **getFacilities** (*String locationName*)

Returns a list of facilities that have the given `locationName`.

**Parameters**

- **locationName** – the name of the location

**Returns** the list of matching facilities

**getFacilityByUuid**

[OpenMRSFacility](#) **getFacilityByUuid** (*String uuid*)

Returns the facility with the given `uuid`.

**Parameters**

- **uuid** – the UUID of the facility

**Returns** the facility with the given UUID

**updateFacility**

[OpenMRSFacility](#) **updateFacility** ([OpenMRSFacility](#) *facility*)

Updates the facility with the information stored in the given `facility`.

**Parameters**

- **facility** – the facility to be used as an update source

**Returns** the updated facility, null if a facility with the UUID given in the `facility` doesn't exist

## OpenMRSObservationService

public interface **OpenMRSObservationService**

Interface for handling observations on the OpenMRS server.

### Methods

#### createObservation

**OpenMRSObservation createObservation** (**OpenMRSObservation** *observation*)

Creates the given *observation* on the OpenMRS server.

##### Parameters

- **observation** – the observation to be created

**Returns** the created observation

#### deleteObservation

void **deleteObservation** (**String** *uuid*)

Deletes the observation with the given *uuid* from the OpenMRS server. If the observation with the given *uuid* doesn't exist an error will be logged.

##### Parameters

- **uuid** – the UUID of the observation

#### findObservation

**OpenMRSObservation findObservation** (**String** *motechId*, **String** *conceptName*)

Returns the latest observation of the concept with the given *conceptName* for a patient with the given *motechId*.

##### Parameters

- **motechId** – the MOTECH ID of the patient
- **conceptName** – the name of the observation concept

**Returns** the latest observation for the given concept name and patient MOTECH ID

#### findObservations

**List<OpenMRSObservation> findObservations** (**String** *patientMotechId*, **String** *conceptName*)

Returns a list of observations of the concept with the given *conceptName* for a patient with the given *motechId*.

##### Parameters

- **patientMotechId** – the MOTECH ID of the patient
- **conceptName** – the name of the observation concept

**Returns** the list of observations for the given concept name and patient MOTECH ID

#### getObservationByUuid

**OpenMRSObservation getObservationByUuid** (**String** *uuid*)

Returns the observation with given *uuid*.

##### Parameters



- **uuid** – the UUID of the observation

**Returns** the observation with the given UUID, null if the observation doesn't exist

#### **voidObservation**

void **voidObservation** ([OpenMRSObservation](#) *observation*, [String](#) *reason*)

Voids the given *observation* with the given *reason*.

##### **Parameters**

- **observation** – the observation to be voided
- **reason** – the reason for voiding the observation

##### **Throws**

- [ObservationNotFoundException](#) – if the observation doesn't exist

### **OpenMRSPatientService**

public interface **OpenMRSPatientService**

Interface for handling patients on the OpenMRS server.

#### **Methods**

##### **createPatient**

[OpenMRSPatient](#) **createPatient** ([OpenMRSPatient](#) *patient*)

Creates the given *patient* on the OpenMRS server.

##### **Parameters**

- **patient** – the patient to be created

**Returns** the created patient

##### **deceasePatient**

void **deceasePatient** ([String](#) *motechId*, [OpenMRSConcept](#) *causeOfDeath*, [Date](#) *dateOfDeath*, [String](#) *comment*)

Marks a patient with the given *motechId* as dead with the given *dateOfDeath*, *causeOfDeath* and a *comment*.

##### **Parameters**

- **motechId** – the MOTECH ID of the patient
- **causeOfDeath** – the cause of death
- **dateOfDeath** – the date of death
- **comment** – the additional information for the cause of death

##### **Throws**

- [PatientNotFoundException](#) – if the patient with the given MOTECH ID doesn't exist

**deletePatient**

void **deletePatient** (*String uuid*)

Deletes the patient with the given *uuid*.

**Parameters**

- **uuid** – the UUID of the patient

**Throws**

- **PatientNotFoundException** – if the patient with the given UUID doesn't exist

**getPatientByMotechId**

*OpenMRSPatient* **getPatientByMotechId** (*String motechId*)

Returns the patient with the given *motechId*.

**Parameters**

- **motechId** – the MOTECH ID of the patient

**Returns** the patient with the given MOTECH ID, null if the patient doesn't exist

**getPatientByUuid**

*OpenMRSPatient* **getPatientByUuid** (*String uuid*)

Returns the patient with the given *uuid*.

**Parameters**

- **uuid** – the UUID of the patient

**Returns** the patient with the given UUID, null if the patient doesn't exist

**search**

*List<OpenMRSPatient>* **search** (*String name*, *String motechId*)

If the *motechId* is null this method will return a list of patients with given *name*, else it will return a list with a single patient that has the given *name* and *motechId*. If there are no matching patients an empty list will be returned.

**Parameters**

- **name** – the name of the patient to be searched for
- **motechId** – the MOTECH ID of the patient to be searched for

**Returns** list of matched patients

**updatePatient**

*OpenMRSPatient* **updatePatient** (*OpenMRSPatient patient*, *String currentMotechId*)

Updates the patient with the given *currentMotechId* with the information stored in the given *patient* (including the new MOTECH ID passed in the given *patient*).

**Parameters**

- **patient** – the patient to be used as an update source
- **currentMotechId** – the current MOTECH ID of the patient to update (used for searching)

**Returns** the updated patient

**updatePatient**

`OpenMRSPatient` **updatePatient** (`OpenMRSPatient` *patient*)

Updates the patient with the information stored in the given *patient*.

**Parameters**

- **patient** – the patient to be used as an update source

**Returns** the updated patient

**OpenMRSPersonService**

public interface **OpenMRSPersonService**

Interface for handling persons on the OpenMRS server.

**Methods****createPerson**

`OpenMRSPerson` **createPerson** (`OpenMRSPerson` *person*)

Creates the given person on the OpenMRS server.

**Parameters**

- **person** – the person to be created

**Returns** the created person

**createPerson**

`OpenMRSPerson` **createPerson** (`String` *firstName*, `String` *lastName*, `DateTime` *dateOfBirth*, `String` *gender*, `String` *address*, `List`<`OpenMRSAttribute`> *attributes*)

Creates a person on the OpenMRS server based on the given information.

**Parameters**

- **firstName** – the person's first name
- **lastName** – the person's last name
- **dateOfBirth** – the person's date of birth
- **gender** – the person's gender
- **address** – the address of the person
- **attributes** – a list of attributes for the person

**deletePerson**

void **deletePerson** (`String` *uuid*)

Deletes the person with the given *uuid* from the OpenMRS server. If the person with the given *uuid* doesn't exist an error will be logged.

**Parameters**

- **uuid** – the UUID of the person

**getPersonByUuid****OpenMRSPerson** **getPersonByUuid** (*String uuid*)Returns the person with the given *uuid*.**Parameters**

- **uuid** – the UUID of the person

**Returns** the person with the given UUID, null if the person doesn't exist**updatePerson****OpenMRSPerson** **updatePerson** (*OpenMRSPerson person*)Updates the person with the information stored in the given *person*.**Parameters**

- **person** – the person to be used as an update source

**Returns** the updated person**OpenMRSProviderService**public interface **OpenMRSProviderService**

Interface for handling providers on the OpenMRS server.

**Methods****createProvider****OpenMRSProvider** **createProvider** (*OpenMRSProvider provider*)

Creates the given provider on the OpenMRS server.

**Parameters**

- **provider** – the provider to be created

**Returns** the created provider**deleteProvider**void **deleteProvider** (*String uuid*)Deletes the provider with the given *uuid*. If the provider with the given *uuid* doesn't exist an error will be logged.**Parameters**

- **uuid** – the UUID of the provider

**getProviderByUuid****OpenMRSProvider** **getProviderByUuid** (*String uuid*)Returns the provider with the given *uuid*.**Parameters**

- **uuid** – the UUID of the provider

**Returns** the provider with the given UUID, null if the provider doesn't exist

## OpenMRSUserService

public interface **OpenMRSUserService**

Interface for handling users on the OpenMRS server.

### Fields

#### **PASSWORD\_KEY**

*String* **PASSWORD\_KEY**

Key to obtain password from `createUser (OpenMRSUser)` and `updateUser (OpenMRSUser)`

#### **USER\_KEY**

*String* **USER\_KEY**

Key to obtain `OpenMRSUser` from `createUser (OpenMRSUser)` and `updateUser (OpenMRSUser)`

### Methods

#### **changeCurrentUserPassword**

void **changeCurrentUserPassword** (*String* `currentPassword`, *String* `newPassword`)

Changes the password of the current user.

##### **Parameters**

- **currentPassword** – the current password
- **newPassword** – the new password

#### **createUser**

`OpenMRSUser` **createUser** (`OpenMRSUser` `user`)

Creates the given `user` on the OpenMRS server.

##### **Parameters**

- **user** – the user to be created

##### **Throws**

- **UserAlreadyExistsException** – if the user already exists

**Returns** the created user

#### **deleteUser**

void **deleteUser** (*String* `uuid`)

Deletes the user with the given `uuid`. If the provider with the given `uuid` doesn't exist an error will be logged.

##### **Parameters**

- **uuid** – the UUID of the user

##### **Throws**

- **UserDeleteException** – if there were problems while deleting user

**getAllUsers**`List<OpenMRSUser> getAllUsers ()`

Returns a list of all users on the OpenMRS server.

**Returns** the list of all users

**getUserByUserName**`OpenMRSUser getUserByUserName (String userName)`

Returns the user with the given `userName`.

**Parameters**

- **userName** – the username of the user

**Returns** the user with the given username, null if the user doesn't exist

**getUserByUuid**`OpenMRSUser getUserByUuid (String uuid)`

Return the user with the given `uuid`.

**Parameters**

- **uuid** – the UUID of the user

**Returns** the user with the given UUID, null if the user doesn't exist

**setNewPasswordForUser**`String setNewPasswordForUser (String username)`

Resets the password of the user with the given `username`.

**Parameters**

- **username** – the name of the user

**Returns** the new password, null if the user doesn't exist

**updateUser**`OpenMRSUser updateUser (OpenMRSUser user)`

Updates the user with the information stored in the given `user`.

**Parameters**

- **user** – the user to be used as an update source

**Returns** the update user, null if the user doesn't exist

## 19.14.76 org.motechproject.pillreminder

**EventKeys**

public final class **EventKeys**

Event keys and subjects used by the module.

## Fields

### **BASE\_SUBJECT**

public static final [String](#) **BASE\_SUBJECT**  
The base for Pill-Reminder subjects.

### **DOSAGE\_ID\_KEY**

public static final [String](#) **DOSAGE\_ID\_KEY**  
The key for Dosage ID key.

### **EXTERNAL\_ID\_KEY**

public static final [String](#) **EXTERNAL\_ID\_KEY**  
The key for External ID key.

### **LAST\_CAPTURE\_DATE**

public static final [String](#) **LAST\_CAPTURE\_DATE**  
The key for the last capture date of the dosage.

### **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT**

public static final [String](#) **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT**  
This subject signals that a pill reminder should be sent to the patient.

### **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_DOSAGE\_STATUS\_KNOWN**

public static final [String](#) **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_DOSAGE\_STATUS\_KNOWN**  
The subject the pill reminder listens for. Sending an event with this subject will update the last capture date for a dosage. This subject is used by a task action from the pill-reminder channel.

### **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_SCHEDULER**

public static final [String](#) **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_SCHEDULER**  
The subject used internally by pill reminder. Used for scheduler jobs.

### **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_UNSUBSCRIBE**

public static final [String](#) **PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_UNSUBSCRIBE**  
The subject the pill reminder listens for. Sending an event with this subject will unsubscribe a patient from the reminder service. This subject is used by a task action from the pill-reminder channel.

### **PILLREMINDER\_RETRY\_INTERVAL**

public static final [String](#) **PILLREMINDER\_RETRY\_INTERVAL**  
They key for the retry interval (in minutes) for repeating reminders for this dosage.

### **PILLREMINDER\_TIMES\_SENT**

public static final [String](#) **PILLREMINDER\_TIMES\_SENT**  
The key for the number of pill reminders already sent for a dosage.

### **PILLREMINDER\_TOTAL\_TIMES\_TO\_SEND**

public static final [String](#) **PILLREMINDER\_TOTAL\_TIMES\_TO\_SEND**  
The key for the total number of pill reminders that can be sent for a dosage.

**PILL\_REGIMEN\_ID**

public static final [String](#) **PILL\_REGIMEN\_ID**

The Pill Regimen ID key.

## 19.14.77 org.motechproject.pillreminder.contract

### DailyPillRegimenRequest

public class **DailyPillRegimenRequest**

Daily Pill Regimen Request represents medicine prescription details and reminder configuration.

**See also:** [DosageRequest](#)

#### Constructors

##### DailyPillRegimenRequest

public **DailyPillRegimenRequest** ([String](#) *externalId*, int *pillWindowInHours*, int *reminderRepeatIntervalInMinutes*, int *bufferTimeForPatientToTakePill*, [List](#)<[DosageRequest](#)> *dosageRequests*)

Creates Daily Pill prescription for given subscriber id and reminder configuration

##### Parameters

- **externalId** – unique id for subscriber
- **pillWindowInHours** – valid time range to retry the reminder in-case of subscriber unreachable, in hours
- **reminderRepeatIntervalInMinutes** – wait time before 2 reminder retries
- **bufferTimeForPatientToTakePill** – additional wait time before sending reminder. (Time to take pill typically 5 min) when set to 5, If pill time is 9:00 am then reminder will be set out at 9:05 am.
- **dosageRequests** – Dosage detail with dose time and medicine list. [DosageRequest](#)

#### Methods

##### getBufferOverDosageTimeInMinutes

public int **getBufferOverDosageTimeInMinutes** ()

Returns the buffer over dosage time in minutes. The first reminder will be fired after this time passes from the dosage time if no there was no adherence confirmation.

**Returns** the time in minutes after the first reminder will be sent, if no adherence confirmation was received for the regimen

##### getDosageRequests

public [List](#)<[DosageRequest](#)> **getDosageRequests** ()

**Returns** all dosage requests scheduled for pill reminder

##### getExternalId

public [String](#) **getExternalId** ()

**Returns** unique Id representing subscriber



**getPillWindowInHours**

```
public int getPillWindowInHours ()
```

**Returns** valid time range to retry the reminder in-case of subscriber unreachable, in hours

**getReminderRepeatIntervalInMinutes**

```
public int getReminderRepeatIntervalInMinutes ()
```

Repeat interval for dosage reminder in-case not able reach subscriber.

**Returns** number of minutes to wait before next retry for the reminder.

**setBufferOverDosageTimeInMinutes**

```
public void setBufferOverDosageTimeInMinutes (int bufferOverDosageTimeInMinutes)
```

Sets the buffer over dosage time in minutes. The first reminder will be fired after this time passes from the dosage time if there was no adherence confirmation.

**Parameters**

- **bufferOverDosageTimeInMinutes** – the time in minutes after the first reminder will be sent, if no adherence confirmation was received for the regiment

**DosageRequest**

```
public class DosageRequest
```

Dosage Request represents Dosage schedule and medicine list for the schedule. `startHour` and `startMinute` represent dosage timings.

**See also:** [MedicineRequest](#)

**Constructors****DosageRequest**

```
public DosageRequest (int startHour, int startMinute, List<MedicineRequest> medicineRequests)
```

Constructs a request.

**Parameters**

- **startHour** – dosage time hour component
- **startMinute** – dosage time minute component
- **medicineRequests** – medicines prescribed for this dose

**Methods****getMedicineRequests**

```
public List<MedicineRequest> getMedicineRequests ()
```

**Returns** medicines prescribed for this dose

**getStartHour**

```
public int getStartHour ()
```

**Returns** dosage time hour component

**getStartMinute**

```
public int getStartMinute ()
```

**Returns** dosage time minute component

**DosageResponse**

```
public class DosageResponse
```

Dosage details returned from pill reminder module, represents single dosage detail (morning or evening) Also contains information about last dosage confirmation.

**Constructors****DosageResponse**

```
public DosageResponse (Long dosageId, Time dosageTime, LocalDate startDate, LocalDate endDate, LocalDate responseLastCapturedDate, List<MedicineResponse> medicines)
```

Constructs a dosage response.

**Parameters**

- **dosageId** – unique dosage identifier
- **dosageTime** – the time of the dosage
- **startDate** – dosage start date, if there are multiple medicines with different start date then first start date will be returned
- **endDate** – dosage end date, if there are multiple medicines with different end date then last ending will be returned
- **responseLastCapturedDate** – last dosage confirmation from subscriber for this dose
- **medicines** – medicines prescribed

**Methods****getDosageHour**

```
public int getDosageHour ()
```

Dosage time – hour component

**getDosageId**

```
public Long getDosageId ()
```

Unique dosage identifier

**getDosageMinute**

```
public int getDosageMinute ()
```

Dosage time – minute component

**getEndDate**

```
public LocalDate getEndDate ()
```

Dosage end date, if there are multiple medicines with different end date then last ending will be returned

**getMedicines**

```
public List<MedicineResponse> getMedicines ()
```

**Returns** Medicines prescribed

**getResponseLastCapturedDate**

```
public LocalDate getResponseLastCapturedDate ()
```

**Returns** Last dosage confirmation from subscriber for this dose.

**getStartDate**

```
public LocalDate getStartDate ()
```

Dosage start date, if there are multiple medicines with different start date then first start date will be returned

**MedicineRequest**

```
public class MedicineRequest
```

MedicineRequest specifies medicine name and prescription effective date range.

**Constructors****MedicineRequest**

```
public MedicineRequest (String name, LocalDate startDate, LocalDate endDate)
```

Constructs a medicine request.

**Parameters**

- **name** – the name of the medicine
- **startDate** – the start date for taking this medicine
- **endDate** – the end date for taking this medicine

**Methods****getEndDate**

```
public LocalDate getEndDate ()
```

**Returns** the end date for taking this medicine

**getName**

```
public String getName ()
```

**Returns** the name of the medicine

**getStartDate**

```
public LocalDate getStartDate ()
```

**Returns** the start date for taking this medicine

## MedicineResponse

public class **MedicineResponse**

Medicine details returned from the pill-reminder module.

**See also:** [DosageResponse](#)

### Constructors

#### MedicineResponse

public **MedicineResponse** ([String](#) name, [LocalDate](#) startDate, [LocalDate](#) endDate)

Constructs a medicine response.

##### Parameters

- **name** – the name of the medicine
- **startDate** – the start date for taking this medicine
- **endDate** – the end date for taking this medicine

### Methods

#### getEndDate

public [LocalDate](#) **getEndDate** ()

**Returns** the end date for taking this medicine

#### getName

public [String](#) **getName** ()

**Returns** the name of the medicine

#### getStartDate

public [LocalDate](#) **getStartDate** ()

**Returns** the start date for taking this medicine

## PillRegimenResponse

public class **PillRegimenResponse**

PillRegimenResponse represents subscribers dosage prescription and reminder configuration.

**See also:** `org.motechproject.pillreminder.service.PillReminderService.getPillRegimen(String)`

### Constructors

#### PillRegimenResponse

public **PillRegimenResponse** ([Long](#) pillRegimenId, [String](#) externalId, int reminderRepeatWindowInHours, int reminderRepeatIntervalInMinutes, int bufferOverDosageTimeInMinutes, [List](#)<[DosageResponse](#)> dosages)

Constructs a pill regimen response.

##### Parameters

- **pillRegimenId** – unique identifier of the pill regimen
- **externalId** – the external ID associated with this regimen, can represent a patient
- **reminderRepeatWindowInHours** – the number of hours during which reminders will be sent
- **reminderRepeatIntervalInMinutes** – the interval between sending subsequent reminders in minutes
- **bufferOverDosageTimeInMinutes** – the time in minutes after the first reminder will be sent, if no adherence confirmation was received for the regimen
- **dosages** – medicine dosages in this pill regimen

## Methods

### **getBufferOverDosageTimeInMinutes**

public int **getBufferOverDosageTimeInMinutes** ()

Returns the buffer over dosage time in minutes. The first reminder will be fired after this time passes from the dosage time if there was no adherence confirmation.

**Returns** the time in minutes after the first reminder will be sent, if no adherence confirmation

### **getDosages**

public [List<DosageResponse>](#) **getDosages** ()

**Returns** medicine dosages in this pill regimen

### **getExternalId**

public [String](#) **getExternalId** ()

**Returns** the external ID associated with this regimen, can represent a patient

### **getPillRegimenId**

public [Long](#) **getPillRegimenId** ()

**Returns** unique identifier of the pill regimen

### **getReminderRepeatIntervalInMinutes**

public int **getReminderRepeatIntervalInMinutes** ()

**Returns** the interval between sending subsequent reminders in minutes

### **getReminderRepeatWindowInHours**

public int **getReminderRepeatWindowInHours** ()

Returns the time window in hours during which reminders for this regimen will be sent. No reminders should be sent after this time passes from the start of sending reminders.

**Returns** the number of hours during which reminders will be sent

**setBufferOverDosageTimeInMinutes**

public void **setBufferOverDosageTimeInMinutes** (int *bufferOverDosageTimeInMinutes*)

Sets the buffer over dosage time in minutes. The first reminder will be fired after this time passes from the dosage time if there was no adherence confirmation.

**Parameters**

- **bufferOverDosageTimeInMinutes** – the time in minutes after the first reminder will be sent, if no adherence confirmation

## 19.14.78 org.motechproject.pillreminder.dao

**PillRegimenDataService**

public interface **PillRegimenDataService** extends MotechDataService<**PillRegimen**>

The data service for managing `org.motechproject.pillreminder.domain.PillRegimen` objects. The service implementation is generated and injected by Motech Data Services.

**Methods****findByExternalId**

**PillRegimen** **findByExternalId** (String *externalId*)

Finds a pill regimen by external ID.

**Parameters**

- **externalId** – the external ID of the pill regimen

**Returns** the matching pill regimen

## 19.14.79 org.motechproject.pillreminder.domain

**DailyScheduleDetails**

public class **DailyScheduleDetails**

Domain object representing the schedule of sending daily pill reminders. This is tied to a pill regimen.

**Constructors****DailyScheduleDetails**

public **DailyScheduleDetails** ()

Constructs an instance without setting any fields.

**DailyScheduleDetails**

public **DailyScheduleDetails** (int *repeatIntervalInMinutes*, int *pillWindowInHours*, int *bufferOverDosageTimeInMinutes*)

Constructs an instance of the schedule.

**Parameters**

- **repeatIntervalInMinutes** – number of minutes to wait before next retry for the reminder
- **pillWindowInHours** – valid time range to retry the reminder in-case of subscriber unreachable, in hours

- **bufferOverDosageTimeInMinutes** – the time in minutes after the first reminder will be sent, if no adherence confirmation

## Methods

### **getBufferOverDosageTimeInMinutes**

public int **getBufferOverDosageTimeInMinutes** ()

Returns the buffer over dosage time in minutes. The first reminder will be fired after this time passes from the dosage time if no there was no adherence confirmation.

**Returns** the time in minutes after the first reminder will be sent, if no adherence confirmation

### **getPillWindowInHours**

public int **getPillWindowInHours** ()

**Returns** valid time range to retry the reminder in-case of subscriber unreachable, in hours

### **getRepeatIntervalInMinutes**

public int **getRepeatIntervalInMinutes** ()

**Returns** number of minutes to wait before next retry for the reminder

### **setBufferOverDosageTimeInMinutes**

public void **setBufferOverDosageTimeInMinutes** (int *bufferOverDosageTimeInMinutes*)

Sets the buffer over dosage time in minutes. The first reminder will be fired after this time passes from the dosage time if no there was no adherence confirmation.

#### **Parameters**

- **bufferOverDosageTimeInMinutes** – the time in minutes after the first reminder will be sent, if no adherence confirmation

### **setPillWindowInHours**

public void **setPillWindowInHours** (int *pillWindowInHours*)

#### **Parameters**

- **pillWindowInHours** – valid time range to retry the reminder in-case of subscriber unreachable, in hours

### **setRepeatIntervalInMinutes**

public void **setRepeatIntervalInMinutes** (int *repeatIntervalInMinutes*)

#### **Parameters**

- **repeatIntervalInMinutes** – number of minutes to wait before next retry for the reminder

## Dosage

public class **Dosage**

The domain object representing a medicine dosage within a pill regimen.

## Constructors

### Dosage

public **Dosage** ()

Constructs an instance without setting any fields.

### Dosage

public **Dosage** (Time *dosageTime*, Set<Medicine> *medicines*)

Creates a dosage using a medicine and a time of day for taking the medicines.

#### Parameters

- **dosageTime** – the time of day at which this dosage should be taken
- **medicines** – all medicines that are part of this dosage

## Methods

### getDosageTime

public Time **getDosageTime** ()

**Returns** the time of day at which this dosage should be taken

### getEndDate

public LocalDate **getEndDate** ()

Gets the end date for taking this dosage. It will check all the medicines for the latest end date.

**Returns** the end date for taking this dosage, or null if there are no medicines with end dates

### getId

public Long **getId** ()

**Returns** the unique dosage identifier

### getMedicines

public Set<Medicine> **getMedicines** ()

**Returns** all medicines that are part of this dosage

### getResponseLastCapturedDate

public LocalDate **getResponseLastCapturedDate** ()

**Returns** the last captured date when the patient reported taking this dosage

### getStartDate

public LocalDate **getStartDate** ()

Gets the start date for taking this dosage. It will check all the medicines for the earliest start date.

**Returns** the start date for taking this dosage



**isToday'sDosageResponseCaptured**

public boolean **isToday'sDosageResponseCaptured** ()

Checks if there is a patient response captured for the last dosage time. This will check if the response was made today or yesterday, provided it's still before today's dosage time.

**Returns** true if the response was captured, false otherwise

**setDosageTime**

public void **setDosageTime** (Time *dosageTime*)

**Parameters**

- **dosageTime** – the time of day at which this dosage should be taken

**setId**

public void **setId** (Long *id*)

**Parameters**

- **id** – the unique dosage identifier

**setMedicines**

public void **setMedicines** (Set<Medicine> *medicines*)

**Parameters**

- **medicines** – all medicines that are part of this dosage

**setResponseLastCapturedDate**

public void **setResponseLastCapturedDate** (LocalDate *responseLastCapturedDate*)

**Parameters**

- **responseLastCapturedDate** – the last captured date when the patient reported taking this dosage

**today'sDosageTime**

public DateTime **today'sDosageTime** ()

Returns a datetime for the dosage that should be taken today, by combining the today date and the dosage time.

**Returns** a datetime for the dosage of today

**updateResponseLastCapturedDate**

public void **updateResponseLastCapturedDate** (LocalDate *lastCapturedDate*)

Update this dosage with a new response date from the patient. If the provided date is null or before the currently set last capture date, it will be ignored.

**Parameters**

- **lastCapturedDate** – the date on which the response was captured

**validate**

```
public void validate ()
```

Validates this dosage by calling `Medicine.validate()` on all medicines for this dosage.

**Throws**

- **ValidationException** – if one of the medicines fails validation

**Medicine**

```
public class Medicine
```

The domain object representing a single medicine in a dosage that is part of a pill regimen.

**Fields****MEDICINE\_END\_DATE\_CANNOT\_BE\_BEFORE\_START\_DATE**

```
public static final String MEDICINE_END_DATE_CANNOT_BE_BEFORE_START_DATE
```

Error message for medicine validation.

**Constructors****Medicine**

```
public Medicine ()
```

Constructs an instance without setting any fields.

**Medicine**

```
public Medicine (String name, LocalDate startDate, LocalDate endDate)
```

Creates a medicine instance using the provided data.

**Parameters**

- **name** – the name of the medicine
- **startDate** – the start date for taking the medicine
- **endDate** – the end date for taking this medicine, null means never-ending

**Methods****equals**

```
public boolean equals (Object o)
```

**getEndDate**

```
public LocalDate getEndDate ()
```

**Returns** the end date for taking the medicine, null means never-ending

**getName**

```
public String getName ()
```

**Returns** the name of the medicine

**getStartDate**

```
public LocalDate getStartDate ()
```

**Returns** the start date for taking the medicine

**hashCode**

```
public int hashCode ()
```

**setEndDate**

```
public void setEndDate (LocalDate endDate)
```

**Parameters**

- **endDate** – the end date for taking the medicine, null means never-ending

**setName**

```
public void setName (String name)
```

**Parameters**

- **name** – the name of the medicine

**setStartDate**

```
public void setStartDate (LocalDate startDate)
```

**Parameters**

- **startDate** – the start date for taking the medicine

**validate**

```
public void validate ()
```

Validates this medicine, by checking if the start date is not after the end date.

**Throws**

- [ValidationException](#) – if the start date is afeter the end date

**[PillRegimen](#)**

```
public class PillRegimen
```

The domain object representing a pill regimen. This is the top level entity in this module.

**Constructors****PillRegimen**

```
public PillRegimen ()
```

Constructs a pill regimen without setting any fields.

**PillRegimen**

public **PillRegimen** (*String externalId*, *Set<Dosage> dosages*, *DailyScheduleDetails scheduleDetails*)

Constructs a pill regimen.

**Parameters**

- **externalId** – the external ID tied to this pill regimen, used for tying patients with the regimen.
- **dosages** – medicine dosages belonging to this pill regimen
- **scheduleDetails** – the schedule details for this pill regimen

**Methods****getDosage**

public *Dosage* **getDosage** (*Long dosageId*)

Retrieves the dosage with given ID belonging to this regimen.

**Parameters**

- **dosageId** – the id of the dosage to retrieve

**Returns** the matching dosage, or null if this regimen does not have a dosage with such an ID

**getDosages**

public *Set<Dosage>* **getDosages** ()

Gets medicine dosages belonging to this pill regimen. Each dosage can have multiple medicines.

**Returns** medicine dosages belonging to this pill regimen

**getExternalId**

public *String* **getExternalId** ()

**Returns** the external ID tied to this pill regimen, used for tying patients with the regimen

**getId**

public *Long* **getId** ()

**Returns** the unique identifier of the pill regimen

**getScheduleDetails**

public *DailyScheduleDetails* **getScheduleDetails** ()

**Returns** the schedule details for this pill regimen

**isFirstReminderFor**

public boolean **isFirstReminderFor** (*Dosage dosage*)

Checks if there were no reminders yet set for the given dosage.

**Parameters**

- **dosage** – the dosage to check

**Returns** true if no reminder was yet set for this dosage, false otherwise

**numberOfTimesPillRemindersSentFor**

public int **numberOfTimesPillRemindersSentFor** (*Dosage dosage*)

Returns the number of times a pill reminder was set for the given dosage. This is done using only a calculation based on the schedule for this regimen and the details from the dosage.

**Parameters**

- **dosage** – the dosage to check

**Returns** the number of time a pill reminder was sent for the dosage

**setDosages**

public void **setDosages** (*Set<Dosage> dosages*)

Sets medicine dosages belonging to this pill regimen. Each dosage can have multiple medicines.

**Parameters**

- **dosages** – medicine dosages belonging to this pill regimen

**setExternalId**

public void **setExternalId** (*String externalId*)

**Parameters**

- **externalId** – the external ID tied to this pill regimen, used for tying patients with the regimen

**setId**

public void **setId** (*Long id*)

**Parameters**

- **id** – the unique identifier of the pill regimen

**setScheduleDetails**

public void **setScheduleDetails** (*DailyScheduleDetails scheduleDetails*)

**Parameters**

- **scheduleDetails** – the schedule details for this pill regimen

**timesPillRemainderWillBeSent**

public int **timesPillRemainderWillBeSent** ()

Calculates the total number of times the pill reminder for this regimen will be sent, in case the patient does not provide an adherence response.

**Returns** the maximum number of reminders that will be sent for this regimen

**validate**

public void **validate** ()

Validates the dosages for this regimen. Will call *Dosage.validate()* on every medicine dosage in this regimen.

**Throws**

- **ValidationException** – if any of the dosages fails validation

## ValidationException

public class **ValidationException** extends [RuntimeException](#)  
Signals input validation errors.

### Constructors

#### ValidationException

public **ValidationException** ([String message](#))

## 19.14.80 org.motechproject.pillreminder.service

### PillRegimenJobScheduler

public class **PillRegimenJobScheduler**  
The scheduler facade used by the pill reminder module.

### Constructors

#### PillRegimenJobScheduler

public **PillRegimenJobScheduler** ([MotechSchedulerService schedulerService](#))

### Methods

#### getSchedulableDailyJob

protected [CronSchedulableJob](#) **getSchedulableDailyJob** ([PillRegimen pillRegimen](#), [Dosage dosage](#))

#### scheduleDailyJob

public void **scheduleDailyJob** ([PillRegimen pillRegimen](#))

Schedules a daily job for each dosage of the pill regimen. The jobs will fire events that will be handled by the pill reminder module.

##### Parameters

- **pillRegimen** – the regimen for which reminders should get scheduled

#### unsubscribeJobs

public void **unsubscribeJobs** ([PillRegimen regimen](#))

Unsubscribes all jobs scheduled for dosages from the given pill regimen.

##### Parameters

- **regimen** – the regimen to unsubscribe the jobs for

### PillReminderService

public interface **PillReminderService**

Pill reminder service supports creating/querying/deleting pill schedule as per prescription.

**See also:** [org.motechproject.pillreminder.contract.DailyPillRegimenRequest](#)

## Methods

### createNew

void **createNew** ([DailyPillRegimenRequest](#) *dailyPillRegimenRequest*)  
Subscribe to a pill reminder.

#### Parameters

- **dailyPillRegimenRequest** – the request for a daily pill regimen

See also: [org.motechproject.pillreminder.contract.DailyPillRegimenRequest](#)

### dosageStatusKnown

void **dosageStatusKnown** ([Long](#) *pillRegimenId*, [Long](#) *dosageId*, [LocalDate](#) *lastCapturedDate*)  
Update the dosage take status. Marking a dosage status as known, will stop the module from sending repeat reminders.

#### Parameters

- **pillRegimenId** – subscription id
- **dosageId** – Dosage id
- **lastCapturedDate** – Dosage confirmation captured date.

### getPillRegimen

[PillRegimenResponse](#) **getPillRegimen** ([String](#) *externalId*)  
Get pill regimen for given subscriber (externalId).

#### Parameters

- **externalId** – the external ID set from regimen

**Returns** Dosage details along with reminder config

See also: [org.motechproject.pillreminder.contract.PillRegimenResponse](#)

### remove

void **remove** ([String](#) *externalID*)  
Unsubscribe from pill reminder service.

#### Parameters

- **externalID** – Unique subscriber id.

### renew

void **renew** ([DailyPillRegimenRequest](#) *newDailyScheduleRequest*)  
Update the pill reminder subscription

#### Parameters

- **newDailyScheduleRequest** – the request used for updating the subscription.

See also: [org.motechproject.pillreminder.contract.DailyPillRegimenRequest](#)

## PillReminderServiceImpl

public class **PillReminderServiceImpl** implements **PillReminderService**  
Implementation of the **PillReminderService**. Uses **PillRegimenJobScheduler** for scheduling jobs and stores information in MDS using **PillRegimenDataService**.

### Constructors

#### PillReminderServiceImpl

public **PillReminderServiceImpl** (**PillRegimenDataService** *pillRegimenDataService*, **PillRegimenJobScheduler** *pillRegimenJobScheduler*)

### Methods

#### createNew

public void **createNew** (**DailyPillRegimenRequest** *dailyPillRegimenRequest*)

#### dosageStatusKnown

public void **dosageStatusKnown** (**Long** *pillRegimenId*, **Long** *dosageId*, **LocalDate** *lastCapturedDate*)

#### getPillRegimen

public **PillRegimenResponse** **getPillRegimen** (**String** *externalId*)

#### remove

public void **remove** (**String** *externalID*)

#### renew

public void **renew** (**DailyPillRegimenRequest** *dailyPillRegimenRequest*)

## 19.14.81 org.motechproject.scheduletracking.domain

### Alert

public class **Alert**

### Constructors

#### Alert

public **Alert** ()

#### Alert

public **Alert** (**Period** *offset*, **Period** *interval*, int *count*, int *index*, boolean *floating*)



## Methods

### **getCount**

public int **getCount** ()

### **getIndex**

public int **getIndex** ()

### **getInterval**

public [Period](#) **getInterval** ()

### **getMilestoneWindow**

public [MilestoneWindow](#) **getMilestoneWindow** ()

### **getOffset**

public [Period](#) **getOffset** ()

### **isFloating**

public boolean **isFloating** ()

### **setCount**

public void **setCount** (int *count*)

### **setFloating**

public void **setFloating** (boolean *floating*)

### **setIndex**

public void **setIndex** (int *index*)

### **setInterval**

public void **setInterval** ([Period](#) *interval*)

### **setMilestoneWindow**

public void **setMilestoneWindow** ([MilestoneWindow](#) *milestoneWindow*)

### **setOffset**

public void **setOffset** ([Period](#) *offset*)

## **AlertWindow**

public class **AlertWindow**

## Constructors

### AlertWindow

```
public AlertWindow (DateTime windowStart, DateTime windowEnd, DateTime enrolledOn, Time preferredAlertTime, Alert alert)
```

## Methods

### allPossibleAlerts

```
public List<DateTime> allPossibleAlerts ()
```

### numberOfAlertsToSchedule

```
public int numberOfAlertsToSchedule ()
```

### scheduledAlertStartDate

```
public DateTime scheduledAlertStartDate ()
```

## Enrollment

```
public class Enrollment
```

## Constructors

### Enrollment

```
public Enrollment ()
```

## Methods

### copyFrom

```
public Enrollment copyFrom (Enrollment enrollment)
```

### fulfillCurrentMilestone

```
public void fulfillCurrentMilestone (DateTime fulfillmentDateTime)
```

### getCurrentMilestoneName

```
public String getCurrentMilestoneName ()
```

### getCurrentMilestoneStartDate

```
public DateTime getCurrentMilestoneStartDate ()
```

### getCurrentWindowAsOf

```
public WindowName getCurrentWindowAsOf (DateTime asOf)
```

### getEndOfWindowForCurrentMilestone

```
public DateTime getEndOfWindowForCurrentMilestone (WindowName windowName)
```

**getEnrolledOn**

```
public DateTime getEnrolledOn ()
```

**getExternalId**

```
public String getExternalId ()
```

**getFulfillments**

```
public List<MilestoneFulfillment> getFulfillments ()
```

**getId**

```
public Long getId ()
```

**getLastFulfilledDate**

```
public DateTime getLastFulfilledDate ()
```

**getMetadata**

```
public Map<String, String> getMetadata ()
```

**getPreferredAlertTime**

```
public Time getPreferredAlertTime ()
```

**getSchedule**

```
public Schedule getSchedule ()
```

**getScheduleName**

```
public String getScheduleName ()
```

**getStartOfSchedule**

```
public DateTime getStartOfSchedule ()
```

**getStartOfWindowForCurrentMilestone**

```
public DateTime getStartOfWindowForCurrentMilestone (WindowName windowName)
```

**getStatus**

```
public EnrollmentStatus getStatus ()
```

**isActive**

```
public boolean isActive ()
```

**isCompleted**

```
public boolean isCompleted ()
```

**setCurrentMilestoneName**

```
public void setCurrentMilestoneName (String currentMilestoneName)
```

**setEnrolledOn**

public void **setEnrolledOn** (*DateTime enrolledOn*)

**setExternalId**

public void **setExternalId** (*String externalId*)

**setFulfillments**

public void **setFulfillments** (*List<MilestoneFulfillment> fulfillments*)

**setId**

public void **setId** (*Long id*)

**setMetadata**

public void **setMetadata** (*Map<String, String> metadata*)

**setPreferredAlertTime**

public void **setPreferredAlertTime** (*Time preferredAlertTime*)

**setSchedule**

public void **setSchedule** (*Schedule schedule*)

**setScheduleName**

public void **setScheduleName** (*String scheduleName*)

**setStartOfSchedule**

public void **setStartOfSchedule** (*DateTime startOfSchedule*)

**setStatus**

public void **setStatus** (*EnrollmentStatus status*)

**EnrollmentBuilder**

public class **EnrollmentBuilder**

The `EnrollmentBuilder` is a convenient class, that allows building `Enrollment` instances. Once all required fields are set, call `toEnrollment()` method to get an instance of `Enrollment` with the properties you have set.

**Constructors****EnrollmentBuilder**

public **EnrollmentBuilder** ()

**Methods****toEnrollment**

public `Enrollment` **toEnrollment** ()

**withCurrentMilestoneName**

```
public EnrollmentBuilder withCurrentMilestoneName (String currentMilestoneName)
```

**withEnrolledOn**

```
public EnrollmentBuilder withEnrolledOn (DateTime enrolledOn)
```

**withExternalId**

```
public EnrollmentBuilder withExternalId (String externalId)
```

**withFulfillments**

```
public EnrollmentBuilder withFulfillments (List<MilestoneFulfillment> fulfillments)
```

**withId**

```
public EnrollmentBuilder withId (Long id)
```

**withMetadata**

```
public EnrollmentBuilder withMetadata (Map<String, String> metadata)
```

**withPreferredAlertTime**

```
public EnrollmentBuilder withPreferredAlertTime (Time preferredAlertTime)
```

**withSchedule**

```
public EnrollmentBuilder withSchedule (Schedule schedule)
```

**withScheduleName**

```
public EnrollmentBuilder withScheduleName (String scheduleName)
```

**withStartOfSchedule**

```
public EnrollmentBuilder withStartOfSchedule (DateTime startOfSchedule)
```

**withStatus**

```
public EnrollmentBuilder withStatus (EnrollmentStatus status)
```

**EnrollmentStatus**

```
public enum EnrollmentStatus
```

**Enum Constants****ACTIVE**

```
public static final EnrollmentStatus ACTIVE
```

**COMPLETED**

```
public static final EnrollmentStatus COMPLETED
```

### DEFAULTED

public static final [EnrollmentStatus](#) **DEFAULTED**

### UNENROLLED

public static final [EnrollmentStatus](#) **UNENROLLED**

## Milestone

public class **Milestone**

### Constructors

#### Milestone

public **Milestone** ()

#### Milestone

public **Milestone** ([String](#) *name*, [Period](#) *earliest*, [Period](#) *due*, [Period](#) *late*, [Period](#) *max*)

### Methods

#### addAlert

public void **addAlert** ([WindowName](#) *windowName*, [Alert...](#) *alertList*)

#### getAlerts

public [List](#)<[Alert](#)> **getAlerts** ()

#### getData

public [Map](#)<[String](#), [String](#)> **getData** ()

#### getMaximumDuration

public [Period](#) **getMaximumDuration** ()

#### getMilestoneWindow

public [MilestoneWindow](#) **getMilestoneWindow** ([WindowName](#) *windowName*)

#### getMilestoneWindows

public [List](#)<[MilestoneWindow](#)> **getMilestoneWindows** ()

#### getName

public [String](#) **getName** ()

#### getSchedule

public [Schedule](#) **getSchedule** ()

**getWindowDuration**

```
public Period getWindowDuration (WindowName windowName)
```

**getWindowEnd**

```
public Period getWindowEnd (WindowName windowName)
```

**getWindowStart**

```
public Period getWindowStart (WindowName windowName)
```

**setData**

```
public void setData (Map<String, String> data)
```

**setSchedule**

```
public void setSchedule (Schedule schedule)
```

**windowElapsed**

```
public boolean windowElapsed (WindowName windowName, DateTime milestoneStartDateTime)
```

**MilestoneAlert**

```
public final class MilestoneAlert implements Serializable
```

**Constructors****MilestoneAlert**

```
public MilestoneAlert (String milestoneName, DateTime earliestDateTime, DateTime dueDateTime,  
                      DateTime lateDateTime, DateTime defaultmentDateTime)
```

**Methods****equals**

```
public boolean equals (Object o)
```

**fromMilestone**

```
public static MilestoneAlert fromMilestone (Milestone milestone, DateTime startOfMilestone)
```

**getDefaultmentDateTime**

```
public DateTime getDefaultmentDateTime ()
```

**getDueDateTime**

```
public DateTime getDueDateTime ()
```

**getEarliestDateTime**

```
public DateTime getEarliestDateTime ()
```

#### **getLateDateTime**

public **DateTime** **getLateDateTime** ()

#### **getMilestoneName**

public **String** **getMilestoneName** ()

#### **hashCode**

public int **hashCode** ()

#### **toString**

public **String** **toString** ()

### **MilestoneFulfillment**

public class **MilestoneFulfillment**

#### **Constructors**

##### **MilestoneFulfillment**

public **MilestoneFulfillment** ()

##### **MilestoneFulfillment**

public **MilestoneFulfillment** (*String milestoneName, DateTime fulfillmentDateTime*)

#### **Methods**

##### **getFulfillmentDateTime**

public **DateTime** **getFulfillmentDateTime** ()

##### **getMilestoneName**

public **String** **getMilestoneName** ()

##### **setFulfillmentDateTime**

public void **setFulfillmentDateTime** (*DateTime fulfillmentDateTime*)

##### **setMilestoneName**

public void **setMilestoneName** (*String milestoneName*)

### **MilestoneWindow**

public class **MilestoneWindow**

#### **Constructors**

##### **MilestoneWindow**

public **MilestoneWindow** ()



**MilestoneWindow**

```
public MilestoneWindow (WindowName name, Period period)
```

**Methods****addAlerts**

```
public void addAlerts (Alert... alertsList)
```

**getAlerts**

```
public List<Alert> getAlerts ()
```

**getMilestone**

```
public Milestone getMilestone ()
```

**getName**

```
public WindowName getName ()
```

**getPeriod**

```
public Period getPeriod ()
```

**setAlerts**

```
public void setAlerts (List<Alert> alerts)
```

**setMilestone**

```
public void setMilestone (Milestone milestone)
```

**setName**

```
public void setName (WindowName name)
```

**setPeriod**

```
public void setPeriod (Period period)
```

**Schedule**

```
public class Schedule
```

**Constructors****Schedule**

```
public Schedule ()
```

**Schedule**

```
public Schedule (String name)
```

## Methods

### **addMilestones**

public void **addMilestones** ([Milestone...](#) *milestonesList*)

### **equals**

public boolean **equals** ([Object](#) *o*)

### **getDuration**

public [Period](#) **getDuration** ()

### **getFirstMilestone**

public [Milestone](#) **getFirstMilestone** ()

### **getMilestone**

public [Milestone](#) **getMilestone** ([String](#) *milestoneName*)

### **getMilestones**

public [List](#)<[Milestone](#)> **getMilestones** ()

### **getName**

public [String](#) **getName** ()

### **getNextMilestoneName**

public [String](#) **getNextMilestoneName** ([String](#) *currentMilestoneName*)

### **hasExpiredSince**

public boolean **hasExpiredSince** ([DateTime](#) *referenceDateTime*, [String](#) *currentMilestoneStr*)

### **hashCode**

public int **hashCode** ()

### **isBasedOnAbsoluteWindows**

public boolean **isBasedOnAbsoluteWindows** ()

### **merge**

public [Schedule](#) **merge** ([Schedule](#) *schedule*)

### **setBasedOnAbsoluteWindows**

public void **setBasedOnAbsoluteWindows** (boolean *value*)

## **ScheduleFactory**

public class **ScheduleFactory**

## Fields

### EMPTY\_PERIOD

public static final [Period](#) **EMPTY\_PERIOD**

## Constructors

### ScheduleFactory

public **ScheduleFactory** ()

## Methods

### build

public [Schedule](#) **build** ([ScheduleRecord](#) *scheduleRecord*)

### build

public [Schedule](#) **build** ([ScheduleRecord](#) *scheduleRecord*, [Locale](#) *locale*)

## WindowName

public enum **WindowName**

## Enum Constants

### due

public static final [WindowName](#) **due**

### earliest

public static final [WindowName](#) **earliest**

### late

public static final [WindowName](#) **late**

### max

public static final [WindowName](#) **max**

## 19.14.82 org.motechproject.scheduletracking.domain.json

### AlertRecord

public class **AlertRecord**

## Methods

### count

public [String](#) **count** ()

### **interval**

public [List<String>](#) **interval** ()

### **isFloating**

public boolean **isFloating** ()

### **offset**

public [List<String>](#) **offset** ()

### **window**

public [String](#) **window** ()

## **MilestoneRecord**

public class **MilestoneRecord**

### **Methods**

#### **alerts**

public [List<AlertRecord>](#) **alerts** ()

#### **data**

public [Map<String, String>](#) **data** ()

#### **name**

public [String](#) **name** ()

#### **scheduleWindowsRecord**

public [ScheduleWindowsRecord](#) **scheduleWindowsRecord** ()

## **ScheduleRecord**

public class **ScheduleRecord**

### **Methods**

#### **equals**

public boolean **equals** ([Object o](#))

#### **hashCode**

public int **hashCode** ()

#### **isAbsoluteSchedule**

public boolean **isAbsoluteSchedule** ()

**milestoneRecords**

```
public List<MilestoneRecord> milestoneRecords ()
```

**name**

```
public String name ()
```

**ScheduleWindowsRecord**

```
public class ScheduleWindowsRecord
```

**Methods****due**

```
public List<String> due ()
```

**earliest**

```
public List<String> earliest ()
```

**late**

```
public List<String> late ()
```

**max**

```
public List<String> max ()
```

## 19.14.83 org.motechproject.scheduletracking.domain.search

**CompletedDuringCriterion**

```
public class CompletedDuringCriterion implements Criterion
```

**Constructors****CompletedDuringCriterion**

```
public CompletedDuringCriterion (DateTime start, DateTime end)
```

**Methods****fetch**

```
public List<Enrollment> fetch (AllEnrollments allEnrollments)
```

**filter**

```
public List<Enrollment> filter (List<Enrollment> enrollments)
```

**Criterion**

```
public interface Criterion
```

## Methods

### fetch

`List<Enrollment> fetch (AllEnrollments allEnrollments)`

### filter

`List<Enrollment> filter (List<Enrollment> enrollments)`

## EndOfWindowCriterion

public class **EndOfWindowCriterion** implements `Criterion`

## Constructors

### EndOfWindowCriterion

public **EndOfWindowCriterion** (`WindowName` windowName, `DateTime` start, `DateTime` end)

## Methods

### fetch

public `List<Enrollment> fetch` (`AllEnrollments` allEnrollments)

### filter

public `List<Enrollment> filter` (`List<Enrollment>` enrollments)

## ExternalIdCriterion

public class **ExternalIdCriterion** implements `Criterion`

## Constructors

### ExternalIdCriterion

public **ExternalIdCriterion** (`String` externalId)

## Methods

### fetch

public `List<Enrollment> fetch` (`AllEnrollments` allEnrollments)

### filter

public `List<Enrollment> filter` (`List<Enrollment>` enrollments)

## InWindowCriterion

public class **InWindowCriterion** implements `Criterion`

## Constructors

### InWindowCriterion

public **InWindowCriterion** ([List](#)<[WindowName](#)> *windowNames*)

## Methods

### fetch

public [List](#)<[Enrollment](#)> **fetch** ([AllEnrollments](#) *allEnrollments*)

### filter

public [List](#)<[Enrollment](#)> **filter** ([List](#)<[Enrollment](#)> *enrollments*)

### MetadataCriterion

public class **MetadataCriterion** implements [Criterion](#)

## Constructors

### MetadataCriterion

public **MetadataCriterion** ([String](#) *key*, [String](#) *value*)

## Methods

### fetch

public [List](#)<[Enrollment](#)> **fetch** ([AllEnrollments](#) *allEnrollments*)

### filter

public [List](#)<[Enrollment](#)> **filter** ([List](#)<[Enrollment](#)> *enrollments*)

### MilestoneCriterion

public class **MilestoneCriterion** implements [Criterion](#)

## Constructors

### MilestoneCriterion

public **MilestoneCriterion** ([String](#) *milestoneName*)

## Methods

### fetch

public [List](#)<[Enrollment](#)> **fetch** ([AllEnrollments](#) *allEnrollments*)

### filter

public [List](#)<[Enrollment](#)> **filter** ([List](#)<[Enrollment](#)> *enrollments*)

## ScheduleCriterion

public class **ScheduleCriterion** implements [Criterion](#)

### Constructors

#### ScheduleCriterion

public **ScheduleCriterion** ([String](#)... *scheduleNames*)

### Methods

#### fetch

public [List](#)<[Enrollment](#)> **fetch** ([AllEnrollments](#) *allEnrollments*)

#### filter

public [List](#)<[Enrollment](#)> **filter** ([List](#)<[Enrollment](#)> *enrollments*)

## StartOfWindowCriterion

public class **StartOfWindowCriterion** implements [Criterion](#)

### Constructors

#### StartOfWindowCriterion

public **StartOfWindowCriterion** ([WindowName](#) *windowName*, [DateTime](#) *start*, [DateTime](#) *end*)

### Methods

#### fetch

public [List](#)<[Enrollment](#)> **fetch** ([AllEnrollments](#) *allEnrollments*)

#### filter

public [List](#)<[Enrollment](#)> **filter** ([List](#)<[Enrollment](#)> *enrollments*)

## StatusCriterion

public class **StatusCriterion** implements [Criterion](#)

### Constructors

#### StatusCriterion

public **StatusCriterion** ([EnrollmentStatus](#) *status*)



## Methods

### fetch

```
public List<Enrollment> fetch (AllEnrollments allEnrollments)
```

### filter

```
public List<Enrollment> filter (List<Enrollment> enrollments)
```

## 19.14.84 org.motechproject.scheduletracking.events

### EnrolledUserEvent

```
public class EnrolledUserEvent
```

## Constructors

### EnrolledUserEvent

```
public EnrolledUserEvent (String externalId, String scheduleName, Time preferredAlertTime, DateTime referendeDateTime, DateTime enrollmentDateTime, String startingMilestoneName)
```

## Methods

### getEnrollmentDate

```
public LocalDate getEnrollmentDate ()
```

### getEnrollmentTime

```
public Time getEnrollmentTime ()
```

### getExternalId

```
public String getExternalId ()
```

### getPreferredAlertTime

```
public Time getPreferredAlertTime ()
```

### getReferenceDate

```
public LocalDate getReferenceDate ()
```

### getReferenceTime

```
public Time getReferenceTime ()
```

### getScheduleName

```
public String getScheduleName ()
```

### getStartingMilestoneName

```
public String getStartingMilestoneName ()
```

**toMotechEvent**

```
public MotechEvent toMotechEvent ()
```

**MilestoneDefaultedEvent**

```
public class MilestoneDefaultedEvent
```

**Constructors****MilestoneDefaultedEvent**

```
public MilestoneDefaultedEvent (MotechEvent event)
```

**MilestoneDefaultedEvent**

```
public MilestoneDefaultedEvent (Long enrollmentId, String jobId, String externalId)
```

**Methods****getEnrollmentId**

```
public Long getEnrollmentId ()
```

**getExternalId**

```
public String getExternalId ()
```

**getJobId**

```
public String getJobId ()
```

**toMotechEvent**

```
public MotechEvent toMotechEvent ()
```

**MilestoneEvent**

```
public class MilestoneEvent
```

This is the event which will be raised as per the alert configuration

**Constructors****MilestoneEvent**

```
public MilestoneEvent (String externalId, String scheduleName, MilestoneAlert milestoneAlert, String  
                        windowName, DateTime referenceDateTime, Map<String, String> milestone-  
                        Data)
```

Creates a MilestoneEvent

**Parameters**

- **externalId** –
- **scheduleName** –
- **milestoneAlert** –

- **windowName** –
- **referenceDateTime** –
- **milestoneData** –

### MilestoneEvent

public **MilestoneEvent** (MotechEvent *motechEvent*)

Creates a MilestoneEvent from an Enrollment by passing in an MotechEvent

#### Parameters

- **motechEvent** –

### MilestoneEvent

public **MilestoneEvent** ([Enrollment](#) *enrollment*, [MilestoneAlert](#) *milestoneAlert*, [MilestoneWindow](#) *milestoneWindow*)

Creates a MilestoneEvent from an Enrollment

#### Parameters

- **enrollment** –
- **milestoneAlert** –
- **milestoneWindow** –

## Methods

### getExternalId

public [String](#) **getExternalId** ()

Returns the External Id of the MilestoneEvent

**Returns** String

### getMilestoneAlert

public [MilestoneAlert](#) **getMilestoneAlert** ()

Returns the MilestoneAlert of the MilestoneEvent

**Returns** MilestoneAlert

### getMilestoneData

public [Map](#)<[String](#), [String](#)> **getMilestoneData** ()

Returns the Milestone Data of the MilestoneEvent

**Returns** Map

### getReferenceDateTime

public [DateTime](#) **getReferenceDateTime** ()

Returns the ReferenceDateTime of the MilestoneEvent

**Returns** DateTime

**getScheduleName**

```
public String getScheduleName ()
```

Returns the Schedule Name of the MilestoneEvent

**Returns** String

**getWindowName**

```
public String getWindowName ()
```

Returns the Window name of the MilestoneEvent

**Returns** String

**toMotechEvent**

```
public MotechEvent toMotechEvent ()
```

Creates an MotechEvent from a MilestoneEvent

**Returns** MotechEvent

**UnenrolledUserEvent**

```
public class UnenrolledUserEvent
```

**Constructors****UnenrolledUserEvent**

```
public UnenrolledUserEvent (String externalId, String scheduleName)
```

**UnenrolledUserEvent**

```
public UnenrolledUserEvent (Map<String, Object> parameters)
```

**Methods****getExternalId**

```
public String getExternalId ()
```

**getScheduleName**

```
public String getScheduleName ()
```

**toMotechEvent**

```
public MotechEvent toMotechEvent ()
```

**19.14.85 org.motechproject.scheduletracking.events.constants****EventDataKeys**

```
public final class EventDataKeys
```

Keys for accessing the Event Payload

## Fields

### **DEFAULTMENT\_DATE\_TIME**

public static final [String](#) **DEFAULTMENT\_DATE\_TIME**

### **DUE\_DATE\_TIME**

public static final [String](#) **DUE\_DATE\_TIME**

### **EARLIEST\_DATE\_TIME**

public static final [String](#) **EARLIEST\_DATE\_TIME**

### **ENROLLMENT\_DATE**

public static final [String](#) **ENROLLMENT\_DATE**

### **ENROLLMENT\_ID**

public static final [String](#) **ENROLLMENT\_ID**

### **ENROLLMENT\_TIME**

public static final [String](#) **ENROLLMENT\_TIME**

### **EXTERNAL\_ID**

public static final [String](#) **EXTERNAL\_ID**

### **LATE\_DATE\_TIME**

public static final [String](#) **LATE\_DATE\_TIME**

### **METADATA**

public static final [String](#) **METADATA**

### **MILESTONE\_DATA**

public static final [String](#) **MILESTONE\_DATA**

### **MILESTONE\_NAME**

public static final [String](#) **MILESTONE\_NAME**

### **PREFERRED\_ALERT\_TIME**

public static final [String](#) **PREFERRED\_ALERT\_TIME**

### **REFERENCE\_DATE**

public static final [String](#) **REFERENCE\_DATE**

### **REFERENCE\_TIME**

public static final [String](#) **REFERENCE\_TIME**

#### SCHEDULE\_NAME

public static final [String](#) SCHEDULE\_NAME

#### WINDOW\_NAME

public static final [String](#) WINDOW\_NAME

### EventSubjects

public final class **EventSubjects**

Event subjects published by the schedule tracking module

#### Fields

#### MILESTONE\_ALERT

public static final [String](#) MILESTONE\_ALERT

#### MILESTONE\_DEFAULTED

public static final [String](#) MILESTONE\_DEFAULTED

#### USER\_ENROLLED

public static final [String](#) USER\_ENROLLED

#### USER\_UNENROLLED

public static final [String](#) USER\_UNENROLLED

## 19.14.86 org.motechproject.scheduletracking.repository

### AllEnrollments

public class **AllEnrollments**

#### Methods

##### completedDuring

public [List](#)<[Enrollment](#)> completedDuring ([DateTime](#) start, [DateTime](#) end)

##### findByCurrentMilestone

public [List](#)<[Enrollment](#)> findByCurrentMilestone ([String](#) currentMilestone)

##### findByExternalId

public [List](#)<[Enrollment](#)> findByExternalId ([String](#) externalId)

##### findByMetadataProperty

public [List](#)<[Enrollment](#)> findByMetadataProperty ([String](#) property, [String](#) value)

**findBySchedule**

```
public List<Enrollment> findBySchedule (List<String> scheduleNames)
```

**findByStatus**

```
public List<Enrollment> findByStatus (EnrollmentStatus status)
```

**retrieveAll**

```
public List<Enrollment> retrieveAll ()
```

**MapQueryExecution**

public class **MapQueryExecution** implements QueryExecution<List<Enrollment>>

The `MapQueryExecution` class prepares a custom MDS query. The query should return only these enrollments, that have got a given key-value entry in their metadata.

See also: `Enrollment`

**Constructors****MapQueryExecution**

```
public MapQueryExecution (String key, String value)
```

**Methods****execute**

```
public List<Enrollment> execute (Query query, InstanceSecurityRestriction instanceSecurityRestriction)
```

**MilestoneRangeQueryExecution**

public class **MilestoneRangeQueryExecution** implements QueryExecution<List<Enrollment>>

The `MilestoneRangeQueryExecution` class prepares a custom MDS query. The query should return all enrollments that have been completed during given time range.

See also: `Enrollment`

**Constructors****MilestoneRangeQueryExecution**

```
public MilestoneRangeQueryExecution (DateTime start, DateTime end)
```

**Methods****execute**

```
public List<Enrollment> execute (Query query, InstanceSecurityRestriction instanceSecurityRestriction)
```

**TrackedSchedulesJsonReader**

```
public interface TrackedSchedulesJsonReader
```

## Methods

### getAllSchedules

`List<ScheduleRecord> getAllSchedules (String definitionsDirectoryName)`

### getSchedule

`ScheduleRecord getSchedule (String scheduleJson)`

## TrackedSchedulesJsonReaderImpl

public class **TrackedSchedulesJsonReaderImpl** implements `TrackedSchedulesJsonReader`

## Constructors

### TrackedSchedulesJsonReaderImpl

public **TrackedSchedulesJsonReaderImpl** ()

## Methods

### getAllSchedules

public `List<ScheduleRecord> getAllSchedules (String definitionsDirectoryName)`

### getSchedule

public `ScheduleRecord getSchedule (String scheduleJson)`

## 19.14.87 org.motechproject.scheduletracking.repository.dataservices

### EnrollmentDataService

public interface **EnrollmentDataService** extends `MotechDataService<Enrollment>`

Motech Data Service interface class for Enrollment entity. The implementation is provided by the Motech Data Services module.

See also: `Enrollment`

## Methods

### findByExternalId

`List<Enrollment> findByExternalId (String externalId)`

### findByExternalIdScheduleNameAndStatus

`Enrollment findByExternalIdScheduleNameAndStatus (String externalId, String scheduleName, EnrollmentStatus status)`

### findById

`Enrollment findById (Long id)`



**findByMilestoneName**

List<Enrollment> **findByMilestoneName** (String *milestoneName*)

**findByScheduleName**

List<Enrollment> **findByScheduleName** (Set<String> *scheduleName*)

**findByStatus**

List<Enrollment> **findByStatus** (EnrollmentStatus *enrollmentStatus*)

**ScheduleDataService**

public interface **ScheduleDataService** extends MotechDataService<Schedule>

Motech Data Service interface class for Schedule entity. The implementation is provided by the Motech Data Services module.

See also: [Schedule](#)

**Methods****findByName**

Schedule **findByName** (String *name*)

**19.14.88 org.motechproject.scheduletracking.service****EnrollmentRecord**

public class **EnrollmentRecord**

ingroup sts This is the record which will be returned when schedule tracking service is queried for enrollments  
It holds the details of an enrollment

**Constructors****EnrollmentRecord**

public **EnrollmentRecord** ()

**Methods****getCurrentMilestoneName**

public String **getCurrentMilestoneName** ()

This returns the current milestone name of an EnrollmentRecord

**Returns** String

**getEnrollmentDateTime**

public DateTime **getEnrollmentDateTime** ()

This returns the enrollment datetime of an EnrollmentRecord

**Returns** DateTime

#### **getExternalId**

public **String** **getExternalId** ()

This returns the External Id of an EnrollmentRecord

**Returns** String

#### **getMetadata**

public **Map**<**String**, **String**> **getMetadata** ()

This returns the metadata associated with the enrollment

**Returns** String

#### **getPreferredAlertTime**

public **Time** **getPreferredAlertTime** ()

This returns the preferred alert time of an EnrollmentRecord

**Returns** Time

#### **getReferenceDateTime**

public **DateTime** **getReferenceDateTime** ()

This returns the reference datetime of an EnrollmentRecord

**Returns** DateTime

#### **getScheduleName**

public **String** **getScheduleName** ()

This returns the Schedule Name of an EnrollmentRecord

**Returns** String

#### **getStartOfDueWindow**

public **DateTime** **getStartOfDueWindow** ()

This returns the due window start datetime of current milestone of an EnrollmentRecord

**Returns** DateTime

#### **getStartOfEarliestWindow**

public **DateTime** **getStartOfEarliestWindow** ()

This returns the earliest window start datetime of current milestone of an EnrollmentRecord

**Returns** DateTime

#### **getStartOfLateWindow**

public **DateTime** **getStartOfLateWindow** ()

This returns the late window start datetime of current milestone of an EnrollmentRecord

**Returns** DateTime

#### **getStartOfMaxWindow**

public **DateTime** **getStartOfMaxWindow** ()

This returns the max window start datetime of current milestone of an EnrollmentRecord

**Returns** DateTime

**getStatus**

```
public String getStatus ()
```

This returns the status of the enrollment

**Returns** String

**setCurrentMilestoneName**

```
public EnrollmentRecord setCurrentMilestoneName (String currentMilestoneName)
```

**setDueStart**

```
public EnrollmentRecord setDueStart (DateTime dueStart)
```

**setEarliestStart**

```
public EnrollmentRecord setEarliestStart (DateTime earliestStart)
```

**setEnrollmentDateTime**

```
public EnrollmentRecord setEnrollmentDateTime (DateTime enrollmentDateTime)
```

**setExternalId**

```
public EnrollmentRecord setExternalId (String externalId)
```

**setLateStart**

```
public EnrollmentRecord setLateStart (DateTime lateStart)
```

**setMaxStart**

```
public EnrollmentRecord setMaxStart (DateTime maxStart)
```

**setMetadata**

```
public EnrollmentRecord setMetadata (Map<String, String> metadata)
```

**setPreferredAlertTime**

```
public EnrollmentRecord setPreferredAlertTime (Time preferredAlertTime)
```

**setReferenceDateTime**

```
public EnrollmentRecord setReferenceDateTime (DateTime referenceDateTime)
```

**setScheduleName**

```
public EnrollmentRecord setScheduleName (String scheduleName)
```

**setStatus**

```
public EnrollmentRecord setStatus (String status)
```

**EnrollmentRequest**

```
public class EnrollmentRequest
```

ingroup sts This is the request document to create an enrollment

## Constructors

### EnrollmentRequest

public **EnrollmentRequest** ()

## Methods

### getEnrollmentDateTime

public **DateTime** **getEnrollmentDateTime** ()

This returns the enrollment date and time of the enrollment

**Returns** DateTime

### getExternalId

public **String** **getExternalId** ()

This returns the External Id of the Enrollment

**Returns** String

### getMetadata

public **Map<String, String>** **getMetadata** ()

This returns the Metadata key value map of the Enrollment

**Returns** String

### getPreferredAlertTime

public **Time** **getPreferredAlertTime** ()

This returns the preferred alert time of the Enrollment

**Returns** String

### getReferenceDate

public **LocalDate** **getReferenceDate** ()

This returns the reference date of the Enrollment

**Returns** String

### getReferenceDateTime

public **DateTime** **getReferenceDateTime** ()

This returns the reference date and time of the enrollment

**Returns** DateTime

### getReferenceTime

public **Time** **getReferenceTime** ()

This returns the reference time of the Enrollment

**Returns** String

**getScheduleName**

```
public String getScheduleName ()
```

This returns the Schedule Name of the Enrollment

**Returns** String

**getStartingMilestoneName**

```
public String getStartingMilestoneName ()
```

This returns the starting milestone name of the Enrollment

**Returns** String

**isStartingMilestoneSpecified**

```
public boolean isStartingMilestoneSpecified ()
```

This returns whether the starting milestone has been specified for the Enrollment

**Returns** true or false

**setEnrollmentDate**

```
public EnrollmentRequest setEnrollmentDate (LocalDate enrollmentDate)
```

This sets the enrollment date of the Enrollment. The date of enrollment. In case of enrollment into milestones other than first milestone, this becomes the start date of that milestone. Default value is today

**setEnrollmentTime**

```
public EnrollmentRequest setEnrollmentTime (Time enrollmentTime)
```

This sets the enrollment time of the Enrollment. The time of enrollment. In case of enrollment into milestones other than first milestone, this becomes the start time of that milestone. Default value is midnight

**setExternalId**

```
public EnrollmentRequest setExternalId (String externalId)
```

This sets the external id of the Enrollment

**setMetadata**

```
public EnrollmentRequest setMetadata (Map<String, String> metadata)
```

This sets the Metadata key value map of the Enrollment. This is list of string key value pairs associated with an enrollment, which can be used to store some additional information about the enrollment. Default value is empty list

**setPreferredAlertTime**

```
public EnrollmentRequest setPreferredAlertTime (Time preferredAlertTime)
```

This sets the preferred alert time of the Enrollment

**setReferenceDate**

```
public EnrollmentRequest setReferenceDate (LocalDate referenceDate)
```

This sets the reference date of the Enrollment. This is the start date of the schedule based on which all the window duration calculations are made. In case of enrollment into milestones other than first milestone, this date is not used. Default value is today

**setReferenceTime**

public [EnrollmentRequest](#) **setReferenceTime** (*Time referenceTime*)

This sets the reference time of the Enrollment. This is the start time of the schedule based on which all the window duration calculations are made. In case of enrollment into milestones other than first milestone, this time is not used. Default value is midnight

**setScheduleName**

public [EnrollmentRequest](#) **setScheduleName** (*String scheduleName*)

This sets the schedule of the Enrollment

**setStartingMilestoneName**

public [EnrollmentRequest](#) **setStartingMilestoneName** (*String startingMilestoneName*)

This sets the starting milestone of the Enrollment. Name of the milestone to enroll against Default value is first milestone name from the schedule

**EnrollmentService**

public interface **EnrollmentService**

**Methods****enroll**

Long **enroll** (*String externalId, String scheduleName, String startingMilestoneName, DateTime referenceDateTime, DateTime enrollmentDateTime, Time preferredAlertTime, Map<String, String> metadata*)

**fulfillCurrentMilestone**

void **fulfillCurrentMilestone** (*Enrollment enrollment, DateTime fulfillmentDateTime*)

**getAlertTimings**

[MilestoneAlerts](#) **getAlertTimings** (*String externalId, String scheduleName, String milestoneName, DateTime referenceDateTime, DateTime enrollmentDateTime, Time preferredAlertTime*)

**unenroll**

void **unenroll** (*Enrollment enrollment*)

**EnrollmentUpdater**

public enum **EnrollmentUpdater**

**Enum Constants****metadataUpdater**

public static final [EnrollmentUpdater](#) **metadataUpdater**

## EnrollmentsQuery

public class **EnrollmentsQuery**

ingroup sts This is the Query builder for retrieving enrollments Provides methods for different query criteria  
The order of criteria matters, as the first criterion is used to fetch the result from database and the other criterion are used to filter the results(in the specified sequence) fetched by the first criterion(in memory)

### Methods

#### completedDuring

public [EnrollmentsQuery](#) **completedDuring** ([DateTime](#) start, [DateTime](#) end)

This adds the criteria using which enrollments are filtered based on them being completed in the given date range

##### Parameters

- **start** –
- **end** – The start and end dates are inclusive

**Returns** returns the instance with completed during criteria added to the criteria list

#### currentlyInWindow

public [EnrollmentsQuery](#) **currentlyInWindow** ([WindowName](#)... windowNames)

This adds the criteria using which enrollments are filtered based on their current window(of current milestone) is in the given window name list

##### Parameters

- **windowNames** –

**Returns** returns the instance with currently in window criteria added to the criteria list

#### getCriteria

public [List](#)<[Criterion](#)> **getCriteria** ()

This gives all the criterion which are present in the built query

**Returns** [List](#)

#### getPrimaryCriterion

public [Criterion](#) **getPrimaryCriterion** ()

This gives the primary criterion in the built query, which is used to fetch the results from database

**Returns** [Criterion](#)

#### getSecondaryCriteria

public [List](#)<[Criterion](#)> **getSecondaryCriteria** ()

This gives all the criterion other than primary criterion in the built query, which are used to filter the results of the primary criterion

**Returns** [List](#)

**havingCurrentMilestone**

```
public EnrollmentsQuery havingCurrentMilestone (String milestoneName)
```

This adds the criteria using which enrollments are filtered based on their current milestone name

**Parameters**

- **milestoneName** –

**Returns** returns the instance with current milestone criteria added to the criteria list

**havingExternalId**

```
public EnrollmentsQuery havingExternalId (String externalId)
```

This adds the criteria using which enrollments are filtered based on external id

**Parameters**

- **externalId** –

**Returns** returns the instance with external id criteria added to the criteria list

**havingMetadata**

```
public EnrollmentsQuery havingMetadata (String key, String value)
```

This adds the criteria using which enrollments are filtered based on them having given metadata values in their metadata map

**Parameters**

- **key** –
- **value** –

**Returns** returns the instance with metadata criteria added to the criteria list

**havingSchedule**

```
public EnrollmentsQuery havingSchedule (String... scheduleNames)
```

This adds the criteria using which enrollments are filtered based on their schedule name

**Parameters**

- **scheduleNames** –
  - one or more string values indicating the schedule names

**Returns** returns the instance with schedule criteria added to the criteria list

**havingState**

```
public EnrollmentsQuery havingState (EnrollmentStatus enrollmentStatus)
```

This adds the criteria using which enrollments are filtered based on their status

**Parameters**

- **enrollmentStatus** –

**Returns** returns the instance with status criteria added to the criteria list



**havingWindowEndingDuring**

```
public EnrollmentsQuery havingWindowEndingDuring (WindowName windowName, DateTime start,
                                                    DateTime end)
```

This adds the criteria using which enrollments are filtered based on their given window end datetime falls during the given period

**Parameters**

- **windowName** –
- **start** –
- **end** – The start and end dates are inclusive

**Returns** returns the instance with window end criteria added to the criteria list

**havingWindowStartingDuring**

```
public EnrollmentsQuery havingWindowStartingDuring (WindowName windowName, DateTime
                                                    start, DateTime end)
```

This adds the criteria using which enrollments are filtered based on their given window start datetime falls during the given period

**Parameters**

- **windowName** –
- **start** –
- **end** – The start and end dates are inclusive

**Returns** returns the instance with window start criteria added to the criteria list

**MilestoneAlerts**

```
public class MilestoneAlerts
```

This contains the alert timings of all the windows of a milestone

**Constructors****MilestoneAlerts**

```
public MilestoneAlerts ()
```

Constructor to create MilestoneAlerts

**Methods****getAlertTimings**

```
public Map<String, List<DateTime>> getAlertTimings ()
```

Gives the alert timings for all the windows. Note : To get the alert timings of a particular window, use the corresponding getWindowAlertTimings method

**Returns** a map of window names to the list of their alert timings

**getDueWindowAlertTimings**

```
public List<DateTime> getDueWindowAlertTimings ()
```

Gives the alert timings of due window of the milestone

**Returns** List : represents the alert timings of due window

**getEarliestWindowAlertTimings**

```
public List<DateTime> getEarliestWindowAlertTimings ()
```

Gives the alert timings of earliest window of the milestone

**Returns** List : represents the alert timings of earliest window

**getLateWindowAlertTimings**

```
public List<DateTime> getLateWindowAlertTimings ()
```

Gives the alert timings of late window of the milestone

**Returns** List : represents the alert timings of late window

**getMaxWindowAlertTimings**

```
public List<DateTime> getMaxWindowAlertTimings ()
```

Gives the alert timings of max window of the milestone

**Returns** List : represents the alert timings of max window

**ScheduleTrackingService**

```
public interface ScheduleTrackingService
```

ingroup sts Schedule Tracking Service interface Provides methods to enroll an external id into a schedule, fulfill milestones in a schedule and uneroll an external id from a schedule. It also provides querying functionality on Enrollments using various search criteria

**Methods****add**

```
void add (String schedule.Json)
```

Saves the given schedule in database.

**Parameters**

- **schedule.Json** – : in JSON format

**enroll**

```
Long enroll (EnrollmentRequest enrollmentRequest)
```

Enrolls a user with external id into a schedule using the details in the EnrollmentRequest and schedules alerts. If the user has already enrolled for the same schedule which is currently active, then we update the existing details and reschedule alerts

**Parameters**

- **enrollmentRequest** –

**Returns** enrollmentId string

**fulfillCurrentMilestone**

```
void fulfillCurrentMilestone (String externalId, String scheduleName, LocalDate fulfillmentDate,  
Time fulfillmentTime)
```

Fulfills the current milestone of the enrollment(with fulfillmentDate and time as the the given date and time) which belongs to the given externalId and schedule name \*

**Parameters**

- `externalId` –
- `scheduleName` –
- `fulfillmentDate` –
- `fulfillmentTime` –

#### **fulfillCurrentMilestone**

void **fulfillCurrentMilestone** (*String externalId, String scheduleName, LocalDate fulfillmentDate*)

Fulfills the current milestone of the enrollment(with fulfillmentDate and time as midnight) which belongs to the given externalId and schedule name \*

##### **Parameters**

- `externalId` –
- `scheduleName` –
- `fulfillmentDate` –

#### **getAlertTimings**

*MilestoneAlerts* **getAlertTimings** (*EnrollmentRequest enrollmentRequest*)

Gives the alert timings of all the windows in the milestone without actually scheduling the alert jobs

##### **Parameters**

- `enrollmentRequest` –

**Returns** *MilestoneAlerts* : contains the alert timings for all the windows of the milestone

#### **getAllSchedules**

*List<Schedule>* **getAllSchedules** ()

#### **getEnrollment**

*EnrollmentRecord* **getEnrollment** (*String externalId, String scheduleName*)

Returns the EnrollmentRecord corresponds to the given externalId and schedule name

##### **Parameters**

- `externalId` –
- `scheduleName` –

#### **getScheduleByName**

*Schedule* **getScheduleByName** (*String scheduleName*)

#### **remove**

void **remove** (*String scheduleName*)

Removes the specified schedule from the database.

##### **Parameters**

- `scheduleName` – : name of the schedule to remove

**search**

List<EnrollmentRecord> **search** (EnrollmentsQuery *query*)

Searches and returns the Enrollment Records as per the criteria in the given EnrollmentsQuery

**Parameters**

- **query** –

**Returns** List

**searchWithWindowDates**

List<EnrollmentRecord> **searchWithWindowDates** (EnrollmentsQuery *query*)

Searches and returns the Enrollment Records(with all window start dates populated in them) as per the criteria in the given EnrollmentsQuery

**Parameters**

- **query** –

**Returns** List

**unenroll**

void **unenroll** (String *externalId*, List<String> *scheduleNames*)

Unenrolls / Removes all the scheduled jobs of enrollments which belongs to the given externalId and schedule names

**Parameters**

- **externalId** –
- **scheduleNames** –

**updateEnrollment**

void **updateEnrollment** (String *externalId*, String *scheduleName*, UpdateCriteria *updateCriteria*)

Updates an active Enrollment which has the given external id and schedule name

**Parameters**

- **updateCriteria** – states the fields to be updated in the enrollment

**ScheduletrackingTasksActionFacade**

public interface **ScheduletrackingTasksActionFacade**

**Methods****enroll**

void **enroll** (String *externalId*, String *scheduleName*, String *preferredAlertTime*, DateTime *referenceDate*, String *referenceTime*, DateTime *enrolmentDate*, String *enrollmentTime*, String *startingMilestoneName*)

**unenroll**

void **unenroll** (String *externalId*, String *scheduleName*)

## 19.14.89 org.motechproject.scheduletracking.service.impl

### EndOfMilestoneListener

public class **EndOfMilestoneListener**

#### Constructors

##### EndOfMilestoneListener

public **EndOfMilestoneListener** ([EnrollmentDataService](#) *enrollmentDataService*)

#### Methods

##### handle

public void **handle** ([MotechEvent](#) *motechEvent*)

### EnrollmentAlertService

public class **EnrollmentAlertService**

#### Constructors

##### EnrollmentAlertService

public **EnrollmentAlertService** ([MotechSchedulerService](#) *schedulerService*, [EventRelay](#) *eventRelay*)

#### Methods

##### getAlertTimings

public [MilestoneAlerts](#) **getAlertTimings** ([Enrollment](#) *enrollment*)

##### scheduleAlertsForCurrentMilestone

public void **scheduleAlertsForCurrentMilestone** ([Enrollment](#) *enrollment*)

##### unscheduleAllAlerts

public void **unscheduleAllAlerts** ([Enrollment](#) *enrollment*)

### EnrollmentDefaultmentService

public class **EnrollmentDefaultmentService**

#### Constructors

##### EnrollmentDefaultmentService

public **EnrollmentDefaultmentService** ([MotechSchedulerService](#) *schedulerService*)

## Methods

### **scheduleJobToCaptureDefaultment**

public void **scheduleJobToCaptureDefaultment** ([Enrollment](#) *enrollment*)

### **unscheduleMilestoneDefaultedJob**

public void **unscheduleMilestoneDefaultedJob** ([Enrollment](#) *enrollment*)

## **EnrollmentRecordMapper**

public class **EnrollmentRecordMapper**

## Methods

### **map**

public [EnrollmentRecord](#) **map** ([Enrollment](#) *enrollment*)

### **mapWithDates**

public [EnrollmentRecord](#) **mapWithDates** ([Enrollment](#) *enrollment*)

## **EnrollmentServiceImpl**

public class **EnrollmentServiceImpl** implements [EnrollmentService](#)

## Constructors

### **EnrollmentServiceImpl**

public **EnrollmentServiceImpl** ([ScheduleDataService](#) *scheduleDataService*, [EnrollmentDataService](#) *enrollmentDataService*, [EnrollmentAlertService](#) *enrollmentAlertService*, [EnrollmentDefaultmentService](#) *enrollmentDefaultmentService*, [EventRelay](#) *eventRelay*)

## Methods

### **enroll**

public [Long](#) **enroll** ([String](#) *externalId*, [String](#) *scheduleName*, [String](#) *startingMilestoneName*, [DateTime](#) *referenceDateTime*, [DateTime](#) *enrollmentDateTime*, [Time](#) *preferredAlertTime*, [Map](#)<[String](#), [String](#)> *metadata*)

### **fulfillCurrentMilestone**

public void **fulfillCurrentMilestone** ([Enrollment](#) *enrollment*, [DateTime](#) *fulfillmentDateTime*)

### **getAlertTimings**

public [MilestoneAlerts](#) **getAlertTimings** ([String](#) *externalId*, [String](#) *scheduleName*, [String](#) *milestoneName*, [DateTime](#) *referenceDateTime*, [DateTime](#) *enrollmentDateTime*, [Time](#) *preferredAlertTime*)

**unenroll**

public void **unenroll** (*Enrollment enrollment*)

**EnrollmentsQueryService**

public class **EnrollmentsQueryService**

**Constructors****EnrollmentsQueryService**

public **EnrollmentsQueryService** (*AllEnrollments allEnrollments*)

**Methods****search**

public *List<Enrollment>* **search** (*EnrollmentsQuery query*)

**ScheduleTrackingServiceImpl**

public class **ScheduleTrackingServiceImpl** implements *ScheduleTrackingService*  
Implementation of *ScheduleTrackingService*

**Constructors****ScheduleTrackingServiceImpl**

public **ScheduleTrackingServiceImpl** (*EnrollmentsQueryService enrollmentsQueryService*, *ScheduleDataService scheduleDataService*, *EnrollmentService enrollmentService*, *EnrollmentRecordMapper enrollmentRecordMapper*, *EnrollmentDataService enrollmentDataService*, *SettingsFacade scheduleTrackingSettings*)

**Methods****add**

public void **add** (*String scheduleJson*)

**enroll**

public *Long* **enroll** (*EnrollmentRequest enrollmentRequest*)

**fulfillCurrentMilestone**

public void **fulfillCurrentMilestone** (*String externalId*, *String scheduleName*, *LocalDate fulfillmentDate*, *Time fulfillmentTime*)

**fulfillCurrentMilestone**

public void **fulfillCurrentMilestone** (*String externalId*, *String scheduleName*, *LocalDate fulfillmentDate*)

**getAlertTimings**

public [MilestoneAlerts](#) **getAlertTimings** ([EnrollmentRequest](#) *enrollmentRequest*)

**getAllSchedules**

public [List](#)<[Schedule](#)> **getAllSchedules** ()

**getEnrollment**

public [EnrollmentRecord](#) **getEnrollment** ([String](#) *externalId*, [String](#) *scheduleName*)

**getScheduleByName**

public [Schedule](#) **getScheduleByName** ([String](#) *scheduleName*)

**remove**

public void **remove** ([String](#) *scheduleName*)

**search**

public [List](#)<[EnrollmentRecord](#)> **search** ([EnrollmentsQuery](#) *query*)

**searchWithWindowDates**

public [List](#)<[EnrollmentRecord](#)> **searchWithWindowDates** ([EnrollmentsQuery](#) *query*)

**unenroll**

public void **unenroll** ([String](#) *externalId*, [List](#)<[String](#)> *scheduleNames*)

**updateEnrollment**

public void **updateEnrollment** ([String](#) *externalId*, [String](#) *scheduleName*, [UpdateCriteria](#) *updateCriteria*)

**ScheduletrackingTasksActionFacadeImpl**

public class **ScheduletrackingTasksActionFacadeImpl** implements [ScheduletrackingTasksActionFacade](#)

**Methods****enroll**

public void **enroll** ([String](#) *externalId*, [String](#) *scheduleName*, [String](#) *preferredAlertTime*, [DateTime](#) *referenceDate*, [String](#) *referenceTime*, [DateTime](#) *enrolmentDate*, [String](#) *enrollmentTime*, [String](#) *startingMilestoneName*)

**unenroll**

public void **unenroll** ([String](#) *externalId*, [String](#) *scheduleName*)



## 19.14.90 org.motechproject.sms.audit

### DeliveryStatus

public enum **DeliveryStatus**  
Cross-provider delivery statuses

#### Enum Constants

##### ABORTED

public static final [DeliveryStatus](#) **ABORTED**  
The maximum number of retries has occurred, so we've given up trying

##### DELIVERY\_CONFIRMED

public static final [DeliveryStatus](#) **DELIVERY\_CONFIRMED**  
Received confirmation from the provider that the message was delivered

##### DISPATCHED

public static final [DeliveryStatus](#) **DISPATCHED**  
SMS was successfully sent to the provider

##### FAILURE\_CONFIRMED

public static final [DeliveryStatus](#) **FAILURE\_CONFIRMED**  
Received confirmation from the provider that the message was not delivered

##### PENDING

public static final [DeliveryStatus](#) **PENDING**  
We just received a message and will either try to send it directly or schedule it for later delivery

##### RECEIVED

public static final [DeliveryStatus](#) **RECEIVED**  
Incoming SMS

##### RETRYING

public static final [DeliveryStatus](#) **RETRYING**  
There was a problem delivering a message to the provider, we're retrying

##### SCHEDULED

public static final [DeliveryStatus](#) **SCHEDULED**  
This message was added to the MOTECH schedule

### SmsAuditService

public interface **SmsAuditService**  
Reading and writing to the SMS audit log

## Methods

### countAllSmsRecords

long **countAllSmsRecords** ([SmsRecordSearchCriteria](#) *criteria*)

Returns the count of `SmsRecords` entries matching the specified search criteria.

**Returns** the count of sms records matching the provided criteria

### findAllSmsRecords

[List<SmsRecord>](#) **findAllSmsRecords** ()

Finds and returns all `SmsRecord` entries in the sms log.

**Returns** all sms records in the sms log

### findAllSmsRecords

[SmsRecords](#) **findAllSmsRecords** ([SmsRecordSearchCriteria](#) *criteria*)

Finds and returns all `SmsRecords` entries matching the specified search criteria.

**Returns** all sms records matching the provided criteria

## SmsAuditServiceImpl

public class **SmsAuditServiceImpl** implements [SmsAuditService](#)  
Service that does the reading and writing to the SMS audit log.

## Constructors

### SmsAuditServiceImpl

public **SmsAuditServiceImpl** ([SmsRecordsDataService](#) *smsRecordsDataService*)

## Methods

### countAllSmsRecords

public long **countAllSmsRecords** ([SmsRecordSearchCriteria](#) *criteria*)

### findAllSmsRecords

public [List<SmsRecord>](#) **findAllSmsRecords** ()

### findAllSmsRecords

public [SmsRecords](#) **findAllSmsRecords** ([SmsRecordSearchCriteria](#) *criteria*)

## SmsDirection

public enum **SmsDirection**

Incoming (MO: Mobile Originated) or outgoing (MT: Mobile Terminated) SMS

## Enum Constants

### INBOUND

public static final [SmsDirection](#) INBOUND

### OUTBOUND

public static final [SmsDirection](#) OUTBOUND

## SmsLoggingRecord

public class **SmsLoggingRecord**

Represents one record in the logging UI

## Constructors

### SmsLoggingRecord

public **SmsLoggingRecord** ([SmsRecord](#) record)

Constructs an instance of SmsLoggingRecord from an [SmsRecord](#). In other words it converts the domain object to the view object.

#### Parameters

- **record** – the domain representation of the record

## Methods

### getConfig

public [String](#) **getConfig** ()

**Returns** the name of the configuration this SMS should be associated with

### getDeliveryStatus

public [String](#) **getDeliveryStatus** ()

**Returns** the delivery status for this SMS

### getErrorMessage

public [String](#) **getErrorMessage** ()

**Returns** the error message from the provider

### getMessageContent

public [String](#) **getMessageContent** ()

**Returns** the content of the SMS message

### getMotechId

public [String](#) **getMotechId** ()

**Returns** the ID used to identify the SMS in Motech

#### **getPhoneNumber**

public **String** **getPhoneNumber** ()

**Returns** the phone number to which the SMS was sent, or from which it was received

#### **getProviderId**

public **String** **getProviderId** ()

**Returns** the provider generated ID for the SMS

#### **getProviderStatus**

public **String** **getProviderStatus** ()

#### **getSmsDirection**

public **String** **getSmsDirection** ()

**Returns** the direction of the SMS, inbound or outbound

#### **getTimestamp**

public **String** **getTimestamp** ()

**Returns** the timestamp for the SMS

#### **setConfig**

public void **setConfig** (**String** *config*)

##### **Parameters**

- **config** – the name of the configuration this SMS should be associated with

#### **setDeliveryStatus**

public void **setDeliveryStatus** (**String** *deliveryStatus*)

##### **Parameters**

- **deliveryStatus** – the delivery status for this SMS

#### **setErrorMessage**

public void **setErrorMessage** (**String** *errorMessage*)

##### **Parameters**

- **errorMessage** – the error message from the provider

#### **setMessageContent**

public void **setMessageContent** (**String** *messageContent*)

##### **Parameters**

- **messageContent** – the content of the SMS message

**setMotechId**

public void **setMotechId** ([String](#) *motechId*)

**Parameters**

- **motechId** – the ID used to identify the SMS in Motech

**setPhoneNumber**

public void **setPhoneNumber** ([String](#) *phoneNumber*)

**Parameters**

- **phoneNumber** – the phone number to which the SMS was sent, or from which it was received

**setProviderId**

public void **setProviderId** ([String](#) *providerId*)

**Parameters**

- **providerId** – the provider generated ID for the SMS

**setProviderStatus**

public void **setProviderStatus** ([String](#) *providerStatus*)

**setSmsDirection**

public void **setSmsDirection** ([String](#) *smsDirection*)

**Parameters**

- **smsDirection** – the direction of the SMS, inbound or outbound

**setTimestamp**

public void **setTimestamp** ([String](#) *timestamp*)

**Parameters**

- **timestamp** – the timestamp for the SMS

**toString**

public [String](#) **toString** ()

**SmsLoggingRecords**

public class **SmsLoggingRecords** implements [Serializable](#)  
SMS record collection for the logging UI

**Constructors****SmsLoggingRecords**

public **SmsLoggingRecords** ([Integer](#) *page*, [Integer](#) *rows*, [Long](#) *totalRecords*, [SmsRecords](#) *smsRecords*)  
Constructs an sms logging view for the jq grid.

**Parameters**

- **page** – the page number
- **rows** – the number of rows per page
- **totalRecords** – the total number of records
- **smsRecords** – the data to display in the grid

## Methods

### **getPage**

public **Integer** **getPage** ()

**Returns** the page number

### **getRecords**

public **Long** **getRecords** ()

**Returns** the total number of records

### **getRows**

public **List**<**SmsLoggingRecord**> **getRows** ()

**Returns** the data display in the grid

### **getTotal**

public **Integer** **getTotal** ()

**Returns** the number of rows per page

### **toString**

public **String** **toString** ()

## **SmsRecord**

public class **SmsRecord**

SMS audit record for the database.

## Constructors

### **SmsRecord**

public **SmsRecord** ()

Constructs a new instance of a record without filling out any fields.

### **SmsRecord**

public **SmsRecord** (**String** *config*, **SmsDirection** *smsDirection*, **String** *number*, **String** *message*, **DateTime** *timestamp*, **DeliveryStatus** *deliveryStatus*, **String** *providerId*, **String** *motechId*, **String** *providerId*, **String** *errorMessage*)

Constructs a new instance using the provided data.

#### **Parameters**

- **config** – the name of the configuration to which this SMS should be associated with

- **smsDirection** – the direction of the SMS - either inbound or outbound
- **number** – the phone number to which this SMS was sent, or which it was received
- **message** – the content of the SMS message
- **timestamp** – the timestamp for this SMS
- **deliveryStatus** – the delivery status for this SMS
- **providerStatus** –
- **motechId** – the ID by which MOTECH identifies this SMS.
- **providerId** – the ID generated by the provider for this SMS
- **errorMessage** – the error message for this SMS, if applicable

## Methods

### **equals**

public boolean **equals** (*Object obj*)

### **getConfig**

public *String* **getConfig** ()

**Returns** the name of the configuration to which this SMS should be associated with

### **getDeliveryStatus**

public *DeliveryStatus* **getDeliveryStatus** ()

**Returns** the delivery status for this SMS

### **getErrorMessage**

public *String* **getErrorMessage** ()

**Returns** the error message for this SMS

### **getMessageContent**

public *String* **getMessageContent** ()

**Returns** the content of the SMS message

### **getMotechId**

public *String* **getMotechId** ()

**Returns** the ID by which MOTECH identifies this SMS.

### **getPhoneNumber**

public *String* **getPhoneNumber** ()

**Returns** the phone number to which this SMS was sent, or which it was received

#### **getProviderId**

public [String](#) **getProviderId** ()

**Returns** the ID generated by the provider for this SMS

#### **getProviderStatus**

public [String](#) **getProviderStatus** ()

#### **getSmsDirection**

public [SmsDirection](#) **getSmsDirection** ()

**Returns** the direction of the SMS - either inbound or outbound

#### **getTimestamp**

public [DateTime](#) **getTimestamp** ()

**Returns** the timestamp for this SMS

#### **hashCode**

public int **hashCode** ()

#### **setConfig**

public void **setConfig** ([String](#) *config*)

##### **Parameters**

- **config** – the name of the configuration to which this SMS should be associated with

#### **setDeliveryStatus**

public void **setDeliveryStatus** ([DeliveryStatus](#) *deliveryStatus*)

##### **Parameters**

- **deliveryStatus** – the delivery status for this SMS

#### **setErrorMessage**

public void **setErrorMessage** ([String](#) *errorMessage*)

##### **Parameters**

- **errorMessage** – the error message for this SMS

#### **setMessageContent**

public void **setMessageContent** ([String](#) *messageContent*)

##### **Parameters**

- **messageContent** – the content of the SMS message

#### **setMotechId**

public void **setMotechId** ([String](#) *motechId*)

##### **Parameters**

- **motechId** – the ID by which MOTECH identifies this SMS.



**setPhoneNumber**

public void **setPhoneNumber** (*String phoneNumber*)

**Parameters**

- **phoneNumber** – the phone number to which this SMS was sent, or which it was received

**setProviderId**

public void **setProviderId** (*String providerId*)

**Parameters**

- **providerId** – the ID generated by the provider for this SMS

**setProviderStatus**

public void **setProviderStatus** (*String providerStatus*)

**setSmsDirection**

public void **setSmsDirection** (*SmsDirection smsDirection*)

**Parameters**

- **smsDirection** – the direction of the SMS - either inbound or outbound

**setTimestamp**

public void **setTimestamp** (*DateTime timestamp*)

**Parameters**

- **timestamp** – the timestamp for this SMS

**toString**

public *String* **toString** ()

**SmsRecordSearchCriteria**

public class **SmsRecordSearchCriteria**

Helper used to generate a database lookup from log filter UI

**Methods****getConfig**

public *String* **getConfig** ()

**Returns** the configuration name which with the SMS should be associated with

**getDeliveryStatuses**

public *Set<String>* **getDeliveryStatuses** ()

**Returns** the set of expected delivery statuses for SMS messages

#### **getErrorMessage**

public [String](#) **getErrorMessage** ()

**Returns** the error message for the SMS

#### **getMessageContent**

public [String](#) **getMessageContent** ()

**Returns** the content of the sms message

#### **getMotechId**

public [String](#) **getMotechId** ()

**Returns** the ID by which MOTECH identifies the SMS

#### **getPhoneNumber**

public [String](#) **getPhoneNumber** ()

**Returns** the phone number which received or sent the sms

#### **getProviderId**

public [String](#) **getProviderId** ()

**Returns** the ID by which the provider identifies the SMS

#### **getProviderStatus**

public [String](#) **getProviderStatus** ()

#### **getQueryParams**

public QueryParams **getQueryParams** ()

**Returns** the query params that will control the result set for the lookup

#### **getSmsDirections**

public [Set](#)<[String](#)> **getSmsDirections** ()

**Returns** the set of expected SMS directions (inbound, outbound)

#### **getTimestampRange**

public [Range](#)<[DateTime](#)> **getTimestampRange** ()

**Returns** the timestamp range into which SNS messages should fall

#### **toString**

public [String](#) **toString** ()

**withConfig**

```
public SmsRecordSearchCriteria withConfig (String config)
```

Sets the configuration name part of the search.

**Parameters**

- **config** – the config name associated with the SMS

**Returns** this instance of the search criteria

**withDeliverystatuses**

```
public SmsRecordSearchCriteria withDeliverystatuses (Set<DeliveryStatus> deliveryStatuses)
```

Sets the delivery statuses the SMS messages retrieved by this search must match.

**Parameters**

- **deliveryStatuses** – the set of delivery status that will be taken into consideration when executing the query

**Returns** this instance of the search criteria

**withErrorMessage**

```
public SmsRecordSearchCriteria withErrorMessage (String errorMessage)
```

Sets the error message part of this search query.

**Parameters**

- **errorMessage** – the error message for the SMS

**Returns** this instance of the search criteria

**withMessageContent**

```
public SmsRecordSearchCriteria withMessageContent (String messageContent)
```

Sets the message content part of this search query.

**Parameters**

- **messageContent** – the content of the SMS

**Returns** this instance of the search criteria

**withMotechId**

```
public SmsRecordSearchCriteria withMotechId (String motechId)
```

Sets the MOTECH ID part of the search query.

**Parameters**

- **motechId** – the ID used by MOTECH to identify this SMS message

**Returns** this instance of the search criteria

**withPhoneNumber**

```
public SmsRecordSearchCriteria withPhoneNumber (String phoneNumber)
```

Sets the phone number part of the search.

**Parameters**

- **phoneNumber** – the phone number which received or sent the SMS

**Returns** this instance of the search criteria

**withProviderId**

public [SmsRecordSearchCriteria](#) **withProviderId** ([String](#) *providerId*)

Sets the provider ID part of the search query.

**Parameters**

- **providerId** – the ID used by the provider to identify this SMS message

**Returns** this instance of the search criteria

**withProviderStatus**

public [SmsRecordSearchCriteria](#) **withProviderStatus** ([String](#) *providerStatus*)

**withQueryParams**

public [SmsRecordSearchCriteria](#) **withQueryParams** ([QueryParams](#) *queryParam*)

Sets the query params that will be used when performing the lookup. This params will control the ordering and size of the result set.

**Parameters**

- **queryParam** – the params that will be passed to the lookup

**Returns** this instance of the search criteria

**withSmsDirections**

public [SmsRecordSearchCriteria](#) **withSmsDirections** ([Set](#)<[SmsDirection](#)> *smsDirections*)

Sets the sms directions which should be included in the query.

**Parameters**

- **smsDirections** – the set of directions (inbound, outbound)

**Returns** this instance of the search criteria

**withTimestamp**

public [SmsRecordSearchCriteria](#) **withTimestamp** ([DateTime](#) *timestamp*)

Sets the expected timestamp for the messages retrieved by this query.

**Parameters**

- **timestamp** – the timestamp of the messages

**Returns** this instance of the search criteria

**withTimestampRange**

public [SmsRecordSearchCriteria](#) **withTimestampRange** ([Range](#)<[DateTime](#)> *timestampRange*)

Sets the expected timestamp range into which the messages retrieved by this query must fall into.

**Parameters**

- **timestampRange** – the timestamp range into which the messages must fall into

**Returns** this instance of the search criteria

**SmsRecords**

public class **SmsRecords** implements [Serializable](#)

SMS audit records from the database. Represents a subset of all records.

## Constructors

### SmsRecords

public **SmsRecords** ()

Constructs an instance with no records and a total count of zero.

### SmsRecords

public **SmsRecords** (int *count*, List<SmsRecord> *records*)

Constructs an instance from the given records and count.

#### Parameters

- **count** – the total count of records in this context
- **records** – the subset of records

## Methods

### getCount

public int **getCount** ()

**Returns** the total number of SMS records in the current context

### getRecords

public List<SmsRecord> **getRecords** ()

**Returns** the actual subset of records

### setCount

public void **setCount** (int *count*)

#### Parameters

- **count** – the total number of SMS records in the current context

### setRecords

public void **setRecords** (List<SmsRecord> *records*)

#### Parameters

- **records** – the actual subset of records

## SmsRecordsDataService

public interface **SmsRecordsDataService** extends MotechDataService<SmsRecord>

Used to query and save audit records in the database. This is a service interface for which MDS will generate the implementation at runtime.

## Methods

### countFindByCriteria

long **countFindByCriteria** (*String config*, *Set<SmsDirection> directions*, *String phoneNumber*, *String messageContent*, *Range<DateTime> timestamp*, *Set<DeliveryStatus> deliveryStatuses*, *String providerStatus*, *String motechId*, *String providerId*, *String errorMessage*)

Retrieves the total count of SMS messages matching the given criteria. All string fields in this lookup are matched using the matches() operator, meaning they will be matched using the same rules that String.matches(String) uses.

#### Parameters

- **config** – the name of the configuration associated with the SMS message
- **directions** – the set of directions (inbound, outbound)
- **phoneNumber** – the number of the phone the message was received from or delivered to
- **messageContent** – the contents of the SMS message
- **timestamp** – the date-time range the timestamp of the SMS should fall into
- **deliveryStatuses** – the set of delivery status for the messages
- **providerStatus** –
- **motechId** – the id by which MOTeCH identifies the message
- **providerId** – the provider generated ID for the SMS
- **errorMessage** – the error message for the SMS

**Returns** the matching records

### findByCriteria

List<SmsRecord> **findByCriteria** (*String config*, *Set<SmsDirection> directions*, *String phoneNumber*, *String messageContent*, *Range<DateTime> timestamp*, *Set<DeliveryStatus> deliveryStatuses*, *String providerStatus*, *String motechId*, *String providerId*, *String errorMessage*, *QueryParams queryParams*)

Retrieves all sms records matching the given criteria. All string fields in this lookup are matched using the matches() operator, meaning they will be matched using the same rules that String.matches(String) uses.

#### Parameters

- **config** – the name of the configuration associated with the SMS message
- **directions** – the set of directions (inbound, outbound)
- **phoneNumber** – the number of the phone the message was received from or delivered to
- **messageContent** – the contents of the SMS message
- **timestamp** – the date-time range the timestamp of the SMS should fall into
- **deliveryStatuses** – the set of delivery status for the messages
- **providerStatus** –
- **motechId** – the id by which MOTeCH identifies the message
- **providerId** – the provider generated ID for the SMS

- **errorMessage** – the error message for the SMS
- **queryParams** – the query params controlling the ordering and size of the lookup

**Returns** the matching records

## 19.14.91 org.motechproject.sms.configs

### Config

public class **Config**

A connection to a particular SMS provider, there may be more than one config for a given provider and/or multiple connections to multiple providers. But realistically, most implementations will have one config.

### Methods

**getExcludeLastFooter**

public **Boolean** **getExcludeLastFooter** ()

**Returns** true to exclude the split footer in the last part of a split SMS message, false otherwise

**getMaxRetries**

public **Integer** **getMaxRetries** ()

Gets the maximum number of retries that will be performed when sending an SMS. If there was an error when sending the SMS, we will initiate a retry.

**Returns** the maximum number of retries

**getName**

public **String** **getName** ()

**Returns** the unique name identifying the configuration

**getProps**

public **List**<**ConfigProp**> **getProps** ()

**Returns** additional configuration properties that will be always passed to the template

**getSplitFooter**

public **String** **getSplitFooter** ()

**Returns** the footer that will be included in split SMS messages

**getSplitHeader**

public **String** **getSplitHeader** ()

**Returns** the header that will be included in split SMS messages

**getTemplateName**

public **String** **getTemplateName** ()

**Returns** the name of the configuration template for this config

**retryOrAbortStatus**

public **DeliveryStatus** **retryOrAbortStatus** (*Integer failureCount*)

Returns an appropriate **DeliveryStatus** for the given failure count. If the failure count is higher than the maximum number of retries for this configuration, then **DeliveryStatus.ABORTED** is returned. Otherwise **DeliveryStatus.RETRYING** is returned, indicating the retries are ongoing.

**Parameters**

- **failureCount** – the failure count to check

**Returns** the delivery status matching the failure count

**retryOrAbortSubject**

public **String** **retryOrAbortSubject** (*Integer failureCount*)

Returns an appropriate subject for an event indicating an SMS send failure. If the failure count is higher than the maximum number of retries for this configuration, then a subject indicating that to abort the SMS will be sent.

**Parameters**

- **failureCount** – the failure count to check

**Returns** the subject for the event

**setExcludeLastFooter**

public void **setExcludeLastFooter** (*Boolean excludeLastFooter*)

**Parameters**

- **excludeLastFooter** – true to exclude the split footer in the last part of a split SMS message, false otherwise

**setMaxRetries**

public void **setMaxRetries** (*Integer maxRetries*)

Sets the maximum number of retries that will be performed when sending an SMS. If there was an error when sending the SMS, we will initiate a retry.

**Parameters**

- **maxRetries** – the maximum number of retries

**setName**

public void **setName** (*String name*)

**Parameters**

- **name** – the unique name identifying the configuration

**setProps**

public void **setProps** (*List<ConfigProp> props*)

**Parameters**

- **props** – additional configuration properties that will be always passed to the template



**setSplitFooter**

```
public void setSplitFooter (String splitFooter)
```

**Parameters**

- **splitFooter** – the footer that will be included in split SMS messages

**setSplitHeader**

```
public void setSplitHeader (String splitHeader)
```

**Parameters**

- **splitHeader** – the header that will be included in split SMS messages

**setTemplateName**

```
public void setTemplateName (String templateName)
```

**Parameters**

- **templateName** – the name of the configuration template for this config

**ConfigProp**

```
public class ConfigProp
```

Provider-specific properties. ie: Clickatell api\_key, most will have at least username & password

**Constructors****ConfigProp**

```
public ConfigProp ()
```

Constructs a new prop without a name and a value.

**ConfigProp**

```
public ConfigProp (String name, String value)
```

Constructs the property using the given name and value

**Parameters**

- **name** – the name of the property
- **value** – the value of the property

**Methods****getName**

```
public String getName ()
```

**Returns** the name of the property

**getValue**

```
public String getValue ()
```

**Returns** the value of the property

### setName

public void **setName** (*String name*)

#### Parameters

- **name** – the name of the property

### setValue

public void **setValue** (*String value*)

#### Parameters

- **value** – the value of the property

## Configs

public class **Configs**

Represents all configs as well as the default config, sms-config.json

## Constructors

### Configs

public **Configs** ()

Creates a new instance, used by Jackson.

## Methods

### getConfig

public *Config* **getConfig** (*String name*)

Fetches the config with the given name.

#### Parameters

- **name** – the name of the config

#### Throws

- **IllegalArgumentException** – if a configuration with the given name does not exist

**Returns** the config with the given name, never null

### getConfigOrDefault

public *Config* **getConfigOrDefault** (*String name*)

Returns the config with the given name, or the default config if the provided name is null or empty.

#### Parameters

- **name** – the name of the desired configuration, or null/empty string for the default configuration

#### Throws

- **IllegalArgumentException** – if a configuration with the given name does not exist
- **IllegalStateException** – if a blank string is provided and there is no default configuration

**Returns** the matching configuration

**getConfigs**

```
public List<Config> getConfigs ()
```

**Returns** all available configurations

**getDefaultConfig**

```
public Config getDefaultConfig ()
```

Returns the default configuration, users can make one of the configurations default.

**Throws**

- **IllegalStateException** – if no default configuration is set

**Returns** the default configuration, never null

**getDefaultConfigName**

```
public String getDefaultConfigName ()
```

Returns the name of the default configuration.

**Returns** the name of the default configuration or null if it was not set

**hasConfig**

```
public boolean hasConfig (String name)
```

Checks whether a configuration with the given name exists.

**Parameters**

- **name** – the name of the configuration

**Returns** true if it exists, false otherwise

**isEmpty**

```
public boolean isEmpty ()
```

Checks whether there are no configurations available.

**Returns** true if there are no configurations, false otherwise

**setConfigs**

```
public void setConfigs (List<Config> configs)
```

**Parameters**

- **configs** – all available configurations

**setDefaultConfigName**

```
public void setDefaultConfigName (String name)
```

Sets the name of the default configuration.

**Parameters**

- **name** – the name of the default configuration

## 19.14.92 org.motechproject.sms.service

### ConfigService

public interface **ConfigService**

Config service, manages SMS Configs. A config represents the way a particular user connects to an SMS provider. See `org.motechproject.sms.configs.Config`

#### Methods

##### getConfig

`Config getConfig (String name)`

Returns the configuration with the given name.

##### Parameters

- **name** – the name of the configuration

##### Throws

- **IllegalArgumentException** – if a config with the given name does not exist

**Returns** the config with the given name, never null

##### getConfigList

`List<Config> getConfigList ()`

Returns a collection containing all SMS configurations from the system.

**Returns** all SMS configurations

##### getConfigOrDefault

`Config getConfigOrDefault (String name)`

Returns the config with the given name, or the default config if the provided name is null or empty.

##### Parameters

- **name** – the name of the desired configuration, or null/empty string for the default configuration

##### Throws

- **IllegalArgumentException** – if a configuration with the given name does not exist
- **IllegalStateException** – if a blank string is provided and there is no default configuration

**Returns** the matching configuration

##### getConfigs

`Configs getConfigs ()`

Returns an object containing all SMS configurations from the system.

**Returns** all SMS configurations

**getDefaultConfig****Config** **getDefaultConfig** ()

Returns the default configuration for sending SMS message. Only one of the configurations can be marked as the default by the user.

**Throws**

- **IllegalStateException** – if no default configuration exists

**Returns** the default configuration, never null

**getServerUrl****String** **getServerUrl** ()

Returns the base url to this MOTECH server. The url is read from the platform settings.

**Returns** the base url to this MOTECH server

**hasConfig**boolean **hasConfig** (**String** *name*)

Checks whether a configuration with the given name exists.

**Parameters**

- **name** – the name of the configuration to check

**Returns** true if the configuration exists, false otherwise

**hasConfigs**boolean **hasConfigs** ()

Checks whether there are no configurations available.

**Returns** true if there are no configurations, false otherwise

**updateConfigs**void **updateConfigs** (**Configs** *configs*)

Updates the SMS configs using the provided object. The configuration will be persisted with the Motech config system. Configurations which are no longer part of the provided object will be dropped.

**Parameters**

- **configs** – an object containing all the SMS configurations

**ConfigServiceImpl**

public class **ConfigServiceImpl** implements **ConfigService**

See [org.motechproject.sms.service.ConfigService](#)

**Constructors****ConfigServiceImpl**

public **ConfigServiceImpl** (**SettingsFacade** *settingsFacade*)

## Methods

### **getConfig**

public [Config](#) **getConfig** ([String](#) *name*)

### **getConfigList**

public [List](#)<[Config](#)> **getConfigList** ()

### **getConfigOrDefault**

public [Config](#) **getConfigOrDefault** ([String](#) *name*)

### **getConfigs**

public [Configs](#) **getConfigs** ()

### **getDefaultConfig**

public [Config](#) **getDefaultConfig** ()

### **getServerUrl**

public [String](#) **getServerUrl** ()

### **handleFileChanged**

public void **handleFileChanged** ([MotechEvent](#) *event*)

### **hasConfig**

public boolean **hasConfig** ([String](#) *name*)

### **hasConfigs**

public boolean **hasConfigs** ()

### **updateConfigs**

public void **updateConfigs** ([Configs](#) *configs*)

## OutgoingSms

public class **OutgoingSms**

Represents an outgoing SMS

## Constructors

### **OutgoingSms**

public **OutgoingSms** ()

Constructs a new instance without setting any fields.

**OutgoingSms**

public **OutgoingSms** (MotechEvent *event*)

Constructs an instance using the field map from the provided event.

**Parameters**

- **event** – the event to parse into the SMS instance

See also: [SmsEventParams](#)

**OutgoingSms**

public **OutgoingSms** (*String config*, *List<String> recipients*, *String message*, *DateTime deliveryTime*)

Creates a new instance using the provided parameters.

**Parameters**

- **config** – use this config to send the SMS, otherwise use the default config
- **recipients** – one or more recipients
- **message** – the message to send
- **deliveryTime** – the expected delivery time of the sms

**OutgoingSms**

public **OutgoingSms** (*String config*, *String recipient*, *String message*, *DateTime deliveryTime*)

Creates a new instance using the provided parameters.

**Parameters**

- **config** – use this config to send the SMS, otherwise use the default config
- **recipient** – the recipient of the SMS
- **message** – the message to send
- **deliveryTime** – the expected delivery time of the sms

**OutgoingSms**

public **OutgoingSms** (*String config*, *List<String> recipients*, *String message*)

Creates a new instance using the provided parameters.

**Parameters**

- **config** – use this config to send the SMS, otherwise use the default config
- **recipients** – the recipients of the SMS
- **message** – the message to send

**OutgoingSms**

public **OutgoingSms** (*String config*, *String recipient*, *String message*)

**OutgoingSms**

public **OutgoingSms** (*List<String> recipients*, *String message*, *DateTime deliveryTime*)

Creates a new instance using the provided parameters.

**Parameters**

- **recipients** – the recipients of the SMS

- **message** – the message to send
- **deliveryTime** – the expected delivery time of the sms

### OutgoingSms

public **OutgoingSms** (*String recipient*, *String message*, *DateTime deliveryTime*)

Creates a new instance using the provided parameters.

#### Parameters

- **recipient** – the recipient of the SMS
- **message** – the message to send
- **deliveryTime** – the expected delivery time of the sms

### OutgoingSms

public **OutgoingSms** (*List<String> recipients*, *String message*)

Creates a new instance using the provided parameters.

#### Parameters

- **recipients** – the recipients of the SMS
- **message** – the message to send

### OutgoingSms

public **OutgoingSms** (*String recipient*, *String message*)

Creates a new instance using the provided parameters.

#### Parameters

- **recipient** – the recipient of the SMS
- **message** – the message to send

## Methods

### equals

public boolean **equals** (*Object obj*)

### getConfig

public *String* **getConfig** ()

If specified, use this config to send the SMS, otherwise use the default config.

**Returns** the config that will be used for sending this SMS

### getCustomParams

public *Map<String, String>* **getCustomParams** ()

**Returns** map of custom parameters for the template



**getDeliveryTime**

```
public DateTime getDeliveryTime ()
```

Gets the delivery time for this SMS. If specified, will schedule the message for future delivery using the MOTECH scheduler.

**Returns** the delivery time for this SMS

**getFailureCount**

```
public Integer getFailureCount ()
```

Gets the failure counter, used by the SMS module to keep track of the failures and execute retries.

**Returns** the internal failure counter for this SMS

**getMessage**

```
public String getMessage ()
```

**Returns** the content of the message to send

**getMotechId**

```
public String getMotechId ()
```

Gets the MOTECH specific message GUID of this SMS.

**Returns** the motech specific GUID for this SMS

**getProviderId**

```
public String getProviderId ()
```

**Returns** the provider specific message id

**getRecipients**

```
public List<String> getRecipients ()
```

Gets the recipients for this message. An SMS can have one or more recipients.

**Returns** the recipients for the SMS

**hasConfig**

```
public Boolean hasConfig ()
```

Checks if this SMS specifies a config for sending the SMS, or whether the default configuration should be used.

**Returns** true if there's a config specified for this SMS, false if the default config is to be used

**hasDeliveryTime**

```
public Boolean hasDeliveryTime ()
```

Checks whether this SMS has a specified delivery time. If specified, will schedule the message for future delivery using the MOTECH scheduler. If not, the message should be immediately delivered.

**Returns** true if this SMS has a specific delivery time, false otherwise

**hasMotechId**

```
public Boolean hasMotechId ()
```

**hashCode**

```
public int hashCode ()
```

**setConfig**

```
public void setConfig (String config)
```

If specified, use this config to send the SMS, otherwise use the default config.

**Parameters**

- **config** – the config that will be used for sending this SMS

**setCustomParams**

```
public void setCustomParams (Map<String, String> customParams)
```

**Parameters**

- **customParams** – map of custom parameters for the template

**setDeliveryTime**

```
public void setDeliveryTime (DateTime deliveryTime)
```

Gets the delivery time for this SMS. If specified will schedule the message for future delivery using the MOTECH scheduler.

**Parameters**

- **deliveryTime** – the delivery time for this SMS

**setFailureCount**

```
public void setFailureCount (Integer failureCount)
```

Sets the failure counter, used by the SMS module to keep track of the failures and execute retries.

**Parameters**

- **failureCount** – the internal failure counter for this SMS

**setMessage**

```
public void setMessage (String message)
```

**Parameters**

- **message** – the content of the message to send

**setMotechId**

```
public void setMotechId (String motechId)
```

Sets the MOTECH specific message GUID of this SMS.

**Parameters**

- **motechId** – the motech specific GUID for this SMS

**setProviderId**

```
public void setProviderId (String providerId)
```

**Parameters**

- **providerId** – the provider specific message id

**setRecipients**

```
public void setRecipients (List<String> recipients)
```

Gets the recipients for this message. An SMS can have one or more recipients.

**Parameters**

- **recipients** – the recipients for the SMS

**toString**

```
public String toString ()
```

**SmsService**

```
public interface SmsService
```

Service that allows sending SMS messages.

**Methods****send**

```
void send (OutgoingSms message)
```

This method allows sending outgoing sms messages through HTTP. The configuration specified in the `OutgoingSms` object will be used for dealing with the provider.

**Parameters**

- **message** – the representation of the sms to send

**SmsServiceImpl**

```
public class SmsServiceImpl implements SmsService
```

Send an SMS, we really don't send here, but rather pass it on to the `SmsHttpService` which does

**Constructors****SmsServiceImpl**

```
public SmsServiceImpl (EventRelay eventRelay, MotechSchedulerService schedulerService, TemplateService templateService, ConfigService configService, SmsRecordsDataService smsRecordsDataService)
```

**Methods****send**

```
public void send (OutgoingSms sms)
```

**TemplateService**

```
public interface TemplateService
```

Template service, manages SMS Templates. A template represents the way all users connect to an SMS provider. See `org.motechproject.sms.templates.Template`

## Methods

### allTemplatesForWeb

`Map<String, TemplateForWeb> allTemplatesForWeb ()`

Retrieves all templates as objects for the UI display.

**Returns** map of the template representations, where keys are the names of the templates

### getTemplate

`Template getTemplate (String name)`

Retrieves the template with the given name.

#### Parameters

- **name** – the name of the template

#### Throws

- **IllegalArgumentException** – if no such template exists

**Returns** the template with the matching name, never null

### importTemplate

`void importTemplate (Template template)`

Imports the custom provider template provided by the user. This template will be added to the additional templates configuration.

#### Parameters

- **template** – the template to import

### importTemplates

`void importTemplates (List<Template> templateList)`

Imports the custom provider templates provided by the user. These templates will be added to the additional templates configuration.

#### Parameters

- **templateList** – the templates to import

## TemplateServiceImpl

public class **TemplateServiceImpl** implements `TemplateService`

See `org.motechproject.sms.service.TemplateService`. This implementation uses the MOTECH configuration system to store the templates.

## Constructors

### TemplateServiceImpl

public **TemplateServiceImpl** (SettingsFacade *settingsFacade*)

## Methods

### **allTemplatesForWeb**

public [Map](#)<[String](#), [TemplateForWeb](#)> **allTemplatesForWeb** ()

### **getTemplate**

public [Template](#) **getTemplate** ([String](#) *name*)

### **handleFileChanged**

public void **handleFileChanged** ([MotechEvent](#) *event*)

### **importTemplate**

public void **importTemplate** ([Template](#) *template*)

### **importTemplates**

public void **importTemplates** ([List](#)<[Template](#)> *templateList*)

## 19.14.93 org.motechproject.sms.util

### Constants

public final class **Constants**

Class holding constant [String](#) values related to permissions and roles.

### Fields

#### **HAS\_MANAGE\_SMS\_ROLE**

public static final [String](#) **HAS\_MANAGE\_SMS\_ROLE**

#### **HAS\_VIEW\_SMS\_LOGS\_ROLE**

public static final [String](#) **HAS\_VIEW\_SMS\_LOGS\_ROLE**

#### **MANAGE\_SMS\_PERMISSION**

public static final [String](#) **MANAGE\_SMS\_PERMISSION**

#### **VIEW\_SMS\_LOGS\_PERMISSION**

public static final [String](#) **VIEW\_SMS\_LOGS\_PERMISSION**

### **SmsEventParams**

public final class **SmsEventParams**

Possible Sms Event payloads (ie: params)

## Fields

### CONFIG

public static final `String` **CONFIG**  
Config that was used for this message

### CUSTOM\_PARAMS

public static final `String` **CUSTOM\_PARAMS**  
map of custom parameters

### DELIVERY\_TIME

public static final `String` **DELIVERY\_TIME**  
time at which this SMS should be sent

### FAILURE\_COUNT

public static final `String` **FAILURE\_COUNT**  
internal SMS failure counter

### MESSAGE

public static final `String` **MESSAGE**  
the text content of the SMS message

### MOTECH\_ID

public static final `String` **MOTECH\_ID**  
MOTECH unique message id

### PROVIDER\_MESSAGE\_ID

public static final `String` **PROVIDER\_MESSAGE\_ID**  
provider unique message id

### PROVIDER\_STATUS

public static final `String` **PROVIDER\_STATUS**  
provider provided SMS delivery status, sometimes holds more information than what MOTECH models

### RECIPIENT

public static final `String` **RECIPIENT**  
incoming SMS recipient (phone number)

### RECIPIENTS

public static final `String` **RECIPIENTS**  
list of recipients (phone numbers)

### SENDER

public static final `String` **SENDER**  
incoming SMS sender (phone number)

**TIMESTAMP**

public static final [String](#) **TIMESTAMP**  
date & time when this event happened

**TIMESTAMP\_DATETIME**

public static final [String](#) **TIMESTAMP\_DATETIME**  
date & time when this event happened

**TIMESTAMP\_TIME**

public static final [String](#) **TIMESTAMP\_TIME**  
UTC time when this event happened

**SmsEventSubjects**

public final class **SmsEventSubjects**  
Event subjects, mirrors DeliveryStatus

**Fields****ABORTED**

public static final [String](#) **ABORTED**

**DELIVERY\_CONFIRMED**

public static final [String](#) **DELIVERY\_CONFIRMED**

**DISPATCHED**

public static final [String](#) **DISPATCHED**

**FAILURE\_CONFIRMED**

public static final [String](#) **FAILURE\_CONFIRMED**

**INBOUND\_SMS**

public static final [String](#) **INBOUND\_SMS**

**PENDING**

public static final [String](#) **PENDING**

**RETRYING**

public static final [String](#) **RETRYING**

**SCHEDULED**

public static final [String](#) **SCHEDULED**

**SEND\_SMS**

public static final [String](#) **SEND\_SMS**

## SmsEvents

public final class **SmsEvents**

MotechEvent Helper class, builds events from provided parameters.

### Methods

#### inboundEvent

public static MotechEvent **inboundEvent** (*String config, String sender, String recipient, String message, String providerMessageId, DateTime timestamp*)

Creates an event which should be published after receiving an inbound SMS.

##### Parameters

- **config** – the configuration which received the SMS
- **sender** – the sender number
- **recipient** – the recipient number
- **message** – the content of the SMS messages
- **providerMessageId** – the provider ID of the message
- **timestamp** – the message timestamp

**Returns** a `MotechEvent` that will indicate an inbound SMS

#### outboundEvent

public static MotechEvent **outboundEvent** (*String subject, String config, List<String> recipients, String message, String motechId, String providerMessageId, Integer failureCount, String providerStatus, DateTime timestamp, Map<String, String> customParams*)

Creates an MOTECH event which will describe an outbound SMS (already sent or to be sent).

##### Parameters

- **subject** – the subject of the event
- **config** – the configuration which
- **recipients** – the recipient numbers for the SMS
- **message** – the SMS message
- **motechId** – the MOTECH ID of the message
- **providerMessageId** – the provider ID of the message
- **failureCount** – the failure counter for this message, will be used for retries
- **providerStatus** – the SMS status coming from the provider
- **timestamp** – the timestamp of the message
- **customParams** – custom parameters for the provider

**Returns** a `MotechEvent` describing the outbound SMS



## 19.14.94 org.motechproject.transliteration.hindi.service

### TransliterationService

public interface **TransliterationService**

A service for transliterating English strings to Hindi.

#### Methods

##### transliterate

*String* **transliterate** (*String data*)

Transliterates the provided English string to Hindi.

##### Parameters

- **data** – the English string to transliterate

**Returns** the input transliterated to Hindi

### TransliterationServiceImpl

public class **TransliterationServiceImpl** implements [TransliterationService](#)

The implementation of the Hindi transliteration service that transliterates English strings to Hindi using ITRANS encoding.

#### Methods

##### transliterate

public *String* **transliterate** (*String data*)

Transliterates a given string from English to Hindi using ITRANS encoding.

##### Parameters

- **data** – the string to transliterate, in English

**Returns** the input string transliterated to Hindi

## 19.15 Pill Reminder Module

### 19.15.1 Description

The Pill Reminder Module provides repeated reminders about pill regimens to users. Users are enrolled in a regimen and then reminded to take the appropriate medicines at the appropriate times of the day.

### 19.15.2 Information for Implementation

The Pill Reminder module provides implementers with the following functionality:

- Enroll users in a custom pill regimen
- Custom per-user reminders for specific medicines during defined time windows

### 19.15.3 OSGi Service API

The Pill Reminder module exposes an OSGi service to perform common operations on pill reminders. The service provides the following API:

```
public interface PillReminderService {
    // creates a new pill regimen entry in Motech Data Services (MDS) and logs a new
    // daily job in Motech's scheduler.
    void createNew(DailyPillRegimenRequest dailyPillRegimenRequest);

    // clears and then recreates existing entries in MDS and scheduler
    void renew(DailyPillRegimenRequest newDailyScheduleRequest);

    // set the date in which a patient's response to a pill reminder was last
    // last recorded.
    void dosageStatusKnown(Long pillRegimenId, Long dosageId, LocalDate lastCapturedDate);

    // get the pill regimen associated with the patient's id
    PillRegimenResponse getPillRegimen(String externalId);

    // remove the patient's pill regimen from MDS and the scheduler
    void remove(String externalID);
}
```

### 19.15.4 Common Actions

In the example code snippets below, it is the implementer's responsibility to provide an external ID for the user.

#### Enroll a User in a Regimen

```
LocalDate startDate = DateUtil.today();
LocalDate endDate = startDate.plusDays(2);

// The user is expected to take two different medicines with overlapping but different time windows
MedicineRequest medicineRequest1 = new MedicineRequest("m1", startDate, endDate);
MedicineRequest medicineRequest2 = new MedicineRequest("m2", startDate.plusDays(1), startDate.plusDays(2));
List<MedicineRequest> medicineRequests = asList(medicineRequest1, medicineRequest2);

// The user must take the medicine at 9:05
DosageRequest dosageRequest = new DosageRequest(9, 5, medicineRequests);

// The user has 5 hours to take their medicine. The system will remind them every 20 minutes with a
// The pill is only taken once a day. (Only one dosage request)
DailyPillRegimenRequest dailyPillRegimenRequest = new DailyPillRegimenRequest(userId, 5, 20, 5, asList(medicineRequests, dosageRequest));

pillReminderService.createNew(dailyPillRegimenRequest);
```

#### Respond to a Reminder Event

```
@MotechListener(subjects = EventKeys.PILLREMINDER_REMINDER_EVENT_SUBJECT)
public void handlePillReminderEvent(MotechEvent motechEvent) {
    Map<String, Object> parameters = motechEvent.getParameters();
    String patientDocId = (String) parameters.get(EventKeys.EXTERNAL_ID_KEY);
    String regimenId = (String) parameters.get(EventKeys.PILLREMINDER_ID_KEY);
```

```
String dosageId = (String) parameters.get(EventKeys.DOSAGE_ID_KEY);
int timesAlreadyCalled = (Integer) parameters.get(EventKeys.PILLREMINDER_TIMES_SENT);
int totalCallsToBeMade = (Integer) parameters.get(EventKeys.PILLREMINDER_TOTAL_TIMES_TO_SEND);

// In this example we place a call to the patient to remind them
call.execute(patientDocId, regimenId, dosageId, timesAlreadyCalled, totalCallsToBeMade);
}
```

### 19.15.5 Configuration

This module does not require any custom configuration.

### 19.15.6 Events

The Pill Reminder module emits and consumes the following events, with the keys of each payload and their corresponding type given below.

#### Emitted Events

```
EventKeys.PILLREMINDER_REMINDER_EVENT_SUBJECT
(String) EventKeys.EXTERNAL_ID_KEY
(String) EventKeys.PILLREMINDER_ID_KEY
(String) EventKeys.DOSAGE_ID_KEY
(Integer) EventKeys.PILLREMINDER_TIMES_SENT
(Integer) EventKeys.PILLREMINDER_TOTAL_TIMES_TO_SEND
```

The Pill Reminder module exposes this event to the Task module as a task trigger, allowing other modules to respond to pill reminder events using the Task module user interface.

#### Consumed Events

```
EventKeys.PILLREMINDER_REMINDER_EVENT_SUBJECT_DOSAGE_STATUS_KNOW
(Long) EventKeys.PILL_REGIMEN_ID
(Long) EventKeys.DOSAGE_ID_KEY
(Date) EventKeys.LAST_CAPTURE_DATE

EventKeys.PILLREMINDER_REMINDER_EVENT_SUBJECT_UNSUBSCRIBE
(String) EventKeys.EXTERNAL_ID_KEY
```

The Pill Reminder module exposes this event to the Task module as a task action, allowing other modules to schedule pill reminders using the Task module user interface.

## 19.16 Schedule Tracking Module

**Contents**

- [Schedule Tracking Module](#)
  - [Description](#)
  - [Information for Implementation](#)
  - [Lifecycle of a Milestone](#)
  - [Specifying a Schedule and its Milestones](#)
  - [Sample Schedule Definition](#)
  - [OSGi Service APIs](#)
  - [Events Consumed and Emitted](#)
  - [Roles and Permissions](#)

### 19.16.1 Description

The schedule tracking module allows a client to be enrolled in a clearly defined schedule. Schedules consist of specified “milestones.” Milestones represent windows of time in which certain criteria, treatments, courses, etc. should be fulfilled before moving on to the next milestone. Individuals enrolled in these schedules are sent alerts when they are due, late, or past due on the period of schedule fulfillment. Clients may be enrolled in an unlimited number of schedules. If a client is enrolled in a schedule they were already enrolled in, the service overwrites the previous enrollment with the new one.

### 19.16.2 Information for Implementation

A schedule defines a set of milestones that should be fulfilled in a specific window of time. Implementers can use the Schedule Tracking module to accomplish the following:

- Enroll a particular client in a schedule for a specific duration of time. This is accomplished by using the `ScheduleTrackingService` interface.
- Send alerts to clients that have been enrolled in a schedule.
- Define JSON documents that specify a schedule and its set of milestones.
- Query for client enrollments in schedules based on ID, schedule name, milestone name, milestone window, enrollment status, completion date, and metadata.

Client enrollments are stored in MOTech Data Services (MDS).

### 19.16.3 Lifecycle of a Milestone

- The first milestone is triggered at the start of the schedule. Subsequent milestones are started when the previous milestone is fulfilled.
- Once started, a milestone is in an *earliest*, *due*, *late*, or *max* window as per the schedule configuration. These window durations form the lifetime of the milestone.
- If a milestone is not fulfilled before the expiration of its lifetime, the associated client enrollment will be defaulted, and all of the previously configured alerts will not be raised.
- A milestone that is not fulfilled before the expiration of its lifetime is defaulted. You can also mark the milestone as defaulted within its lifetime.

### 19.16.4 Specifying a Schedule and its Milestones

Specifying the characteristics of a schedule and its milestones is typically accomplished using a JSON (JavaScript Object Notation) document. The **schedule\_tracking.properties** file specifies the directory to scan for schedule definitions. Any JSON file in the directory will be treated as a schedule definition. Multiple schedules can be defined, each in their own file. Schedules have a name and a list of milestones. A milestone has:

- A name
- A list of four standard time windows (earliest, due, late, and max)
- A list of alerts
- A map of data (not currently used)

Window durations can be customized.

### 19.16.5 Sample Schedule Definition

For a sample schedule definition, including specifying window times, absolute window schedules, alerts, and floating alerts, see *sample-schedule-definition*.

### 19.16.6 OSGi Service APIs

The schedule tracking module exposes two OSGi interfaces, `ScheduleTrackingService` and `EnrollmentActionService`.

#### `ScheduleTrackingService`

`ScheduleTrackingService` enrolls and unenrolls users, updates enrollments, fulfills milestones, queries enrollments, and provides a preview of alert timings for a given enrollment.

```
public interface ScheduleTrackingService {

    String enroll(EnrollmentRequest enrollmentRequest);

    void fulfillCurrentMilestone(String externalId, String scheduleName,
        LocalDate fulfillmentDate, Time fulfillmentTime);

    void fulfillCurrentMilestone(String externalId, String scheduleName,
        LocalDate fulfillmentDate);

    void unenroll(String externalId, List<String> scheduleNames);

    EnrollmentRecord getEnrollment(String externalId, String scheduleName);

    void updateEnrollment(String externalId, String scheduleName,
        UpdateCriteria updateCriteria);

    List<EnrollmentRecord> search(EnrollmentsQuery query);

    List<EnrollmentRecord> searchWithWindowDates(EnrollmentsQuery query);

    MilestoneAlerts getAlertTimings(EnrollmentRequest enrollmentRequest);

    void add(String scheduleJson);
```

```
Schedule getScheduleByName(String scheduleName);

List<Schedule> getAllSchedules();

void remove(String scheduleName);

}
```

### Enrolling a User in a Schedule

To enroll a user into a schedule, an `EnrollmentRequest` must be passed to the `enroll()` method in the `ScheduleTrackingService`. An `EnrollmentRequest` has the following information:

Parameter	Type	Description
externalID	String	A unique ID for the client
scheduleName	String	The name of the schedule to enroll the user in; defined in the JSON document
startingMilestoneName	String	The name of the first milestone into which the user will be directly enrolled
referenceDate	LocalDate	The date on which the schedule will start
referenceTime	Time	(Optional; defaults to midnight) Time, for fine-grained referencing
enrollmentDate	LocalDate	The date on which the user is enrolled into the schedule
enrollmentTime	Time	(Optional; defaults to midnight) Time, for fine-grained referencing
preferredAlertTime	Time	Time of day to send alerts to user
metadata	Map<String, String>	Additional information stored as property=>value pairs, e.g., facility_id=>1234

When the `ScheduleTrackingService`'s `enroll()` method is invoked, the service determines whether that client is already enrolled and active in the schedule. If the client is already enrolled, the service overwrites the previous enrollment with the new one. A new enrollment record for the client is created and added to the database.

### Fulfilling Milestones

The `fulfillCurrentMilestone()` method of the module's `ScheduleTrackingService` fulfills the current milestone of the client within an enrollment. After fulfillment of a milestone, the client moves to the next milestone in the schedule. If no more milestones remain the schedule, the enrollment is marked as complete. This fulfillment date and time is mandatory while fulfilling a milestone. The fulfillment date and time are used to make the fulfillment process idempotent. This ensures that invoking `fulfillCurrentMilestone()` more than once with the same fulfillment date and time will not make multiple fulfillments.

### Defaulted Enrollments

For any milestone in an enrollment, if the milestone has not been fulfilled by the last day of the milestone, then that enrollment is marked as defaulted. The last day of the milestone is the day when all four windows of the milestone

elapse. A defaulted enrollment will not raise any more alerts. It also cannot move to an active state, which is the default state of an enrollment that raises alerts.

### Unenrolling a User from a Schedule

The `unenroll()` method of the module's `ScheduleTrackingService` removes a user from an active enrollment. Only active enrollments can be removed. Invoking `unenroll()` on an enrollment will cause the enrollment to be marked as *UNENROLLED* in the database. *DEFAULTED* and *COMPLETED* enrollments are also preserved in the database for record keeping.

### Updating an Active Enrollment

The `updateEnrollment()` method of the module's `ScheduleTrackingService` updates an active enrollment. Currently, MOTECH supports updating only the metadata field of an active enrollment. Metadata property => value pair can be updated or inserted but cannot be deleted from an existing enrollment.

Example:

Consider an active enrollment with the following attributes:

external id : "foo" schedule name : "some\_schedule" metadata value pairs : {foo1: bar1; foo2: bar2}

```
HashMap<String, String> toBeUpdatedMetadata = new HashMap<String, String>();  
  
toBeUpdatedMetadata.put("foo2", "val2");  
  
toBeUpdatedMetadata.put("foo3", "val3");  
  
UpdateCriteria updateCriteria = new UpdateCriteria().Metadata(toBeUpdatedMetadata);  
  
scheduleTrackingService.updateEnrollment("foo", "some_schedule", updateCriteria);
```

will update the metadata of the enrollment as {foo1:bar1; foo2: val2; foo3: val3}

### Previewing Alert Timings

The `getAlertTimings()` method of the module's `ScheduleTrackingService` provides a preview of alert timings given a particular enrollment request. For the given enrollment request, the alert timings of all windows of the current milestone will be returned. This gives an idea of the alerts that a client might miss if the enrollment were to be scheduled on the date specified by the enrollment request. This is useful in cases where the client needs to be enrolled into the middle of the milestone but should not miss any alerts. After previewing the alert timings, clients would be enrolled on a reference date that results in a schedule with no elapsed alerts.

### Querying the API

The `search()` method of the module's `ScheduleTrackingService` allows for querying enrollments. This allows clients to find enrollments using various criteria. Queries can be performed based on the following list of criteria:

- `havingExternalId(externalId)`
- `havingSchedule(scheduleNames...)`
- `havingCurrentMilestone(milestoneName)`
- `havingWindowStartingDuring(WindowName, DateTime start, DateTime end)`

- `havingWindowEndingDuring(WindowName, DateTime start, DateTime end)`
- `currentlyInWindow(WindowNames...)`
- `havingState(EnrollmentStatus)` (ACTIVE, DEFAULTED, COMPLETED, or UNENROLLED)
- `completedDuring(DateTime start, DateTime end)`
- `havingMetadata(key, value)`

Each of these methods returns an `EnrollmentsQuery` object, which the `search()` method takes as a parameter.

Examples:

```
scheduleTrackingService.search(new EnrollmentsQuery().havingState("active"))
```

will find all active enrollments.

```
scheduleTrackingService.search(  
    new EnrollmentsQuery()  
        .havingSchedule("IPTI Schedule")  
        .havingState("active")  
        .havingWindowStartingDuring(WindowName.due, weeksAgo(1), now))
```

will find active enrollments enrolled into the IPTI Schedule that will enter the due window any time in the next one week.

The return value of the `search()` method is a list of `EnrollmentRecords`. An `EnrollmentRecord` represents an enrollment in the system. `EnrollmentRecords` contain an external id, schedule name, preferred alert time, reference date and time, enrollment date and time, start dates for each of the four windows, and a reference to the current milestone.

```
public class EnrollmentRecord {  
    private String externalId;  
    private String scheduleName;  
    private String currentMilestoneName;  
    private DateTime referenceDateTime;  
    private DateTime enrollmentDateTime;  
    private Time preferredAlertTime;  
    private DateTime earliestStart;  
    private DateTime dueStart;  
    private DateTime lateStart;  
    private DateTime maxStart;  
    private String status;  
    private Map<String, String> metadata;  
}
```



## EnrollmentActionService

EnrollmentActionService is a facade for ScheduleTrackingService that acts as a proxy for the Tasks module. Its two methods, enroll() and unenroll(), are exposed as task actions.

```
public interface EnrollmentActionService {  
  
    void enroll(String externalId, String scheduleName, String preferredAlertTime,  
               DateTime referenceDate, String referenceTime, DateTime enrollmentDate,  
               String enrollmentTime, String startingMilestoneName);  
  
    void unenroll(String externalId, String scheduleName);  
  
}
```

## 19.16.7 Events Consumed and Emitted

### Consumed Event

#### MILESTONE\_DEFAULTED

This module exposes an EndOfMilestoneListener, which handles and consumes events with the subject EventSubjects.MILESTONE\_DEFAULTED.

EventSubjects.MILESTONE\_DEFAULTED (org.motechproject.scheduletracking.milestone.defaulted)

Parameters/Payload:

```
EventDataKeys.ENROLLMENT_ID (enrollmentId)  
  
EventDataKeys.EXTERNAL_ID (externalId)  
  
MotechSchedulerService.JOB_ID_KEY (jobId)
```

### Emitted Events

#### MILESTONE\_ALERT

The EnrollmentAlertService emits events with the subject EventSubjects.MILESTONE\_ALERT.

EventSubjects.MILESTONE\_ALERT (org.motechproject.scheduletracking.milestone.alert)

Parameters/Payload:

```
EventDataKeys.WINDOW_NAME (windowName)  
  
EventDataKeys.MILESTONE_NAME (milestoneAlert)  
  
EventDataKeys.SCHEDULE_NAME (scheduleName)  
  
EventDataKeys.EXTERNAL_ID (externalId)  
  
EventDataKeys.REFERENCE_DATE (referenceDateTime)
```

Milestone alert events are scheduled for the current milestone per each alert definition.

## **MILESTONE\_DEFAULTED**

The EnrollmentDefaultmentService emits events with the subject EventSubjects.MILESTONE\_DEFAULTED.

EventSubjects.MILESTONE\_DEFAULTED (org.motechproject.scheduletracking.milestone.defaulted)

Parameters/Payload:

EventDataKeys.ENROLLMENT\_ID (enrollmentId)

EventDataKeys.EXTERNAL\_ID (externalId)

MotechSchedulerService.JOB\_ID\_KEY (jobId)

If the milestone has not been fulfilled by the last day of the milestone, then it is defaulted. Defaultment jobs are scheduled on the day a milestone would be defaulted. Their role is to capture and save the defaulted state of the milestone.

## **USER\_ENROLLED**

The EnrollmentService implementation emits events with the subject EventSubjects.USER\_ENROLLED.

EventSubjects.USER\_ENROLLED (org.motechproject.scheduletracking.user.enrolled)

Parameters/Payload:

EventDataKeys.EXTERNAL\_ID (externalID)

EventDataKeys.SCHEDULE\_NAME (scheduleName)

EventDataKeys.MILESTONE\_NAME (startingMilestoneName)

EventDataKeys.PREFERRED\_ALERT\_TIME (preferredAlertTime)

EventDataKeys.REFERENCE\_DATE (referenceDate)

EventDataKeys.REFERENCE\_TIME (referenceTime)

EventDataKeys.ENROLLMENT\_DATE (enrollmentDate)

EventDataKeys.ENROLLMENT\_TIME (enrollmentTime)

## **USER\_UNENROLLED**

The EnrollmentService implementation emits events with the subject EventSubjects.USER\_UNENROLLED.

EventSubjects.USER\_UNENROLLED (org.motechproject.scheduletracking.user.unenrolled)

Parameters/Payload:

EventDataKeys.EXTERNAL\_ID (externalID)

EventDataKeys.SCHEDULE\_NAME (scheduleName)

## 19.16.8 Roles and Permissions

The Schedule Tracking module does not define any roles or permissions.

## 19.17 SMS Module

### 19.17.1 Description

The SMS API module provides implementers with an event-driven SMS framework. Implementers can use this module on its own to send SMS messages via the user interface. This module can also be combined with other modules along with MOTeCH core functionalities to create an events based SMS messaging campaign.

### 19.17.2 Setup and Configuration

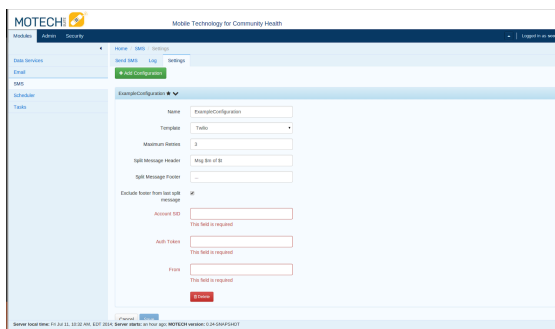
#### Choose SMS Provider

Choose an SMS service provider and set up an account:

- [Clickatell](#)
- [KooKoo](#)
- [Nuntium](#)
- [Plivo](#)
- [Rancard](#)
- [Twilio](#)
- [Voxeo](#)

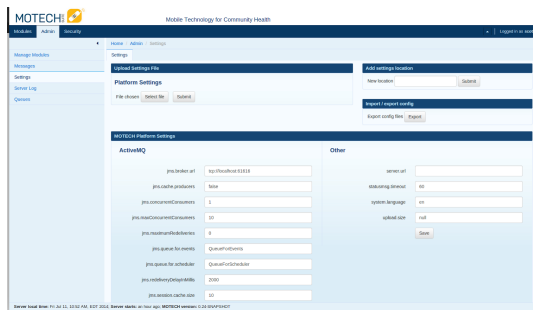
#### SMS Provider Configurations

Click on Modules >> SMS >> Settings

The screenshot shows the MOTeCH web interface for configuring SMS settings. The page has a sidebar on the left with navigation links: Home, Modules, Settings, and Tools. The main content area is titled 'SMS Settings' and contains a 'Send SMS' button and a 'Log' button. Below these is a section for 'SMS Configuration' with a dropdown menu showing 'ExampleConfiguration'. The configuration fields include: Name (ExampleConfiguration), Template (Text), Maximum Retries (3), Split Message Header (Split by all B), Split Message Footer ( ), Exclude from this list ( ), Account ID ( ), Auth Token ( ), and From ( ). Each of the last four fields has a red error message 'This field is required'. At the bottom right is a 'Save' button. The footer of the page contains the text: 'Copyright 2012-2013, 2014-2015, 2016-2017, 2018-2019, 2020-2021, 2022-2023, 2024-2025, 2026-2027, 2028-2029, 2030-2031, 2032-2033, 2034-2035, 2036-2037, 2038-2039, 2040-2041, 2042-2043, 2044-2045, 2046-2047, 2048-2049, 2050-2051, 2052-2053, 2054-2055, 2056-2057, 2058-2059, 2060-2061, 2062-2063, 2064-2065, 2066-2067, 2068-2069, 2070-2071, 2072-2073, 2074-2075, 2076-2077, 2078-2079, 2080-2081, 2082-2083, 2084-2085, 2086-2087, 2088-2089, 2090-2091, 2092-2093, 2094-2095, 2096-2097, 2098-2099, 2100-2101, 2102-2103, 2104-2105, 2106-2107, 2108-2109, 2110-2111, 2112-2113, 2114-2115, 2116-2117, 2118-2119, 2120-2121, 2122-2123, 2124-2125, 2126-2127, 2128-2129, 2130-2131, 2132-2133, 2134-2135, 2136-2137, 2138-2139, 2140-2141, 2142-2143, 2144-2145, 2146-2147, 2148-2149, 2150-2151, 2152-2153, 2154-2155, 2156-2157, 2158-2159, 2160-2161, 2162-2163, 2164-2165, 2166-2167, 2168-2169, 2170-2171, 2172-2173, 2174-2175, 2176-2177, 2178-2179, 2180-2181, 2182-2183, 2184-2185, 2186-2187, 2188-2189, 2190-2191, 2192-2193, 2194-2195, 2196-2197, 2198-2199, 2200-2201, 2202-2203, 2204-2205, 2206-2207, 2208-2209, 2210-2211, 2212-2213, 2214-2215, 2216-2217, 2218-2219, 2220-2221, 2222-2223, 2224-2225, 2226-2227, 2228-2229, 2230-2231, 2232-2233, 2234-2235, 2236-2237, 2238-2239, 2240-2241, 2242-2243, 2244-2245, 2246-2247, 2248-2249, 2250-2251, 2252-2253, 2254-2255, 2256-2257, 2258-2259, 2260-2261, 2262-2263, 2264-2265, 2266-2267, 2268-2269, 2270-2271, 2272-2273, 2274-2275, 2276-2277, 2278-2279, 2280-2281, 2282-2283, 2284-2285, 2286-2287, 2288-2289, 2290-2291, 2292-2293, 2294-2295, 2296-2297, 2298-2299, 2300-2301, 2302-2303, 2304-2305, 2306-2307, 2308-2309, 2310-2311, 2312-2313, 2314-2315, 2316-2317, 2318-2319, 2320-2321, 2322-2323, 2324-2325, 2326-2327, 2328-2329, 2330-2331, 2332-2333, 2334-2335, 2336-2337, 2338-2339, 2340-2341, 2342-2343, 2344-2345, 2346-2347, 2348-2349, 2350-2351, 2352-2353, 2354-2355, 2356-2357, 2358-2359, 2360-2361, 2362-2363, 2364-2365, 2366-2367, 2368-2369, 2370-2371, 2372-2373, 2374-2375, 2376-2377, 2378-2379, 2380-2381, 2382-2383, 2384-2385, 2386-2387, 2388-2389, 2390-2391, 2392-2393, 2394-2395, 2396-2397, 2398-2399, 2400-2401, 2402-2403, 2404-2405, 2406-2407, 2408-2409, 2410-2411, 2412-2413, 2414-2415, 2416-2417, 2418-2419, 2420-2421, 2422-2423, 2424-2425, 2426-2427, 2428-2429, 2430-2431, 2432-2433, 2434-2435, 2436-2437, 2438-2439, 2440-2441, 2442-2443, 2444-2445, 2446-2447, 2448-2449, 2450-2451, 2452-2453, 2454-2455, 2456-2457, 2458-2459, 2460-2461, 2462-2463, 2464-2465, 2466-2467, 2468-2469, 2470-2471, 2472-2473, 2474-2475, 2476-2477, 2478-2479, 2480-2481, 2482-2483, 2484-2485, 2486-2487, 2488-2489, 2490-2491, 2492-2493, 2494-2495, 2496-2497, 2498-2499, 2500-2501, 2502-2503, 2504-2505, 2506-2507, 2508-2509, 2510-2511, 2512-2513, 2514-2515, 2516-2517, 2518-2519, 2520-2521, 2522-2523, 2524-2525, 2526-2527, 2528-2529, 2530-2531, 2532-2533, 2534-2535, 2536-2537, 2538-2539, 2540-2541, 2542-2543, 2544-2545, 2546-2547, 2548-2549, 2550-2551, 2552-2553, 2554-2555, 2556-2557, 2558-2559, 2560-2561, 2562-2563, 2564-2565, 2566-2567, 2568-2569, 2570-2571, 2572-2573, 2574-2575, 2576-2577, 2578-2579, 2580-2581, 2582-2583, 2584-2585, 2586-2587, 2588-2589, 2590-2591, 2592-2593, 2594-2595, 2596-2597, 2598-2599, 2600-2601, 2602-2603, 2604-2605, 2606-2607, 2608-2609, 2610-2611, 2612-2613, 2614-2615, 2616-2617, 2618-2619, 2620-2621, 2622-2623, 2624-2625, 2626-2627, 2628-2629, 2630-2631, 2632-2633, 2634-2635, 2636-2637, 2638-2639, 2640-2641, 2642-2643, 2644-2645, 2646-2647, 2648-2649, 2650-2651, 2652-2653, 2654-2655, 2656-2657, 2658-2659, 2660-2661, 2662-2663, 2664-2665, 2666-2667, 2668-2669, 2670-2671, 2672-2673, 2674-2675, 2676-2677, 2678-2679, 2680-2681, 2682-2683, 2684-2685, 2686-2687, 2688-2689, 2690-2691, 2692-2693, 2694-2695, 2696-2697, 2698-2699, 2700-2701, 2702-2703, 2704-2705, 2706-2707, 2708-2709, 2710-2711, 2712-2713, 2714-2715, 2716-2717, 2718-2719, 2720-2721, 2722-2723, 2724-2725, 2726-2727, 2728-2729, 2730-2731, 2732-2733, 2734-2735, 2736-2737, 2738-2739, 2740-2741, 2742-2743, 2744-2745, 2746-2747, 2748-2749, 2750-2751, 2752-2753, 2754-2755, 2756-2757, 2758-2759, 2760-2761, 2762-2763, 2764-2765, 2766-2767, 2768-2769, 2770-2771, 2772-2773, 2774-2775, 2776-2777, 2778-2779, 2780-2781, 2782-2783, 2784-2785, 2786-2787, 2788-2789, 2790-2791, 2792-2793, 2794-2795, 2796-2797, 2798-2799, 2800-2801, 2802-2803, 2804-2805, 2806-2807, 2808-2809, 2810-2811, 2812-2813, 2814-2815, 2816-2817, 2818-2819, 2820-2821, 2822-2823, 2824-2825, 2826-2827, 2828-2829, 2830-2831, 2832-2833, 2834-2835, 2836-2837, 2838-2839, 2840-2841, 2842-2843, 2844-2845, 2846-2847, 2848-2849, 2850-2851, 2852-2853, 2854-2855, 2856-2857, 2858-2859, 2860-2861, 2862-2863, 2864-2865, 2866-2867, 2868-2869, 2870-2871, 2872-2873, 2874-2875, 2876-2877, 2878-2879, 2880-2881, 2882-2883, 2884-2885, 2886-2887, 2888-2889, 2890-2891, 2892-2893, 2894-2895, 2896-2897, 2898-2899, 2900-2901, 2902-2903, 2904-2905, 2906-2907, 2908-2909, 2910-2911, 2912-2913, 2914-2915, 2916-2917, 2918-2919, 2920-2921, 2922-2923, 2924-2925, 2926-2927, 2928-2929, 2930-2931, 2932-2933, 2934-2935, 2936-2937, 2938-2939, 2940-2941, 2942-2943, 2944-2945, 2946-2947, 2948-2949, 2950-2951, 2952-2953, 2954-2955, 2956-2957, 2958-2959, 2960-2961, 2962-2963, 2964-2965, 2966-2967, 2968-2969, 2970-2971, 2972-2973, 2974-2975, 2976-2977, 2978-2979, 2980-2981, 2982-2983, 2984-2985, 2986-2987, 2988-2989, 2990-2991, 2992-2993, 2994-2995, 2996-2997, 2998-2999, 3000-3001, 3002-3003, 3004-3005, 3006-3007, 3008-3009, 3010-3011, 3012-3013, 3014-3015, 3016-3017, 3018-3019, 3020-3021, 3022-3023, 3024-3025, 3026-3027, 3028-3029, 3030-3031, 3032-3033, 3034-3035, 3036-3037, 3038-3039, 3040-3041, 3042-3043, 3044-3045, 3046-3047, 3048-3049, 3050-3051, 3052-3053, 3054-3055, 3056-3057, 3058-3059, 3060-3061, 3062-3063, 3064-3065, 3066-3067, 3068-3069, 3070-3071, 3072-3073, 3074-3075, 3076-3077, 3078-3079, 3080-3081, 3082-3083, 3084-3085, 3086-3087, 3088-3089, 3090-3091, 3092-3093, 3094-3095, 3096-3097, 3098-3099, 3100-3101, 3102-3103, 3104-3105, 3106-3107, 3108-3109, 3110-3111, 3112-3113, 3114-3115, 3116-3117, 3118-3119, 3120-3121, 3122-3123, 3124-3125, 3126-3127, 3128-3129, 3130-3131, 3132-3133, 3134-3135, 3136-3137, 3138-3139, 3140-3141, 3142-3143, 3144-3145, 3146-3147, 3148-3149, 3150-3151, 3152-3153, 3154-3155, 3156-3157, 3158-3159, 3160-3161, 3162-3163, 3164-3165, 3166-3167, 3168-3169, 3170-3171, 3172-3173, 3174-3175, 3176-3177, 3178-3179, 3180-3181, 3182-3183, 3184-3185, 3186-3187, 3188-3189, 3190-3191, 3192-3193, 3194-3195, 3196-3197, 3198-3199, 3200-3201, 3202-3203, 3204-3205, 3206-3207, 3208-3209, 3210-3211, 3212-3213, 3214-3215, 3216-3217, 3218-3219, 3220-3221, 3222-3223, 3224-3225, 3226-3227, 3228-3229, 3230-3231, 3232-3233, 3234-3235, 3236-3237, 3238-3239, 3240-3241, 3242-3243, 3244-3245, 3246-3247, 3248-3249, 3250-3251, 3252-3253, 3254-3255, 3256-3257, 3258-3259, 3260-3261, 3262-3263, 3264-3265, 3266-3267, 3268-3269, 3270-3271, 3272-3273, 3274-3275, 3276-3277, 3278-3279, 3280-3281, 3282-3283, 3284-3285, 3286-3287, 3288-3289, 3290-3291, 3292-3293, 3294-3295, 3296-3297, 3298-3299, 3300-3301, 3302-3303, 3304-3305, 3306-3307, 3308-3309, 3310-3311, 3312-3313, 3314-3315, 3316-3317, 3318-3319, 3320-3321, 3322-3323, 3324-3325, 3326-3327, 3328-3329, 3330-3331, 3332-3333, 3334-3335, 3336-3337, 3338-3339, 3340-3341, 3342-3343, 3344-3345, 3346-3347, 3348-3349, 3350-3351, 3352-3353, 3354-3355, 3356-3357, 3358-3359, 3360-3361, 3362-3363, 3364-3365, 3366-3367, 3368-3369, 3370-3371, 3372-3373, 3374-3375, 3376-3377, 3378-3379, 3380-3381, 3382-3383, 3384-3385, 3386-3387, 3388-3389, 3390-3391, 3392-3393, 3394-3395, 3396-3397, 3398-3399, 3400-3401, 3402-3403, 3404-3405, 3406-3407, 3408-3409, 3410-3411, 3412-3413, 3414-3415, 3416-3417, 3418-3419, 3420-3421, 3422-3423, 3424-3425, 3426-3427, 3428-3429, 3430-3431, 3432-3433, 3434-3435, 3436-3437, 3438-3439, 3440-3441, 3442-3443, 3444-3445, 3446-3447, 3448-3449, 3450-3451, 3452-3453, 3454-3455, 3456-3457, 3458-3459, 3460-3461, 3462-3463, 3464-3465, 3466-3467, 3468-3469, 3470-3471, 3472-3473, 3474-3475, 3476-3477, 3478-3479, 3480-3481, 3482-3483, 3484-3485, 3486-3487, 3488-3489, 3490-3491, 3492-3493, 3494-3495, 3496-3497, 3498-3499, 3500-3501, 3502-3503, 3504-3505, 3506-3507, 3508-3509, 3510-3511, 3512-3513, 3514-3515, 3516-3517, 3518-3519, 3520-3521, 3522-3523, 3524-3525, 3526-3527, 3528-3529, 3530-3531, 3532-3533, 3534-3535, 3536-3537, 3538-3539, 3540-3541, 3542-3543, 3544-3545, 3546-3547, 3548-3549, 3550-3551, 3552-3553, 3554-3555, 3556-3557, 3558-3559, 3560-3561, 3562-3563, 3564-3565, 3566-3567, 3568-3569, 3570-3571, 3572-3573, 3574-3575, 3576-3577, 3578-3579, 3580-3581, 3582-3583, 3584-3585, 3586-3587, 3588-3589, 3590-3591, 3592-3593, 3594-3595, 3596-3597, 3598-3599, 3600-3601, 3602-3603, 3604-3605, 3606-3607, 3608-3609, 3610-3611, 3612-3613, 3614-3615, 3616-3617, 3618-3619, 3620-3621, 3622-3623, 3624-3625, 3626-3627, 3628-3629, 3630-3631, 3632-3633, 3634-3635, 3636-3637, 3638-3639, 3640-3641, 3642-3643, 3644-3645, 3646-3647, 3648-3649, 3650-3651, 3652-3653, 3654-3655, 3656-3657, 3658-3659, 3660-3661, 3662-3663, 3664-3665, 3666-3667, 3668-3669, 3670-3671, 3672-3673, 3674-3675, 3676-3677, 3678-3679, 3680-3681, 3682-3683, 3684-3685, 3686-3687, 3688-3689, 3690-3691, 3692-3693, 3694-3695, 3696-3697, 3698-3699, 3700-3701, 3702-3703, 3704-3705, 3706-3707, 3708-3709, 3710-3711, 3712-3713, 3714-3715, 3716-3717, 3718-3719, 3720-3721, 3722-3723, 3724-3725, 3726-3727, 3728-3729, 3730-3731, 3732-3733, 3734-3735, 3736-3737, 3738-3739, 3740-3741, 3742-3743, 3744-3745, 3746-3747, 3748-3749, 3750-3751, 3752-3753, 3754-3755, 3756-3757, 3758-3759, 3760-3761, 3762-3763, 3764-3765, 3766-3767, 3768-3769, 3770-3771, 3772-3773, 3774-3775, 3776-3777, 3778-3779, 3780-3781, 3782-3783, 3784-3785, 3786-3787, 3788-3789, 3790-3791, 3792-3793, 3794-3795, 3796-3797, 3798-3799, 3800-3801, 3802-3803, 3804-3805, 3806-3807, 3808-3809, 3810-3811, 3812-3813, 3814-3815, 3816-3817, 3818-3819, 3820-3821, 3822-3823, 3824-3825, 3826-3827, 3828-3829, 3830-3831, 3832-3833, 3834-3835, 3836-3837, 3838-3839, 3840-3841, 3842-3843, 3844-3845, 3846-3847, 3848-3849, 3850-3851, 3852-3853, 3854-3855, 3856-3857, 3858-3859, 3860-3861, 3862-3863, 3864-3865, 3866-3867, 3868-3869, 3870-3871, 3872-3873, 3874-3875, 3876-3877, 3878-3879, 3880-3881, 3882-3883, 3884-3885, 3886-3887, 3888-3889, 3890-3891, 3892-3893, 3894-3895, 3896-3897, 3898-3899, 3900-3901, 3902-3903, 3904-3905, 3906-3907, 3908-3909, 3910-3911, 3912-3913, 3914-3915, 3916-3917, 3918-3919, 3920-3921, 3922-3923, 3924-3925, 3926-3927, 3928-3929, 3930-3931, 3932-3933, 3934-3935, 3936-3937, 3938-3939, 3940-3941, 3942-3943, 3944-3945, 3946-3947, 3948-3949, 3950-3951, 3952-3953, 3954-3955, 3956-3957, 3958-3959, 3960-3961, 3962-3963, 3964-3965, 3966-3967, 3968-3969, 3970-3971, 3972-3973, 3974-3975, 3976-3977, 3978-3979, 3980-3981, 3982-3983, 3984-3985, 3986-3987, 3988-3989, 3990-3991, 3992-3993, 3994-3995, 3996-3997, 3998-3999, 4000-4001, 4002-4003, 4004-4005, 4006-4007, 4008-4009, 4010-4011, 4012-4013, 4014-4015, 4016-4017, 4018-4019, 4020-4021, 4022-4023, 4024-4025, 4026-4027, 4028-4029, 4030-4031, 4032-4033, 4034-4035, 4036-4037, 4038-4039, 4040-4041, 4042-4043, 4044-4045, 4046-4047, 4048-4049, 4050-4051, 4052-4053, 4054-4055, 4056-4057, 4058-4059, 4060-4061, 4062-4063, 4064-4065, 4066-4067, 4068-4069, 4070-4071, 4072-4073, 4074-4075, 4076-4077, 4078-4079, 4080-4081, 4082-4083, 4084-4085, 4086-4087, 4088-4089, 4090-4091, 4092-4093, 4094-4095, 4096-4097, 4098-4099,

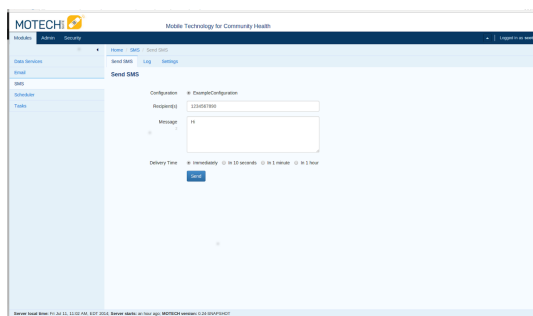
## Configure Your Server URL

- Click on Admin >> Settings
- Enter your server URL, including /motech-platform-server
- Example: <http://123.456.789.00:8080/motech-platform-server>



## Send a Test SMS

1. Click on Modules >> SMS
2. Select which configuration you want to use
3. Enter the phone number(s) you want to use, separated by a comma
4. Enter the message body
5. Select your delivery time
6. Click on Send



## 19.17.3 Tasks

The SMS module produces the following task triggers:

- Outbound SMS - Retrying
- Outbound SMS - Aborted
- Outbound SMS - Scheduled
- Outbound SMS - Pending
- Outbound SMS - Dispatched
- Outbound SMS - Delivery Confirmed
- Outbound SM - Failure Confirmed

- Inbound SMS

These events can be used with the Tasks module to trigger actions in other modules. The SMS module also listens for the “Send SMS” event. When a task emits a send SMS event, the SMS module consumes the event, extracts the payload, and sends an SMS based on the information in the payload. See the Tasks Module documentation for more information on how to create tasks.

### 19.17.4 Events

All of the events used by SMS module can also be implemented programmatically when developing a new MOTECH module. The following is a list of all events emitted and consumed by the SMS module.

#### Events Consumed

- SmsEventSubjects.SEND\_SMS
- SmsEventSubjects.PENDING
- SmsEventSubjects.SCHEDULED
- SmsEventSubjects.RETRYING
- SmsEventSubjects.INBOUND\_SMS

#### Events Emitted

- SmsEventSubjects.PENDING
- SmsEventSubjects.SCHEDULED
- SmsEventSubjects.RETRYING
- SmsEventSubjects.ABORTED
- SmsEventSubjects.DISPATCHED
- SmsEventSubjects.DELIVERY\_CONFIRMED
- SmsEventSubjects.FAILURE\_CONFIRMED

### 19.17.5 API

The SMS module provides a single service interface for raising SendSMS events in the system. The SMS API will split a single message into multiple messages based on a maximum character count of 160. Each split message will have its own send SMS event raised in the system.



It is possible to listen for an inbound SMS by implementing an event handler. Use the @MotechListener annotation to listen for the inbound SMS event:

```
@MotechListener(subjects = "inbound_sms")
```



## A

- ABORTED (Java field), 493, 523
- ABSOLUTE (Java field), 365
- AbsoluteCampaign (Java class), 357
- AbsoluteCampaign() (Java constructor), 357
- AbsoluteCampaign(String, List) (Java constructor), 357
- AbsoluteCampaignMessage (Java class), 357
- AbsoluteCampaignMessage(CampaignMessageRecord) (Java constructor), 357
- AbsoluteCampaignMessage(String, List, List, String, Time, LocalDate) (Java constructor), 357
- AbsoluteCampaignMessage(Time, LocalDate) (Java constructor), 357
- AbstractID (Java class), 207
- AbstractUniqueID (Java class), 208
- ACCEPTED (Java field), 336
- AccountConfig (Java class), 133
- ACTIVE (Java field), 111, 360, 457
- Active (Java field), 387
- ActivityDataService (Java interface), 389
- ActivityRecord (Java class), 382
- ActivityRecord() (Java constructor), 382
- ActivityRecord(String, String, String, String, DateTime, DateTime, ActivityState) (Java constructor), 382
- ActivityRecord(String, String, String, String, String, double, DateTime, DateTime, ActivityState) (Java constructor), 382
- ActivityService (Java interface), 391
- ActivityState (Java enum), 383
- add(String) (Java method), 486, 491
- addAlert(WindowName, Alert) (Java method), 458
- addAlerts(Alert) (Java method), 461
- addAppointment(Appointment) (Java method), 96
- addAppointments(List) (Java method), 96
- addAttribute(OpenMRSAttribute) (Java method), 411
- addAttribute(String, String) (Java method), 162
- addContent(Content) (Java method), 119, 122
- addContent(String, String, String, String, MultipartFile) (Java method), 127
- addDependentObservation(OpenMRSObservation) (Java method), 408
- addFieldValue(String, String) (Java method), 144
- addFormValueElement(String, FormValueElement) (Java method), 162
- addLanguage(String) (Java method), 131
- addLoggableEvents(List) (Java method), 308
- addMilestones(Milestone) (Java method), 462
- addQueryParams(URIBuilder) (Java method), 188
- Address (Java class), 208
- Address() (Java constructor), 209
- Address(Set, String) (Java constructor), 209
- Address(String) (Java constructor), 209
- AddressDataService (Java interface), 259
- AddressLine (Java class), 210
- AddressLine() (Java constructor), 210
- AddressLine(String) (Java constructor), 210
- AddressLine(String, String) (Java constructor), 210
- AddressLineDataService (Java interface), 259
- afterCompletion(MessageContext, Object, Exception) (Java method), 267
- Alert (Java class), 82, 452
- Alert() (Java constructor), 82, 452
- Alert(Period, Period, int, int, boolean) (Java constructor), 452
- Alert(String, AlertType, AlertStatus, int, Map) (Java constructor), 82
- Alert(String, String, String, AlertType, AlertStatus, int, Map) (Java constructor), 82
- ALERT\_DATA (Java field), 74
- ALERT\_DESCRIPTION (Java field), 74
- ALERT\_ID (Java field), 74
- ALERT\_NAME (Java field), 74
- ALERT\_PRIORITY (Java field), 74, 79
- ALERT\_STATUS (Java field), 74, 79
- ALERT\_TYPE (Java field), 74, 79
- AlertCriteria (Java class), 75
- AlertCriteria() (Java constructor), 75
- AlertEventHandler (Java class), 87
- AlertFilter (Java class), 88
- AlertFilter(AlertsDataService) (Java constructor), 88

alertPriority() (Java method), 75  
AlertRecord (Java class), 463  
alerts() (Java method), 464  
AlertsController (Java class), 90  
AlertsController(AlertsDataService) (Java constructor), 90  
AlertsDataService (Java interface), 78  
AlertService (Java interface), 77  
AlertServiceImpl (Java class), 88  
AlertStatus (Java enum), 85  
alertStatus() (Java method), 75  
AlertType (Java enum), 86  
alertType() (Java method), 75  
AlertUpdater (Java enum), 86  
AlertWindow (Java class), 453  
AlertWindow(DateTime, DateTime, DateTime, Time, Alert) (Java constructor), 454  
allConfigs() (Java method), 350  
AllEnrollments (Java class), 474  
allFacilities() (Java method), 263  
allOrganizations() (Java method), 263  
allPossibleAlerts() (Java method), 454  
allProviders() (Java method), 264  
allServices() (Java method), 264  
allTemplates() (Java method), 351  
allTemplatesForWeb() (Java method), 520, 521  
API\_KEY (Java field), 176  
ApplicationErrors (Java enum), 98, 325  
Appointment (Java class), 90  
Appointment() (Java constructor), 90  
Appointment(String, DateTime, DateTime, AppointmentStatus, boolean, Period, DateTime) (Java constructor), 90  
AppointmentChangeRecord (Java class), 93  
AppointmentChangeRecord(String, String, DateTime, AppointmentStatus, AppointmentStatus) (Java constructor), 93  
AppointmentChangeRecordDataService (Java interface), 95  
AppointmentDataService (Java interface), 95  
AppointmentException (Java class), 94  
AppointmentException(String) (Java constructor), 94  
AppointmentException(String, Throwable) (Java constructor), 94  
AppointmentService (Java interface), 96  
AppointmentStatus (Java enum), 94  
AppSchemaChangeEventHandler (Java class), 170  
AppStructureResponseJson (Java class), 138  
ATTRIBUTE\_OPTION\_COMBO (Java field), 274, 302  
AttributeDto (Java class), 277  
ATTRIBUTES (Java field), 176  
attrValue(String) (Java method), 411  
autocompleteField(String, String) (Java method), 127

## B

BAD\_REQUEST (Java field), 98, 325  
BASE\_SUBJECT (Java field), 74, 352, 419, 435  
BaseDto (Java class), 278  
BaseMainEntity (Java class), 211  
BatchError (Java class), 99  
BatchError() (Java constructor), 99  
BatchErrors (Java interface), 100  
BatchException (Java class), 101  
BatchException(BatchErrors) (Java constructor), 101  
BatchException(BatchErrors, String) (Java constructor), 101  
BatchException(BatchErrors, Throwable) (Java constructor), 101  
BatchException(BatchErrors, Throwable, String) (Java constructor), 101  
BatchJob (Java class), 103  
BatchJob() (Java constructor), 103  
BatchJob(Integer, String) (Java constructor), 103  
BatchJob(Integer, String, String) (Java constructor), 103  
BatchJobDTO (Java class), 107  
BatchJobListDTO (Java class), 109  
BatchJobMDSService (Java interface), 106  
BatchJobParameterMDSService (Java interface), 106  
BatchJobParameters (Java class), 105  
BatchJobParameters() (Java constructor), 105  
BatchJobParameters(Integer, String) (Java constructor), 105  
BatchJobParameters(Integer, String, String) (Java constructor), 105  
BatchJobUpdateParams (Java class), 109  
Bookmark (Java class), 384  
Bookmark() (Java constructor), 384  
Bookmark(String, String, String, String, Map) (Java constructor), 384  
BookmarkDataService (Java interface), 389  
BookmarkService (Java interface), 392  
build() (Java method), 301, 404  
build(List) (Java method), 304  
build(List, List) (Java method), 302  
build(ScheduleRecord) (Java method), 463  
build(ScheduleRecord, Locale) (Java method), 463  
buildFrom(MotechEvent) (Java method), 133  
byApplicationName(String) (Java method), 193  
byDateRange(DateTime, DateTime) (Java method), 75  
byExternalId(String) (Java method), 76  
byPriority(Integer) (Java method), 76  
bySourceConfiguration(String) (Java method), 193  
byStatus(AlertStatus) (Java method), 76  
byType(AlertType) (Java method), 76

## C

CALL\_FAILED (Java field), 340  
CALL\_INITIATED (Java field), 340



- CallDetailRecord (Java class), 340
- CallDetailRecord() (Java constructor), 340
- CallDetailRecord(String, String, String, String, CallDirection, String, String, String, String, Map) (Java constructor), 341
- CallDetailRecordDataService (Java interface), 349
- CallDirection (Java enum), 344
- CallInitiationException (Java class), 349
- CallInitiationException(String) (Java constructor), 349
- CallInitiationException(String, Throwable) (Java constructor), 349
- Campaign (Java class), 358
- Campaign() (Java constructor), 358
- Campaign(String, List) (Java constructor), 358
- Campaign(String, List, Period) (Java constructor), 358
- CAMPAIGN\_COMPLETED (Java field), 352
- CAMPAIGN\_MESSAGE\_RECORD\_UPDATED (Java field), 352
- CAMPAIGN\_NAME\_KEY (Java field), 352
- CAMPAIGN\_RECORD\_UPDATED (Java field), 353
- CampaignAlreadyEndedException (Java class), 371
- CampaignAlreadyEndedException(String, Throwable) (Java constructor), 371
- campaignCompleted(String, String) (Java method), 378, 380
- CampaignEnrollment (Java class), 359
- CampaignEnrollment(String, String) (Java constructor), 359
- CampaignEnrollmentDataService (Java interface), 355
- CampaignEnrollmentException (Java class), 371
- CampaignEnrollmentException(String) (Java constructor), 371
- CampaignEnrollmentRecord (Java class), 375
- CampaignEnrollmentRecord(String, String, LocalDate, CampaignEnrollmentStatus) (Java constructor), 375
- CampaignEnrollmentRecordMapper (Java class), 376
- CampaignEnrollmentsQuery (Java class), 376
- CampaignEnrollmentStatus (Java enum), 360
- CampaignJsonException (Java class), 371
- CampaignJsonException(Throwable) (Java constructor), 371
- CampaignJsonLoader (Java class), 374
- CampaignJsonLoader() (Java constructor), 374
- CampaignJsonLoader(MotechJsonReader) (Java constructor), 374
- CampaignMessage (Java class), 360
- CampaignMessage(CampaignMessageRecord) (Java constructor), 360
- CampaignMessage(String, List, List, String, Time) (Java constructor), 361
- CampaignMessageRecord (Java class), 362
- CampaignMessageRecordService (Java interface), 356
- CampaignMessageValidationException (Java class), 371
- CampaignMessageValidationException(String) (Java constructor), 371
- CampaignMessageValidationException(String, Exception) (Java constructor), 371
- campaignName() (Java method), 354
- CampaignNotFoundException (Java class), 371
- CampaignNotFoundException(String) (Java constructor), 372
- CampaignRecord (Java class), 364
- CampaignRecordService (Java interface), 356
- CampaignRequest (Java class), 354
- CampaignRequest() (Java constructor), 354
- CampaignRequest(String, String, LocalDate, Time) (Java constructor), 354
- CampaignType (Java enum), 365
- CampaignValidationException (Java class), 372
- CampaignValidationException(String) (Java constructor), 372
- CampaignValidationException(String, Exception) (Java constructor), 372
- canMakeConnection() (Java method), 146, 335
- CASE\_ACTION (Java field), 176
- CASE\_DATA\_XMLNS (Java field), 176
- CASE\_EVENT (Java field), 178
- CASE\_ID (Java field), 176
- CASE\_IDS (Java field), 176
- CASE\_NAME (Java field), 176
- CASE\_TYPE (Java field), 176
- CaseEvent (Java class), 171
- CaseEvent(MotechEvent) (Java constructor), 171
- CaseEvent(String) (Java constructor), 171
- CaseInfo (Java class), 138
- CaseJson (Java class), 140
- CaseParser (Java class), 181
- CaseParser(Class, String) (Java constructor), 181
- CaseParserException (Java class), 179
- CaseParserException(Exception, String) (Java constructor), 179
- CaseParserException(String) (Java constructor), 179
- CaseRequest (Java class), 184
- CaseRequest() (Java constructor), 185
- CaseRequest(String, String, String, String) (Java constructor), 185
- CaseResponseJson (Java class), 142
- CasesInfo (Java class), 145
- CasesInfo(List, CommcareMetadataInfo) (Java constructor), 145
- CaseTask (Java class), 142
- CaseXml (Java class), 144
- CaseXml() (Java constructor), 144
- CATEGORY\_OPTION\_COMBO (Java field), 274, 302
- changeCurrentUserPassword(String, String) (Java method), 433

- [changeMaxUploadSize\(MotechEvent\)](#) (Java method), 380
- [ChannelRequestBuilder](#) (Java class), 301
- [ChannelRequestBuilder\(BundleContext, ProgramService, StageService, TrackedEntityAttributeService, TrackedEntityService\)](#) (Java constructor), 301
- [Chapter](#) (Java class), 385
- [Chapter\(\)](#) (Java constructor), 385
- [Chapter\(String, CourseUnitState, String\)](#) (Java constructor), 385
- [Chapter\(String, CourseUnitState, String, List\)](#) (Java constructor), 385
- [Chapter\(String, CourseUnitState, String, List, Quiz\)](#) (Java constructor), 385
- [ChapterDataService](#) (Java interface), 389
- [clearLoggableEvents\(\)](#) (Java method), 308
- [CLOSE\\_ALERT\\_SUBJECT](#) (Java field), 74
- [CLOSED](#) (Java field), 85
- [closedAlerts\(\)](#) (Java method), 90
- [CloseElement](#) (Java class), 186
- [CloseTask](#) (Java class), 146
- [CloseTask\(boolean\)](#) (Java constructor), 146
- [CMSLiteException](#) (Java class), 114
- [CMSLiteException\(String\)](#) (Java constructor), 114
- [CMSLiteException\(String, Throwable\)](#) (Java constructor), 114
- [CMSLiteService](#) (Java interface), 119
- [CMSLiteServiceImpl](#) (Java class), 122
- [CodedType](#) (Java class), 213
- [CodedType\(\)](#) (Java constructor), 214
- [CodedType\(String, String\)](#) (Java constructor), 214
- [CodedType\(String, String, String\)](#) (Java constructor), 214
- [CodedTypeDataService](#) (Java interface), 259
- [CommcareAccountSettings](#) (Java class), 146
- [CommcareApplicationDataService](#) (Java interface), 193
- [CommcareApplicationJson](#) (Java class), 147
- [CommcareApplicationJson\(\)](#) (Java constructor), 147
- [CommcareApplicationJson\(String, String, String, List\)](#) (Java constructor), 147
- [CommcareAppStructureService](#) (Java interface), 193
- [CommcareAuthenticationException](#) (Java class), 179
- [CommcareAuthenticationException\(Exception, String\)](#) (Java constructor), 180
- [CommcareAuthenticationException\(String\)](#) (Java constructor), 180
- [CommcareCaseService](#) (Java interface), 194
- [CommcareConfigService](#) (Java interface), 198
- [CommcareConnectionFailureException](#) (Java class), 180
- [CommcareConnectionFailureException\(Exception, String\)](#) (Java constructor), 180
- [CommcareConnectionFailureException\(String\)](#) (Java constructor), 180
- [CommcareDataForwardingEndpoint](#) (Java class), 148
- [CommcareDataForwardingEndpoint\(\)](#) (Java constructor), 149
- [CommcareDataForwardingEndpoint\(String, String, String, String\)](#) (Java constructor), 149
- [CommcareDataForwardingEndpointService](#) (Java interface), 199
- [CommcareDataForwardingEndpointsJson](#) (Java class), 150
- [CommcareFixture](#) (Java class), 150
- [CommcareFixtureService](#) (Java interface), 200
- [CommcareFixturesJson](#) (Java class), 151
- [CommcareForm](#) (Java class), 151
- [CommcareFormBuilder](#) (Java class), 133
- [CommcareFormList](#) (Java class), 153
- [CommcareFormService](#) (Java interface), 201
- [CommcareLocation](#) (Java class), 153
- [CommcareLocationService](#) (Java interface), 202
- [CommcareLocationsJson](#) (Java class), 154
- [CommcareMetadataInfo](#) (Java class), 155
- [CommcareMetadataJson](#) (Java class), 155
- [CommcareModuleJson](#) (Java class), 156
- [CommcareRequestData](#) (Java class), 186
- [CommcareRequestData\(String, MetaElement, CaseRequest\)](#) (Java constructor), 187
- [CommcareSchemaService](#) (Java interface), 203
- [CommcareTasksNotifier](#) (Java class), 205
- [CommcareTasksNotifier\(BundleContext, CommcareDataProvider, CommcareConfigService, CommcareSchemaService\)](#) (Java constructor), 205
- [CommcareUser](#) (Java class), 156
- [CommcareUserService](#) (Java interface), 204
- [CommcareUsersJson](#) (Java class), 158
- [COMMENT](#) (Java field), 274, 302, 419
- [CommunicationProtocol](#) (Java enum), 215
- [compare\(ResourceDto, ResourceDto\)](#) (Java method), 127
- [compareTo\(Alert\)](#) (Java method), 83
- [COMPLETE\\_DATE](#) (Java field), 274, 302
- [COMPLETED](#) (Java field), 360, 383, 457
- [completedDuring\(DateTime, DateTime\)](#) (Java method), 474, 483
- [CompletedDuringCriterion](#) (Java class), 465
- [CompletedDuringCriterion\(DateTime, DateTime\)](#) (Java constructor), 465
- [CONCEPT\\_CONCEPT\\_CLASS](#) (Java field), 419
- [CONCEPT\\_DATA\\_TYPE](#) (Java field), 419
- [CONCEPT\\_NAME](#) (Java field), 419
- [CONCEPT\\_UUID](#) (Java field), 419
- [ConceptNameAlreadyInUseException](#) (Java class), 417
- [ConceptNameAlreadyInUseException\(String\)](#) (Java constructor), 417
- [Config](#) (Java class), 134, 215, 344, 507
- [CONFIG](#) (Java field), 522
- [Config\(\)](#) (Java constructor), 215

- Config(boolean, String, String, String, String) (Java constructor), 215
- Config(String, boolean, String, String, List, String, String, HttpMethod, boolean, String, boolean, List) (Java constructor), 344
- CONFIG\_BASE\_URL (Java field), 134, 176
- CONFIG\_CREATED (Java field), 178
- CONFIG\_DELETED (Java field), 178
- CONFIG\_DOMAIN (Java field), 134, 176
- CONFIG\_NAME (Java field), 134, 177
- CONFIG\_PASSWORD (Java field), 134
- CONFIG\_UPDATED (Java field), 178
- CONFIG\_USERNAME (Java field), 134
- ConfigController (Java class), 267
- configCreated(MotechEvent) (Java method), 174
- configDeleted(MotechEvent) (Java method), 174
- ConfigNotFoundException (Java class), 259, 348
- ConfigNotFoundException(String) (Java constructor), 259, 348
- ConfigNotFoundException(String, Throwable) (Java constructor), 259, 348
- ConfigProp (Java class), 509
- ConfigProp() (Java constructor), 509
- ConfigProp(String, String) (Java constructor), 509
- Configs (Java class), 135, 510
- Configs() (Java constructor), 136, 510
- Configs(List) (Java constructor), 136
- ConfigService (Java interface), 262, 350, 512
- ConfigServiceImpl (Java class), 513
- ConfigServiceImpl(SettingsFacade) (Java constructor), 513
- configUpdated(MotechEvent) (Java method), 174
- ConfigurationEventHandler (Java class), 173
- ConfigurationNotFoundException (Java class), 180
- ConfigurationNotFoundException(String) (Java constructor), 180
- configuredConvertEventToDbLog(MotechEvent, LoggableEvent) (Java method), 306
- CONFIRMED (Java field), 95
- Constants (Java class), 352, 521
- consume(String) (Java method), 266
- CONSUME\_XML\_EVENT\_BASE (Java field), 206
- CONSUME\_XML\_EVENT\_WILDCARD (Java field), 206
- ContactPoint (Java class), 217
- ContactPoint() (Java constructor), 217
- ContactPoint(CodedType) (Java constructor), 217
- ContactPoint(CodedType, String, String, String) (Java constructor), 217
- ContactPointDataService (Java interface), 259
- contains(Config) (Java method), 136
- containsAttribute(String) (Java method), 162
- containsLanguage(String) (Java method), 124
- Content (Java interface), 115
- ContentDistributionService (Java interface), 336
- ContentDistributionServiceImpl (Java class), 338
- ContentDistributionServiceImpl() (Java constructor), 338
- ContentDistributionServiceImpl(HubTopicMDSService, HubSubscriptionMDSService, HubPublisherTransactionMDSService, HubSubscriberTransactionMDSService, HubDistributionContentMDSService) (Java constructor), 338
- ContentNotFoundException (Java class), 115
- ContentNotFoundException() (Java constructor), 115
- convertToLog(MotechEvent) (Java method), 306
- CoordinateDto (Java class), 279
- copyFrom(CampaignEnrollment) (Java method), 359
- copyFrom(Enrollment) (Java method), 454
- count() (Java method), 463
- countAllSmsRecords(SmsRecordSearchCriteria) (Java method), 494
- countFindByCriteria(String, Set, String, String, Range, Set, String, String, String, String) (Java method), 506
- countFindByStatus(AlertStatus) (Java method), 78
- countJobs() (Java method), 112
- Course (Java class), 386
- Course(String, CourseUnitState, String) (Java constructor), 386
- Course(String, CourseUnitState, String, List) (Java constructor), 386
- CourseDataService (Java interface), 390
- CourseUnitMetadata (Java class), 386
- CourseUnitMetadata(String, CourseUnitState, String) (Java constructor), 386
- CourseUnitState (Java enum), 387
- create() (Java method), 198
- create(MotechEvent) (Java method), 87
- create(String, String) (Java method), 301
- create(String, String, String, AlertType, AlertStatus, int, Map) (Java method), 77, 88
- CREATE\_ALERT\_SUBJECT (Java field), 75
- CREATE\_AND\_ENROLL (Java field), 275
- CREATE\_ENTITY (Java field), 275
- CREATE\_TRACKED\_ENTITY\_INSTANCE (Java field), 302
- createActivity(ActivityRecord) (Java method), 391
- createBookmark(Bookmark) (Java method), 392
- createChapter(Chapter) (Java method), 394
- createConcept(OpenMRSConcept) (Java method), 423
- createCourse(Course) (Java method), 394
- CREATED\_NEW\_CONCEPT\_SUBJECT (Java field), 420
- CREATED\_NEW\_ENCOUNTER\_SUBJECT (Java field), 420
- CREATED\_NEW\_FACILITY\_SUBJECT (Java field), 420

- CREATED\_NEW\_OBSERVATION\_SUBJECT (Java field), 420
  - CREATED\_NEW\_PATIENT\_SUBJECT (Java field), 420
  - CREATED\_NEW\_PERSON\_SUBJECT (Java field), 420
  - CREATED\_NEW\_PROVIDER\_SUBJECT (Java field), 420
  - CREATED\_UPDATED\_OBSERVATION\_SUBJECT (Java field), 420
  - CreateElement (Java class), 187
  - CreateElement(String, String, String) (Java constructor), 187
  - createEncounter(OpenMRSEncounter) (Java method), 425
  - createEncounterType(OpenMRSEncounterType) (Java method), 425
  - createEnrollment(EnrollmentDto) (Java method), 294
  - createEvent(DhisEventDto) (Java method), 294
  - createFacility(OpenMRSFacility) (Java method), 426
  - createFromDetails(DataElementDto) (Java method), 297
  - createFromDetails(OrganisationUnitDto) (Java method), 297
  - createFromDetails(ProgramDto) (Java method), 298
  - createFromDetails(ProgramStageDto, String, boolean) (Java method), 299
  - createFromDetails(TrackedEntityAttributeDto) (Java method), 300
  - createFromDetails(TrackedEntityDto) (Java method), 301
  - CreateInstanceActionBuilder (Java class), 302
  - createLesson(Lesson) (Java method), 394
  - createNew(DailyPillRegimenRequest) (Java method), 451, 452
  - createNewDataForwardingRule(CommcareDataForwardingRuleConfig) (Java method), 199
  - createObservation(OpenMRSObservation) (Java method), 428
  - createPatient(OpenMRSPatient) (Java method), 429
  - createPerson(OpenMRSPerson) (Java method), 431
  - createPerson(String, String, DateTime, String, String, List) (Java method), 431
  - createProvider(OpenMRSPProvider) (Java method), 432
  - createQuiz(Quiz) (Java method), 394
  - CreateTask (Java class), 158
  - createTrackedEntityInstance(TrackedEntityInstanceDto) (Java method), 294
  - createUser(OpenMRSUser) (Java method), 433
  - Credential (Java class), 218
  - Credential() (Java constructor), 218
  - Credential(CodedType, String) (Java constructor), 218
  - Credential(CodedType, String, String, DateTime, DateTime, Set) (Java constructor), 219
  - CredentialDataService (Java interface), 259
  - Credentials (Java class), 320
  - Credentials(String, String) (Java constructor), 320
  - Criterion (Java enum), 79
  - Criterion (Java interface), 465
  - CRITICAL (Java field), 86
  - CRON (Java field), 365
  - CronBasedCampaign (Java class), 365
  - CronBasedCampaign() (Java constructor), 365
  - CronBasedCampaign(String, List) (Java constructor), 366
  - CronBasedCampaign(String, List, Period) (Java constructor), 366
  - CronBasedCampaignMessage (Java class), 366
  - CronBasedCampaignMessage(CampaignMessageRecord) (Java constructor), 366
  - CronBasedCampaignMessage(String) (Java constructor), 366
  - CronBasedCampaignMessage(String, List, List, String, String) (Java constructor), 366
  - CronJobScheduleParam (Java class), 110
  - CSD (Java class), 211
  - CSD() (Java constructor), 212
  - CSD(OrganizationDirectory, ServiceDirectory, FacilityDirectory, ProviderDirectory) (Java constructor), 213
  - CSD\_CLASSES (Java field), 206
  - CSD\_SCHEMA (Java field), 206
  - CSD\_TASK\_REST\_UPDATE (Java field), 206
  - CSD\_TASK\_SOAP\_UPDATE (Java field), 206
  - CSD\_UPDATE\_BASE (Java field), 206
  - CSD\_UPDATE\_CUSTOM (Java field), 206
  - CSD\_UPDATE\_SCHEDULED (Java field), 207
  - CSD\_UPDATE\_TASK (Java field), 207
  - CSDConstants (Java class), 206
  - CSDController (Java class), 266
  - CSDEndpoint (Java class), 266
  - CSDEndpointInterceptor (Java class), 266
  - CSDEndpointInterceptor() (Java constructor), 266
  - CSDEventKeys (Java class), 206
  - CSDReferenceFinder (Java class), 265
  - CSDReferenceFinder() (Java constructor), 265
  - CSDService (Java interface), 261
  - currentlyInWindow(WindowName) (Java method), 483
  - CUSTOM\_PARAMS (Java field), 522
- ## D
- DAILY\_REPEATING\_DAYS\_CRON\_EXPRESSION (Java field), 352
  - DailyPillRegimenRequest (Java class), 436
  - DailyPillRegimenRequest(String, int, int, int, List) (Java constructor), 436
  - DailyScheduleDetails (Java class), 442
  - DailyScheduleDetails() (Java constructor), 442
  - DailyScheduleDetails(int, int, int) (Java constructor), 442
  - DATA (Java field), 87, 321
  - data() (Java method), 464
  - data(Map) (Java method), 81
  - DATA\_ELEMENT (Java field), 274, 302



- DATA\_SET (Java field), 274, 302
- DATA\_UPDATER (Java field), 86
- DATA\_VALUES (Java field), 274, 303
- DATABASE\_OPERATION\_FAILED (Java field), 98
- DataElement (Java class), 267
- DataElement() (Java constructor), 268
- DataElement(String, String) (Java constructor), 268
- DataElementDataService (Java interface), 276
- DataElementDto (Java class), 280
- DataElementService (Java interface), 296
- DataValueDto (Java class), 280
- DataValueSetDto (Java class), 281
- DATE (Java field), 275
- DATE\_MODIFIED (Java field), 177
- DATE\_OF\_DEATH (Java field), 420
- DATE\_RANGE (Java field), 79
- DATE\_TIME\_PICKER\_FORMAT (Java field), 215
- DAY\_OF\_WEEK (Java field), 365
- DayOfTheWeek (Java enum), 220
- DayOfWeek (Java enum), 366
- DayOfWeekCampaign (Java class), 367
- DayOfWeekCampaign() (Java constructor), 367
- DayOfWeekCampaign(String, List, Period) (Java constructor), 367
- DayOfWeekCampaignMessage (Java class), 368
- DayOfWeekCampaignMessage(CampaignMessageRecord) (Java constructor), 368
- DayOfWeekCampaignMessage(String, List, List, String, Time, List) (Java constructor), 368
- DayOfWeekCampaignMessage(Time, List) (Java constructor), 368
- DbEventLogger (Java class), 309
- DbEventLogger(EventLogService, DefaultDbToLogConverter) (Java constructor), 309
- DbEventLoggingService (Java class), 318
- DbEventLoggingService() (Java constructor), 318
- DbEventLoggingService(AllEventMappings) (Java constructor), 318
- DbEventLoggingService(List) (Java constructor), 318
- DbEventQueryService (Java class), 319
- DbLoggableEvent (Java class), 310
- DbLoggableEvent(List, List, LogMappings) (Java constructor), 310
- deceasePatient(String, OpenMRSCConcept, Date, String) (Java method), 429
- DEFAULT\_CONCEPT\_NAME\_TYPE (Java field), 400
- DEFAULT\_LOCALE (Java field), 400
- DEFAULT\_ROWS (Java field), 124
- DefaultDbToLogConverter (Java class), 306
- DEFAULTED (Java field), 458
- DefaultFileToLogConverter (Java class), 306
- DEFAULTMENT\_DATE\_TIME (Java field), 473
- DELETE (Java field), 325
- delete() (Java method), 262
- delete(CampaignEnrollment) (Java method), 377
- delete(String) (Java method), 263–265
- delete(T) (Java method), 297
- deleteActivity(long) (Java method), 391
- deleteAll() (Java method), 263–265, 297
- deleteAllActivityForUser(String) (Java method), 391
- deleteAllBookmarksForUser(String) (Java method), 392
- deleteBookmark(long) (Java method), 393
- deleteCampaign(String) (Java method), 378, 380
- deleteChapter(long) (Java method), 394
- deleteConcept(String) (Java method), 423
- deleteConfig(String) (Java method), 136, 198
- deleteCourse(long) (Java method), 394
- DELETED\_CONCEPT\_SUBJECT (Java field), 420
- DELETED\_ENCOUNTER\_SUBJECT (Java field), 420
- DELETED\_FACILITY\_SUBJECT (Java field), 420
- DELETED\_OBSERVATION\_SUBJECT (Java field), 420
- DELETED\_PATIENT\_SUBJECT (Java field), 420
- DELETED\_PERSON\_SUBJECT (Java field), 420
- DELETED\_PROVIDER\_SUBJECT (Java field), 421
- deleteEncounter(String) (Java method), 425
- deleteEncounterType(String) (Java method), 425
- deleteFacility(String) (Java method), 427
- deleteLesson(long) (Java method), 395
- deleteObservation(String) (Java method), 428
- deletePatient(String) (Java method), 430
- deletePerson(String) (Java method), 431
- deleteProvider(String) (Java method), 432
- deleteQuiz(long) (Java method), 395
- deleteUser(String) (Java method), 433
- deliverTime() (Java method), 354
- DELIVERY\_CONFIRMED (Java field), 493, 523
- DELIVERY\_TIME (Java field), 522
- DeliveryStatus (Java enum), 493
- DESCRIPTION (Java field), 87
- description(String) (Java method), 81
- DESCRIPTION\_UPDATER (Java field), 86
- DEVICE\_LOG (Java field), 183
- DEVICE\_LOG\_EVENT (Java field), 179
- DEVICE\_REPORT\_ELEMENT (Java field), 183
- DHIS2\_DISPLAY\_NAME (Java field), 303
- DhisDataValueStatusResponse (Java class), 282
- DhisEventDto (Java class), 283
- DhisSchemaController (Java class), 304
- DhisSettingsController (Java class), 305
- DhisStatus (Java enum), 286
- DhisStatusResponse (Java class), 286
- DhisWebException (Java class), 294
- DhisWebException(String) (Java constructor), 294
- DhisWebException(String, Throwable) (Java constructor), 294
- DhisWebService (Java interface), 294
- DISPATCHED (Java field), 493, 523
- DisplayNames (Java class), 302

distribute(String) (Java method), 336, 338  
distribute(String, String, MediaType, String) (Java method), 337, 339  
DistributionServiceDelegate (Java interface), 337  
DistributionServiceDelegateImpl (Java class), 339  
DistributionServiceDelegateImpl(HttpAgent, SettingsFacade) (Java constructor), 339  
DistributionStatusLookup (Java enum), 335  
Dosage (Java class), 443  
Dosage() (Java constructor), 444  
Dosage(Time, Set) (Java constructor), 444  
DOSAGE\_ID\_KEY (Java field), 435  
DosageRequest (Java class), 437  
DosageRequest(int, int, List) (Java constructor), 437  
DosageResponse (Java class), 438  
DosageResponse(Long, Time, LocalDate, LocalDate, LocalDate, List) (Java constructor), 438  
dosageStatusKnown(Long, Long, LocalDate) (Java method), 451, 452  
due (Java field), 463  
due() (Java method), 465  
DUE\_DATE\_TIME (Java field), 473  
DUPLICATE\_JOB (Java field), 98

## E

earliest (Java field), 463  
earliest() (Java method), 465  
EARLIEST\_DATE\_TIME (Java field), 473  
editStreamContent(String, String, MultipartFile) (Java method), 128  
editStringContent(String, String, String) (Java method), 128  
ELEMENT\_NAME (Java field), 177  
EMPTY\_PERIOD (Java field), 463  
ENCOUNTER\_DATE (Java field), 421  
ENCOUNTER\_ID (Java field), 421  
ENCOUNTER\_TYPE (Java field), 421  
EndOfMilestoneListener (Java class), 489  
EndOfMilestoneListener(EnrollmentDataService) (Java constructor), 489  
EndOfWindowCriterion (Java class), 466  
EndOfWindowCriterion(WindowName, DateTime, DateTime) (Java constructor), 466  
EndpointNotSupported (Java class), 180  
EndpointNotSupported(String) (Java constructor), 180  
enroll(CampaignRequest) (Java method), 378, 380  
enroll(EnrollmentRequest) (Java method), 486, 491  
enroll(String, String, String, DateTime, DateTime, Time, Map) (Java method), 482, 490  
enroll(String, String, String, DateTime, String, DateTime, String, String) (Java method), 488, 492  
ENROLL\_IN\_PROGRAM (Java field), 275  
ENROLL\_USER\_SUBJECT (Java field), 353  
ENROLLED\_USER\_SUBJECT (Java field), 353

EnrolledUserEvent (Java class), 469  
EnrolledUserEvent(String, String, Time, DateTime, DateTime, String) (Java constructor), 469  
Enrollment (Java class), 454  
Enrollment() (Java constructor), 454  
ENROLLMENT\_DATE (Java field), 303, 473  
ENROLLMENT\_ID (Java field), 473  
ENROLLMENT\_TIME (Java field), 473  
EnrollmentAlertService (Java class), 489  
EnrollmentAlertService(MotechSchedulerService, EventRelay) (Java constructor), 489  
EnrollmentAlreadyExists (Java class), 372  
EnrollmentAlreadyExists(String, String) (Java constructor), 372  
EnrollmentAlreadyExists(String, String, Throwable) (Java constructor), 372  
EnrollmentBuilder (Java class), 456  
EnrollmentBuilder() (Java constructor), 456  
EnrollmentDataService (Java interface), 476  
EnrollmentDefaultmentService (Java class), 489  
EnrollmentDefaultmentService(MotechSchedulerService) (Java constructor), 489  
EnrollmentDto (Java class), 287  
EnrollmentNotFoundException (Java class), 372  
EnrollmentNotFoundException() (Java constructor), 372  
EnrollmentNotFoundException(String) (Java constructor), 372  
EnrollmentNotFoundException(String, Throwable) (Java constructor), 372  
EnrollmentRecord (Java class), 477  
EnrollmentRecord() (Java constructor), 477  
EnrollmentRecordMapper (Java class), 490  
EnrollmentRequest (Java class), 479  
EnrollmentRequest() (Java constructor), 480  
EnrollmentService (Java class), 377  
EnrollmentService (Java interface), 482  
EnrollmentServiceImpl (Java class), 490  
EnrollmentServiceImpl(ScheduleDataService, EnrollmentDataService, EnrollmentAlertService, EnrollmentDefaultmentService, EventRelay) (Java constructor), 490  
EnrollmentsQuery (Java class), 483  
EnrollmentsQueryService (Java class), 491  
EnrollmentsQueryService(AllEnrollments) (Java constructor), 491  
EnrollmentStatus (Java enum), 457  
EnrollmentUpdater (Java enum), 482  
ENTITY\_TYPE (Java field), 275  
Equal (Java class), 79  
Equal(String, T) (Java constructor), 79  
equalAgeAndBirthDates(OpenMRSPerson) (Java method), 411  
equalNameData(OpenMRSPerson) (Java method), 412

- `equals(Object)` (Java method), 83, 116, 118, 131, 147, 208–211, 213–215, 217, 219, 221, 223, 226–229, 232, 234, 236, 239–243, 245, 247, 250–252, 254, 255, 257, 258, 278, 280, 283, 286–288, 290–293, 341, 344, 347, 362, 364, 398, 399, 401, 406, 408, 410, 412, 415, 446, 459, 462, 464, 499, 516
  - `equalsContent(ResourceDto, String, String)` (Java method), 131
  - `equalTimes(CampaignMessageRecord)` (Java method), 362
  - `ERROR` (Java field), 286
  - `evaluate(Object)` (Java method), 79
  - `evaluateValue(T, T)` (Java method), 80
  - `EVENT_DATE` (Java field), 303
  - `EventDataKeys` (Java class), 176, 321, 472
  - `EventFlag` (Java interface), 311
  - `EventHandler` (Java class), 273
  - `EventHandler()` (Java constructor), 273
  - `EventHandler(DhisWebService, TrackedEntityInstanceMappingService)` (Java constructor), 273
  - `EventKeys` (Java class), 74, 352, 419, 434
  - `EventLog` (Java class), 307
  - `EventLog()` (Java constructor), 307
  - `EventLog(String, Map, DateTime)` (Java constructor), 307
  - `EventLogger` (Java class), 308
  - `EventLoggingService` (Java interface), 316
  - `EventLoggingServiceManager` (Java interface), 317
  - `EventLoggingServiceManagerImpl` (Java class), 319
  - `EventLogService` (Java interface), 316
  - `EventParams` (Java class), 274
  - `EventQueryService` (Java interface), 317
  - `EventSubjects` (Java class), 178, 275, 321, 474
  - `EventToLogConverter` (Java interface), 305
  - `execute(Query, InstanceSecurityRestriction)` (Java method), 475
  - `execute(String, Object, Method)` (Java method), 322
  - `execute(String, Object, Method, Credentials)` (Java method), 322
  - `execute(String, Object, Method, Map)` (Java method), 322
  - `execute(String, Object, Method, Map, Credentials)` (Java method), 323
  - `executeSync(String, Object, Method)` (Java method), 323
  - `executeSync(String, Object, Method, Credentials)` (Java method), 323
  - `executeSync(String, Object, Method, Map)` (Java method), 323
  - `executeSync(String, Object, Method, Map, Credentials)` (Java method), 324
  - `executeWithReturnTypeSync(String, Object, Method)` (Java method), 324
  - `executeWithReturnTypeSync(String, Object, Method, Integer, Long)` (Java method), 324
  - `exists(String)` (Java method), 198
  - `Extension` (Java class), 220
  - `Extension()` (Java constructor), 221
  - `Extension(String, String)` (Java constructor), 221
  - `ExtensionDataService` (Java interface), 259
  - `EXTERNAL_ID` (Java field), 79, 275, 303, 473
  - `EXTERNAL_ID_KEY` (Java field), 75, 353, 435
  - `externalId()` (Java method), 76, 354
  - `ExternalIdCriterion` (Java class), 466
  - `ExternalIdCriterion(String)` (Java constructor), 466
- ## F
- `Facility` (Java class), 221
  - `Facility()` (Java constructor), 223
  - `Facility(String, Set, Record, String)` (Java constructor), 223
  - `Facility(String, Set, Set, Record, Set, String, Set, Set, Set, Set, Set, Geocode, Set, Set)` (Java constructor), 223
  - `FACILITY_COUNTRY` (Java field), 421
  - `FACILITY_COUNTY_DISTRICT` (Java field), 421
  - `FACILITY_ID` (Java field), 421
  - `FACILITY_NAME` (Java field), 421
  - `FACILITY_REGION` (Java field), 421
  - `FACILITY_STATE_PROVINCE` (Java field), 421
  - `FacilityDataService` (Java interface), 260
  - `FacilityDirectory` (Java class), 225
  - `FacilityDirectory()` (Java constructor), 225
  - `FacilityDirectory(Set)` (Java constructor), 225
  - `FacilityOrganization` (Java class), 226
  - `FacilityOrganization()` (Java constructor), 226
  - `FacilityOrganization(String)` (Java constructor), 227
  - `FacilityOrganization(String, Set, Set)` (Java constructor), 227
  - `FacilityOrganizationDataService` (Java interface), 260
  - `FacilityOrganizationService` (Java class), 227
  - `FacilityOrganizationService()` (Java constructor), 228
  - `FacilityOrganizationService(String)` (Java constructor), 228
  - `FacilityOrganizationService(String, Set, Set, Set, String, Set)` (Java constructor), 228
  - `FacilityOrganizationServiceDataService` (Java interface), 260
  - `FacilityService` (Java interface), 263
  - `FAILED_FORM_MESSAGE` (Java field), 177
  - `FAILURE` (Java field), 335
  - `FAILURE_CONFIRMED` (Java field), 493, 523
  - `FAILURE_COUNT` (Java field), 522
  - `fetch(AllEnrollments)` (Java method), 465–469
  - `fetchAndUpdate(String)` (Java method), 262
  - `fetchAndUpdateUsingREST(String)` (Java method), 262

fetchAndUpdateUsingSOAP(String, DateTime) (Java method), 262

FIELD\_VALUES (Java field), 177

FILE\_READING\_WRTING\_FAILED (Java field), 98

FileEventLogger (Java class), 309

FileEventLogger(DefaultFileToLogConverter) (Java constructor), 309

FileEventLogger(List, List, DefaultFileToLogConverter) (Java constructor), 310

FileEventLoggingService (Java class), 320

FileEventLoggingService() (Java constructor), 320

FileEventLoggingService(List) (Java constructor), 320

filter(GridSettings, List) (Java method), 132

filter(List) (Java method), 465–469

findAll() (Java method), 297

findAllSmsRecords() (Java method), 494

findAllSmsRecords(SmsRecordSearchCriteria) (Java method), 494

findAppointmentByExternalIdAndStatus(String, AppointmentStatus) (Java method), 95

findAppointmentById(String) (Java method), 96

findAppointmentByStatus(AppointmentStatus) (Java method), 96

findAppointmentsByExternalId(String) (Java method), 96

findAppointmentsByExternalId(String, AppointmentStatus) (Java method), 97

findBookmarkById(Long) (Java method), 389

findBookmarksForUser(String) (Java method), 389

findBookmarksForUserParams(String, QueryParams) (Java method), 389

findByAlertType(AlertType) (Java method), 78

findByCampaignName(String) (Java method), 355

findByCriteria(String, Set, String, String, Range, Set, String, String, String, String, QueryParams) (Java method), 506

findByCriteria(String, String, DateTime, AppointmentStatus, AppointmentStatus) (Java method), 95

findByCurrentMilestone(String) (Java method), 474

findByDateTime(Range) (Java method), 78

findByDistStatus(Integer) (Java method), 333

findByEntityID(String) (Java method), 260, 261

findByExternalId(String) (Java method), 78, 301, 355, 442, 474, 476

findByExternalIdAndCampaignName(String, String) (Java method), 355

findByExternalIdScheduleNameAndStatus(String, String, EnrollmentStatus) (Java method), 476

findByExternalName(String) (Java method), 277

findById(Long) (Java method), 476

findById(String) (Java method), 297–301

findByJobId(Integer) (Java method), 106

findByJobName(String) (Java method), 106

findByLanguageAndName(String, String) (Java method), 123

findByMetadataProperty(String, String) (Java method), 474

findByMilestoneName(String) (Java method), 477

findByMotechCallId(String) (Java method), 349

findByName(String) (Java method), 276, 297, 356, 357, 477

findByNameAndType(CampaignType, String) (Java method), 356

findByPriority(Long) (Java method), 78

findByProviderCallId(String) (Java method), 349

findByProviderCallIdAndCallStatus(String, String) (Java method), 349

findByRegistration(Boolean) (Java method), 276

findByRegistration(boolean) (Java method), 298

findBySchedule(List) (Java method), 475

findByScheduleName(Set) (Java method), 477

findByStatus(AlertStatus) (Java method), 78

findByStatus(CampaignEnrollmentStatus) (Java method), 356

findByStatus(EnrollmentStatus) (Java method), 475, 477

findBySubject(String) (Java method), 316

findBySubStatus(Integer) (Java method), 334

findByTopicUrl(String) (Java method), 334

findByUuid(String) (Java method), 276, 277

findChapterById(Long) (Java method), 390

findChapterByName(String) (Java method), 390

findCourseById(Long) (Java method), 390

findCourseByName(String) (Java method), 390

findLessonById(Long) (Java method), 390

findLessonByName(String) (Java method), 390

findObservation(String, String) (Java method), 428

findObservations(String, String) (Java method), 428

findParentEntity(Object) (Java method), 265

findPubTransactionByTopicId(Integer) (Java method), 333

findQuizById(Long) (Java method), 391

findQuizByName(String) (Java method), 391

findRecordById(Long) (Java method), 389

findRecordsForUser(String) (Java method), 389

findRecordsForUserByState(String, ActivityState) (Java method), 389

findReferences(AbstractID) (Java method), 265

findReferences(AbstractID, Class, Field) (Java method), 265

findReferencingFields(Class) (Java method), 265

findSubByCallbackUrl(String) (Java method), 334

findSubByCallbackUrlAndTopicId(String, Integer) (Java method), 334

findSubByTopicId(Integer) (Java method), 334

findSubTransBySubId(Integer) (Java method), 334

firstValue() (Java method), 168

FORM (Java field), 183



FORM\_DATA\_ELEMENT (Java field), 183  
 FORM\_ID (Java field), 177  
 FORM\_STUB\_EVENT (Java field), 179  
 FORM\_STUB\_FAIL\_EVENT (Java field), 179  
 FormAdapter (Java class), 182  
 FormListRequest (Java class), 188  
 FormNode (Java interface), 159  
 FORMS\_EVENT (Java field), 179  
 FORMS\_FAIL\_EVENT (Java field), 179  
 FormSchemaJson (Java class), 159  
 FormSchemaQuestionJson (Java class), 160  
 FormSchemaQuestionOptionJson (Java class), 161  
 FormStubJson (Java class), 161  
 FormValueAttribute (Java class), 162  
 FormValueAttribute(String) (Java constructor), 162  
 FormValueElement (Java class), 162  
 FRIDAY (Java field), 220  
 Friday (Java field), 367  
 fromCaseXml(CaseXml, String) (Java method), 171  
 fromDate() (Java method), 76  
 fromMilestone(Milestone, DateTime) (Java method), 459  
 fulfillCurrentMilestone(DateTime) (Java method), 454  
 fulfillCurrentMilestone(Enrollment, DateTime) (Java method), 482, 490  
 fulfillCurrentMilestone(String, String, LocalDate) (Java method), 487, 491  
 fulfillCurrentMilestone(String, String, LocalDate, Time) (Java method), 486, 491  
 FullFormEvent (Java class), 174  
 FullFormEvent(FormValueElement, String, String) (Java constructor), 174  
 FullFormFailureEvent (Java class), 175  
 FullFormFailureEvent(String, String) (Java constructor), 175  
 FullFormParser (Java class), 183  
 FullFormParser(String) (Java constructor), 184  
 FullFormParserException (Java class), 180  
 FullFormParserException(Exception, String) (Java constructor), 181  
 FullFormParserException(String) (Java constructor), 181

## G

generate() (Java method), 417  
 GENERATED\_MESSAGE\_KEY (Java field), 353  
 GenericCrudService (Java interface), 297  
 Geocode (Java class), 229  
 Geocode() (Java constructor), 229  
 Geocode(double, double) (Java constructor), 229  
 Geocode(double, double, double, String) (Java constructor), 229  
 GeocodeDataService (Java interface), 260  
 GET (Java field), 325, 347  
 get(Long) (Java method), 77, 89  
 get(String) (Java method), 77, 89  
 getAccountConfig() (Java method), 134  
 getAction() (Java method), 144, 171  
 getActivityById(long) (Java method), 391  
 getAddress() (Java method), 412  
 getAddresses() (Java method), 223, 236, 243, 254  
 getAddressLines() (Java method), 209  
 getAge() (Java method), 412  
 getAlerts() (Java method), 458, 461  
 getAlertTimings() (Java method), 485  
 getAlertTimings(Enrollment) (Java method), 489  
 getAlertTimings(EnrollmentRequest) (Java method), 487, 492  
 getAlertTimings(String, String, String, DateTime, DateTime, Time) (Java method), 482, 490  
 getAlertType() (Java method), 83  
 getAll() (Java method), 81  
 getAllActivityForUser(String) (Java method), 392  
 getAllApplications() (Java method), 193  
 getAllApplications(String) (Java method), 193  
 getAllBookmarksForUser(String) (Java method), 393  
 getAllCampaignRecords() (Java method), 378, 380  
 getAllCaseTypes() (Java method), 203  
 getAllCaseTypes(String) (Java method), 203  
 getAllChapters() (Java method), 395  
 getAllConcepts() (Java method), 424  
 getAllContents() (Java method), 120, 122  
 getAllCourses() (Java method), 395  
 getAllDataForwardingEndpoints(Config) (Java method), 199  
 getAllElements(String) (Java method), 163  
 getAllElements(String, List) (Java method), 163  
 getAllEventMappings() (Java method), 318  
 getAllEventsByParameter(String, String) (Java method), 317, 319  
 getAllEventsBySubject(String) (Java method), 317, 319  
 getAllEventsBySubjectAndParameter(String, String, String) (Java method), 318, 319  
 getAllFacilities() (Java method), 427  
 getAllFormSchemas() (Java method), 204  
 getAllFormSchemas(String) (Java method), 203  
 getAllLanguages() (Java method), 128  
 getAllLessons() (Java method), 395  
 getAllQuizzes() (Java method), 395  
 getAllSchedules() (Java method), 487, 492  
 getAllSchedules(String) (Java method), 476  
 getAllUsers() (Java method), 434  
 getAltitude() (Java method), 230  
 getAnswer() (Java method), 388  
 getApiKey() (Java method), 144, 172  
 getApplication() (Java method), 99, 326  
 getApplicationName() (Java method), 147  
 getApplications() (Java method), 138  
 getAppointment(String) (Java method), 97  
 getAppointmentDate() (Java method), 91, 93

`getAppointmentId()` (Java method), 93  
`getApptId()` (Java method), 91  
`getAppVersion()` (Java method), 188  
`getAssigningAuthorityName()` (Java method), 241  
`getAttribute()` (Java method), 278  
`getAttributeOptionCombo()` (Java method), 281  
`getAttributes()` (Java method), 163, 269, 287, 293, 304, 412  
`getBaseUrl()` (Java method), 133, 198  
`getBatchException()` (Java method), 103  
`getBatchJobDtoList()` (Java method), 109  
`getBatchJobId()` (Java method), 105  
`getBatchJobStatusId()` (Java method), 104  
`getBeginEffectiveDate()` (Java method), 234  
`getBeginningHour()` (Java method), 234  
`getBirthDateEstimated()` (Java method), 412  
`getBookmarkById(long)` (Java method), 393  
`getBufferOverDosageTimeInMinutes()` (Java method), 436, 441, 443  
`getByLastModification(DateTime)` (Java method), 262  
`getByName(String)` (Java method), 136, 198  
`getCallbackUrl()` (Java method), 331  
`getCallDirection()` (Java method), 341  
`getCallStatus()` (Java method), 341  
`getCampaign()` (Java method), 361, 362  
`getCampaignMessage(CampaignMessageRecord)` (Java method), 357, 358, 366–368, 370  
`getCampaignName()` (Java method), 359, 375  
`getCampaignRecord(String)` (Java method), 378, 380  
`getCampaignTimings(String, String, DateTime, DateTime)` (Java method), 378, 381  
`getCampaignType()` (Java method), 364  
`getCaseAction()` (Java method), 181  
`getCaseByCaseId(String)` (Java method), 194  
`getCaseByCaseId(String, String)` (Java method), 194  
`getCaseDataXmlns()` (Java method), 144, 172  
`getCaseId()` (Java method), 138, 140, 142, 144, 172, 185, 191  
`getCaseIds()` (Java method), 161  
`getCaseInfoList()` (Java method), 146  
`getCaseName()` (Java method), 138, 144, 158, 170, 172, 188  
`getCaseProperties()` (Java method), 156  
`getCases()` (Java method), 142  
`getCases(Integer, Integer)` (Java method), 194  
`getCases(Integer, Integer, String)` (Java method), 194  
`getCasesByCasesNameAndTimeWithMetadata(String, String, String, Integer, Integer)` (Java method), 195  
`getCasesByCasesNameAndTimeWithMetadata(String, String, String, Integer, Integer, String)` (Java method), 194  
`getCasesByCasesNameWithMetadata(String, Integer, Integer)` (Java method), 195  
`getCasesByCasesNameWithMetadata(String, Integer, Integer, String)` (Java method), 195  
`getCasesByCasesTimeWithMetadata(String, String, Integer, Integer)` (Java method), 196  
`getCasesByCasesTimeWithMetadata(String, String, Integer, Integer, String)` (Java method), 195  
`getCasesByType(String, Integer, Integer)` (Java method), 196  
`getCasesByType(String, Integer, Integer, String)` (Java method), 196  
`getCasesByUserId(String, Integer, Integer)` (Java method), 196  
`getCasesByUserId(String, Integer, Integer, String)` (Java method), 196  
`getCasesByUserIdAndType(String, String, Integer, Integer)` (Java method), 197  
`getCasesByUserIdAndType(String, String, Integer, Integer, String)` (Java method), 196  
`getCasesWithMetadata(Integer, Integer)` (Java method), 197  
`getCasesWithMetadata(Integer, Integer, String)` (Java method), 197  
`getCaseType()` (Java method), 138, 144, 156, 158, 170, 172, 188, 191  
`getCategoryOptionCombo()` (Java method), 280, 281  
`getCcCase()` (Java method), 187  
`getCertificate()` (Java method), 217  
`getChapterById(long)` (Java method), 395  
`getChapterByName(String)` (Java method), 396  
`getChapterIdentifier()` (Java method), 384  
`getChapterName()` (Java method), 382  
`getChapters()` (Java method), 386  
`getChecksum()` (Java method), 116  
`getChildElement(String)` (Java method), 163  
`getChildElements(String)` (Java method), 163  
`getCloseElement()` (Java method), 185  
`getCloseTask()` (Java method), 142  
`getCode()` (Java method), 100, 214, 241, 278, 279, 327  
`getCodedType()` (Java method), 217, 219, 257  
`getCodedTypes()` (Java method), 223, 236, 248  
`getCodingScheme()` (Java method), 214  
`getCommcareAppId()` (Java method), 148  
`getCommcareBaseUrl()` (Java method), 146, 168  
`getCommcareDomain()` (Java method), 147, 168  
`getCommcareFixtureById(String)` (Java method), 200  
`getCommcareFixtureById(String, String)` (Java method), 200  
`getCommcareLocationById(String)` (Java method), 202  
`getCommcareLocationById(String, String)` (Java method), 202  
`getCommcareLocations(Integer, Integer)` (Java method), 203  
`getCommcareLocations(Integer, Integer, String)` (Java method), 203

[getCommcareUserById\(String\) \(Java method\), 204](#)  
[getCommcareUserById\(String, String\) \(Java method\), 204](#)  
[getCommcareUsers\(Integer, Integer\) \(Java method\), 204](#)  
[getCommcareUsers\(Integer, Integer, String\) \(Java method\), 204](#)  
[getComment\(\) \(Java method\), 280, 281](#)  
[getCommonNames\(\) \(Java method\), 232, 245](#)  
[getCommunicationProtocol\(\) \(Java method\), 215](#)  
[getCompletedActivityForUser\(String\) \(Java method\), 392](#)  
[getCompleteDate\(\) \(Java method\), 282](#)  
[getCompletionTime\(\) \(Java method\), 382](#)  
[getComponent\(\) \(Java method\), 210](#)  
[getConceptByUuid\(String\) \(Java method\), 424](#)  
[getConceptClass\(\) \(Java method\), 399](#)  
[getConceptName\(\) \(Java method\), 408](#)  
[getConceptNameType\(\) \(Java method\), 401](#)  
[getConcepts\(int, int\) \(Java method\), 424](#)  
[getConfig\(\) \(Java method\), 495, 499, 501, 516](#)  
[getConfig\(String\) \(Java method\), 262, 350, 510, 512, 514](#)  
[getConfigList\(\) \(Java method\), 512, 514](#)  
[getConfigName\(\) \(Java method\), 138, 148, 150, 151, 172, 341](#)  
[getConfigOrDefault\(String\) \(Java method\), 510, 512, 514](#)  
[getConfigs\(\) \(Java method\), 136, 198, 262, 267, 511, 512, 514](#)  
[getContact\(\) \(Java method\), 239](#)  
[getContactPoints\(\) \(Java method\), 223, 236, 243, 254](#)  
[getContacts\(\) \(Java method\), 223, 236](#)  
[getContent\(\) \(Java method\), 116, 329, 386](#)  
[getContent\(String\) \(Java method\), 337, 339](#)  
[getContent\(String, String, String\) \(Java method\), 128](#)  
[getContentId\(\) \(Java method\), 330](#)  
[getContents\(GridSettings\) \(Java method\), 129](#)  
[getContentType\(\) \(Java method\), 116, 329](#)  
[getContentType\(Content\) \(Java method\), 132](#)  
[getCoordinate\(\) \(Java method\), 283](#)  
[getCoordinateSystem\(\) \(Java method\), 230](#)  
[getCount\(\) \(Java method\), 453, 505](#)  
[getCountry\(\) \(Java method\), 406](#)  
[getCountyDistrict\(\) \(Java method\), 406](#)  
[getCourseById\(long\) \(Java method\), 396](#)  
[getCourseByName\(String\) \(Java method\), 396](#)  
[getCourseIdentifier\(\) \(Java method\), 384](#)  
[getCourseName\(\) \(Java method\), 382](#)  
[getCreated\(\) \(Java method\), 255, 279, 293](#)  
[getCreatedAt\(\) \(Java method\), 153](#)  
[getCreatedBy\(\) \(Java method\), 107](#)  
[getCreateElement\(\) \(Java method\), 185](#)  
[getCreateTask\(\) \(Java method\), 143](#)  
[getCreateTime\(\) \(Java method\), 107](#)  
[getCreationDate\(\) \(Java method\), 207](#)  
[getCreator\(\) \(Java method\), 402](#)  
[getCredentialIssueDate\(\) \(Java method\), 219](#)  
[getCredentialRenewalDate\(\) \(Java method\), 219](#)  
[getCredentials\(\) \(Java method\), 237, 248](#)  
[getCriteria\(\) \(Java method\), 376, 483](#)  
[getCron\(\) \(Java method\), 362, 366](#)  
[getCronExpression\(\) \(Java method\), 104, 107, 110](#)  
[getCSD\(\) \(Java method\), 231, 262](#)  
[getCurrentMilestoneName\(\) \(Java method\), 454, 477](#)  
[getCurrentMilestoneStartDate\(\) \(Java method\), 454](#)  
[getCurrentTimestamp\(\) \(Java method\), 341](#)  
[getCurrentWindowAsOf\(DateTime\) \(Java method\), 454](#)  
[getCustomParams\(\) \(Java method\), 516](#)  
[getData\(\) \(Java method\), 83, 458](#)  
[getDataElement\(\) \(Java method\), 280, 291](#)  
[getDataElementByHref\(String\) \(Java method\), 295](#)  
[getDataElements\(\) \(Java method\), 270, 295, 304](#)  
[getDataForwardingEndpoints\(Integer, Integer, Config\) \(Java method\), 200](#)  
[getDataSet\(\) \(Java method\), 282](#)  
[getDataType\(\) \(Java method\), 399](#)  
[getDataValueCount\(\) \(Java method\), 283](#)  
[getDataValues\(\) \(Java method\), 282, 284](#)  
[getDataXmlns\(\) \(Java method\), 185](#)  
[getDate\(\) \(Java method\), 111, 358, 362, 402, 408](#)  
[getDateClosed\(\) \(Java method\), 138, 140](#)  
[getDateModified\(\) \(Java method\), 141, 143, 144, 172, 185](#)  
[getDateOfBirth\(\) \(Java method\), 243, 412](#)  
[getDateOfEnrollment\(\) \(Java method\), 287](#)  
[getDateOfIncident\(\) \(Java method\), 287](#)  
[getDateOpened\(\) \(Java method\), 138, 170](#)  
[getDateTime\(\) \(Java method\), 83](#)  
[getDateTimeInMillis\(\) \(Java method\), 83](#)  
[getDaysOfTheWeek\(\) \(Java method\), 234](#)  
[getDaysOfWeek\(\) \(Java method\), 368](#)  
[getDbEventLoggers\(\) \(Java method\), 318](#)  
[getDeathDate\(\) \(Java method\), 412](#)  
[getDefault\(\) \(Java method\), 136, 198](#)  
[getDefaultConfig\(\) \(Java method\), 511, 513, 514](#)  
[getDefaultConfigName\(\) \(Java method\), 137, 511](#)  
[getDefaultDbEventLogger\(\) \(Java method\), 318](#)  
[getDefaultDbToLogConverter\(\) \(Java method\), 318](#)  
[getDefaultmentDateTime\(\) \(Java method\), 459](#)  
[getDefaultPhoneNumber\(\) \(Java method\), 157](#)  
[getDeleted\(\) \(Java method\), 288](#)  
[getDeliverTime\(\) \(Java method\), 359](#)  
[getDeliveryStatus\(\) \(Java method\), 495, 499](#)  
[getDeliveryStatuses\(\) \(Java method\), 501](#)  
[getDeliveryTime\(\) \(Java method\), 517](#)  
[getDemographic\(\) \(Java method\), 248](#)  
[getDependentObservations\(\) \(Java method\), 408](#)  
[getDescription\(\) \(Java method\), 83, 283, 405](#)  
[getDhis2Name\(\) \(Java method\), 272](#)  
[getDhis2Uuid\(\) \(Java method\), 272](#)  
[getDisplay\(\) \(Java method\), 399](#)  
[getDisplayName\(\) \(Java method\), 278](#)

getDistributionServiceDelegate() (Java method), 339  
getDistributionStatusCode() (Java method), 329  
getDistributionStatusId() (Java method), 329  
getDomain() (Java method), 133, 138, 141, 149, 153  
getDosage(Long) (Java method), 448  
getDosageHour() (Java method), 438  
getDosageId() (Java method), 438  
getDosageMinute() (Java method), 438  
getDosageRequests() (Java method), 436  
getDosages() (Java method), 441, 448  
getDosageTime() (Java method), 444  
getDueDate() (Java method), 284  
getDueDateTime() (Java method), 459  
getDueWindowAlertTimings() (Java method), 485  
getDuration() (Java method), 462  
getEarliestDateTime() (Java method), 459  
getEarliestWindowAlertTimings() (Java method), 486  
getElement(String) (Java method), 164  
getElement(String, List) (Java method), 164  
getElementByAttribute(String, String) (Java method), 164  
getElementByAttribute(String, String, List) (Java method), 164  
getElementName() (Java method), 164  
getElements() (Java method), 165  
getElementsByAttribute(String, String) (Java method), 165  
getElementsByAttribute(String, String, List) (Java method), 165  
getElementsByAttribute(String, String, List, boolean) (Java method), 165  
getEmail() (Java method), 157  
getEncounterByUuid(String) (Java method), 425  
getEncounterId() (Java method), 402  
getEncountersByEncounterType(String, String) (Java method), 426  
getEncounterType() (Java method), 402  
getEncounterTypeByUuid(String) (Java method), 426  
getEndDate() (Java method), 438–440, 444, 446  
getEndEffectiveDate() (Java method), 234  
getEndingHour() (Java method), 234  
getEndKey() (Java method), 311  
getEndOfWindowForCurrentMilestone(WindowName) (Java method), 454  
getEndValue() (Java method), 311  
getEnrolledOn() (Java method), 455  
getEnrollment() (Java method), 284  
getEnrollment(String, String) (Java method), 487, 492  
getEnrollmentDate() (Java method), 469  
getEnrollmentDateTime() (Java method), 477, 480  
getEnrollmentId() (Java method), 470  
getEnrollmentStatus() (Java method), 284  
getEnrollmentTime() (Java method), 469  
getEntityID() (Java method), 208, 211, 258  
getEquipment() (Java method), 217  
getError() (Java method), 101, 327  
getErrorCode() (Java method), 99, 101, 326, 327, 348  
getErrorMessage() (Java method), 99, 102, 326, 327, 495, 499, 502  
getErrorMessageDetails() (Java method), 102, 327  
getEvent() (Java method), 284  
getEventData() (Java method), 284  
getEventStrategy() (Java method), 135, 168  
getEventSubjects() (Java method), 314  
getExcludeLastFooter() (Java method), 507  
getExcludes() (Java method), 314  
getExclusions() (Java method), 312  
getExtensions() (Java method), 219, 223, 227, 228, 237, 248, 251, 252, 254, 257  
getExternalId() (Java method), 83, 91, 93, 153, 359, 375, 382, 384, 436, 441, 448, 455, 469–472, 478, 480  
getExternalName() (Java method), 272  
getFacilities() (Java method), 226  
getFacilities(int, int) (Java method), 427  
getFacilities(String) (Java method), 427  
getFacility() (Java method), 402, 410  
getFacilityByEntityID(String) (Java method), 263  
getFacilityByUuid(String) (Java method), 427  
getFacilityDirectory() (Java method), 213  
getFacilityId() (Java method), 406  
getFacilityOrganizations() (Java method), 224  
getFacilityOrganizationSet() (Java method), 223  
getFailureCount() (Java method), 517  
getFields() (Java method), 150  
getFieldValues() (Java method), 139, 144, 170, 172  
getFilters() (Java method), 76  
getFirstMilestone() (Java method), 462  
getFirstName() (Java method), 157, 412  
getFixtures(Integer, Integer) (Java method), 201  
getFixtures(Integer, Integer, String) (Java method), 201  
getFixtureType() (Java method), 150  
getFlags() (Java method), 314  
getForename() (Java method), 232, 245  
getForm() (Java method), 151  
getFormats() (Java method), 361, 362  
getFormId() (Java method), 161  
getFormName() (Java method), 160  
getFormNames() (Java method), 160  
getFormSchemas() (Java method), 156  
getFreeBusyURI() (Java method), 228, 252  
getFrom() (Java method), 342  
getFromStatus() (Java method), 93  
getFulfillmentDateTime() (Java method), 460  
getFulfillments() (Java method), 455  
getFullName() (Java method), 412  
getGender() (Java method), 243, 412  
getGeocode() (Java method), 224



getGroups() (Java method), 157  
 getHonorific() (Java method), 232, 245  
 getHostName() (Java method), 99, 326  
 getHref() (Java method), 279  
 getHttpAgentImpl() (Java method), 339, 340  
 getHttpStatus() (Java method), 100, 103, 327, 328  
 getHubBaseUrl() (Java method), 335, 336  
 getHubDistributionStatusId() (Java method), 331  
 getHubException() (Java method), 328  
 getHubSubscriptionId() (Java method), 331  
 getHubSubscriptionStatusId() (Java method), 331  
 getHubTopicId() (Java method), 330, 331  
 getId() (Java method), 83, 149–151, 153, 157, 207, 279, 342, 359, 362, 364, 399, 403, 406, 409, 412, 444, 448, 455  
 getIdentifier() (Java method), 414  
 getIgnored() (Java method), 288  
 getIgnoredStatusFields() (Java method), 344  
 getImportCount() (Java method), 286  
 getImportCountDto() (Java method), 283  
 getImported() (Java method), 288  
 getIncludes() (Java method), 314  
 getInclusions() (Java method), 313  
 getIndex() (Java method), 453  
 getIndexElement() (Java method), 185  
 getIndexNodeName() (Java method), 191  
 getIndexTask() (Java method), 143  
 getIndices() (Java method), 139, 141, 167  
 getInstanceID() (Java method), 192  
 getInterval() (Java method), 453  
 getIssuingAuthority() (Java method), 219  
 getJobContent() (Java method), 104  
 getJobExecutionHistory(String) (Java method), 113  
 getJobExecutionHistoryList() (Java method), 111  
 getJobId() (Java method), 107, 470  
 getJobName() (Java method), 104, 107, 109, 110  
 getJsonExtraParamsList() (Java method), 345  
 getKey() (Java method), 373  
 getKeyValuePairsPresent() (Java method), 315  
 getLabel() (Java method), 161  
 getLang() (Java method), 242, 245  
 getLanguage() (Java method), 115, 116, 118  
 getLanguages() (Java method), 124, 131, 224, 228, 237, 248, 253, 361, 362  
 getLastFulfilledDate() (Java method), 455  
 getLastModified() (Java method), 154, 215, 231  
 getLastName() (Java method), 157, 412  
 getLastUpdated() (Java method), 107, 279  
 getLastUpdatedBy() (Java method), 107  
 getLateDateTime() (Java method), 460  
 getLatestBookmarkByUserId(String) (Java method), 393  
 getLatestCampaignMessage(String, String) (Java method), 378, 381  
 getLatestEncounterByPatientMotechId(String, String) (Java method), 426  
 getLateWindowAlertTimings() (Java method), 486  
 getLatitude() (Java method), 154, 230, 280  
 getLeaseSeconds() (Java method), 331  
 getLessonById(long) (Java method), 396  
 getLessonByName(String) (Java method), 396  
 getLessonIdentifier() (Java method), 384  
 getLessonName() (Java method), 382  
 getLessons() (Java method), 385  
 getLimit() (Java method), 155  
 getListOfJobs() (Java method), 112  
 getLocale() (Java method), 401  
 getLocationData() (Java method), 154  
 getLocationId() (Java method), 154  
 getLocationType() (Java method), 154  
 getLoggableEvents() (Java method), 308  
 getLoggedEventSubjects() (Java method), 316, 318, 320  
 getLongitude() (Java method), 154, 230, 280  
 getMappings() (Java method), 310, 313, 314  
 getMaxDuration() (Java method), 358, 364  
 getMaximumDuration() (Java method), 458  
 getMaxRetries() (Java method), 507  
 getMaxWindowAlertTimings() (Java method), 486  
 getMd5() (Java method), 151  
 getMedicineRequests() (Java method), 437  
 getMedicines() (Java method), 439, 444  
 getMessage() (Java method), 100, 175, 327, 517  
 getMessageCampaignsJsonFile() (Java method), 374  
 getMessageContent() (Java method), 495, 499, 502  
 getMessageKey() (Java method), 361, 362, 373  
 getMessageNature() (Java method), 192  
 getMessages() (Java method), 358, 364  
 getMessageText() (Java method), 192  
 getMessageType() (Java method), 362  
 getMeta() (Java method), 150, 151, 153, 154, 158, 187  
 getMetadata() (Java method), 115, 116, 118, 138, 142, 151, 455, 478, 480  
 getMetadataInfo() (Java method), 146  
 getMiddleName() (Java method), 412  
 getMilestone() (Java method), 461  
 getMilestone(String) (Java method), 462  
 getMilestoneAlert() (Java method), 471  
 getMilestoneData() (Java method), 471  
 getMilestoneName() (Java method), 460  
 getMilestones() (Java method), 462  
 getMilestoneWindow() (Java method), 453  
 getMilestoneWindow(WindowName) (Java method), 458  
 getMilestoneWindows() (Java method), 458  
 getModificationDate() (Java method), 207  
 getModifications(GetModificationsRequest) (Java method), 266  
 GetModificationsRequest (Java class), 230  
 GetModificationsResponse (Java class), 231

getModifiedAfter(DateTime) (Java method), 263–265  
getModules() (Java method), 148  
getMotechCallId() (Java method), 342  
getMotechId() (Java method), 410, 495, 499, 502, 517  
getMotechTimestamp() (Java method), 342  
getName() (Java method), 84, 115, 116, 118, 124, 131, 135, 154, 268–271, 279, 345, 347, 358, 361, 362, 364, 386, 398, 399, 401, 405, 406, 439, 440, 446, 458, 461, 462, 507, 509  
getNames() (Java method), 228, 243, 253, 399  
getNextCampaignMessage(String, String) (Java method), 378, 381  
getNextMilestoneName(String) (Java method), 462  
getNextPage() (Java method), 289  
getNextPageQueryString() (Java method), 155, 156  
getNotificationTime() (Java method), 330  
getNumber() (Java method), 219  
getObjects() (Java method), 150, 151, 153, 154, 158  
getObservationByUuid(String) (Java method), 428  
getObservationId() (Java method), 409  
getObservations() (Java method), 403  
getOffset() (Java method), 155, 156, 453  
getOperatingHours() (Java method), 224, 228, 251, 253  
getOptions() (Java method), 160  
getOrganisationUnitByHref(String) (Java method), 295  
getOrganisationUnits() (Java method), 290, 295  
getOrganization() (Java method), 253  
getOrganizationByEntityID(String) (Java method), 264  
getOrganizationDirectory() (Java method), 213  
getOrganizations() (Java method), 240  
getOrgUnit() (Java method), 281, 282, 284, 293  
getOrgUnitName() (Java method), 284  
getOrgUnits() (Java method), 305  
getOtherIDs() (Java method), 224, 237, 248  
getOtherNames() (Java method), 224, 233, 237, 245  
getOutgoingCallMethod() (Java method), 345  
getOutgoingCallUriTemplate() (Java method), 345  
getOwnerId() (Java method), 139, 144, 159, 170, 172, 188  
getPage() (Java method), 124, 132, 289, 498  
getPageCount() (Java method), 289  
getPageNumber() (Java method), 188  
getPager() (Java method), 289  
getPageSize() (Java method), 189  
getParameterName() (Java method), 105  
getParameters() (Java method), 107, 307  
getParameterValue() (Java method), 105  
getParams() (Java method), 373  
getParamsMap() (Java method), 109, 110  
getParent() (Java method), 154, 237  
getParentOrganization() (Java method), 237  
getPassPercentage() (Java method), 388  
getPassword() (Java method), 134, 147, 168, 298, 320, 345  
getPatient() (Java method), 403  
getPatientByMotechId(String) (Java method), 430  
getPatientByUuid(String) (Java method), 430  
getPatientId() (Java method), 409, 410  
getPeriod() (Java method), 281, 282, 461  
getPeriodDays() (Java method), 215  
getPeriodHours() (Java method), 216  
getPeriodMinutes() (Java method), 216  
getPerson() (Java method), 239, 410, 414, 415  
getPersonByUuid(String) (Java method), 432  
getPersonId() (Java method), 412  
getPhoneNumber() (Java method), 496, 499, 502  
getPhoneNumbers() (Java method), 157  
getPillRegimen(String) (Java method), 451, 452  
getPillRegimenId() (Java method), 441  
getPillWindowInHours() (Java method), 437, 443  
getPreferredAlertTime() (Java method), 455, 469, 478, 480  
getPreferredName() (Java method), 413  
getPreviousPageQueryString() (Java method), 155, 156  
getPrevPage() (Java method), 289  
getPrimaryCriterion() (Java method), 376, 483  
getPrimaryName() (Java method), 224, 237  
getPriority() (Java method), 84  
getProgram() (Java method), 270, 284, 287, 292  
getProgramByHref(String) (Java method), 295  
getPrograms() (Java method), 296, 305  
getProgramStage() (Java method), 284, 291  
getProgramStageByHref(String) (Java method), 295  
getProgramStageDataElements() (Java method), 292  
getProgramStages() (Java method), 290, 295  
getProgramTrackedEntityAttributes() (Java method), 290  
getProgress() (Java method), 384  
getProperties() (Java method), 141  
getProps() (Java method), 507  
getProvider() (Java method), 403  
getProviderByEntityID(String) (Java method), 264  
getProviderByUuid(String) (Java method), 432  
getProviderCallId() (Java method), 342  
getProviderDirectory() (Java method), 213  
getProviderEntityID() (Java method), 239  
getProviderExtraData() (Java method), 342  
getProviderFacilities() (Java method), 248  
getProviderFacilitySet() (Java method), 248  
getProviderId() (Java method), 414, 496, 500, 502, 517  
getProviderOrganization() (Java method), 253  
getProviderOrganizations() (Java method), 248  
getProviderOrganizationSet() (Java method), 248  
getProviders() (Java method), 250  
getProviderStatus() (Java method), 496, 500, 502  
getProviderTimestamp() (Java method), 342  
getPurpose() (Java method), 217  
getQueryParams() (Java method), 502  
getQuestion() (Java method), 388  
getQuestionLabel() (Java method), 160

getQuestionRepeat() (Java method), 160  
getQuestions() (Java method), 160, 388  
getQuestionTag() (Java method), 160  
getQuestionValue() (Java method), 160  
getQuiz() (Java method), 385  
getQuizById(long) (Java method), 396  
getQuizByName(String) (Java method), 397  
getQuizForChapter(long) (Java method), 397  
getQuizName() (Java method), 382  
getQuizScore() (Java method), 383  
getReason() (Java method), 102, 103, 328  
getReceivedOn() (Java method), 151, 162  
getReceivedOnEnd() (Java method), 189  
getReceivedOnStart() (Java method), 189  
getRecipients() (Java method), 517  
getRecord() (Java method), 211  
getRecords() (Java method), 132, 498, 505  
getReference() (Java method), 286  
getReferenceDate() (Java method), 359, 375, 469, 480  
getReferenceDateTime() (Java method), 471, 478, 480  
getReferenceTime() (Java method), 469, 480  
getRegion() (Java method), 406  
getRegistration() (Java method), 290  
getReminderInterval() (Java method), 91  
getReminderRepeatIntervalInMinutes() (Java method), 437, 441  
getReminderRepeatWindowInHours() (Java method), 441  
getReminderStartTime() (Java method), 91  
getRepeatEvery() (Java method), 362  
getRepeatInterval() (Java method), 370  
getRepeatIntervalInMillis() (Java method), 370  
getRepeatIntervalInMinutes() (Java method), 443  
getRepeatIntervalInSeconds() (Java method), 370  
getRepeatOn() (Java method), 363  
getResources() (Java method), 289  
getResourceStream(String) (Java method), 205  
getResourceUi() (Java method), 157  
getResourceUri() (Java method), 148, 149, 151, 154  
getResponseLastCapturedDate() (Java method), 439, 444  
getRows() (Java method), 124, 133, 498  
getSchedulableDailyJob(PillRegimen, Dosage) (Java method), 450  
getSchedule() (Java method), 455, 458  
getSchedule(String) (Java method), 476  
getScheduleByName(String) (Java method), 487, 492  
getScheduleDetails() (Java method), 448  
getScheduleName() (Java method), 455, 469, 472, 478, 481  
getSecondaryCriteria() (Java method), 376, 483  
getSecret() (Java method), 332  
getSecurityRole() (Java method), 415  
getSendReminders() (Java method), 91  
getSerializedModules() (Java method), 148  
getServerDateModified() (Java method), 139, 141  
getServerDateOpened() (Java method), 139, 141  
getServerModifiedOn() (Java method), 144, 172  
getServerURI() (Java method), 298  
getServerUrl() (Java method), 513, 514  
getServiceByEntityID(String) (Java method), 265  
getServiceDirectory() (Java method), 213  
getServices() (Java method), 227, 251, 258  
getServicesMap() (Java method), 345  
getServicesMapString() (Java method), 345  
getSettings() (Java method), 299, 305  
getSidx() (Java method), 124  
getSingleEvent() (Java method), 290  
getSiteCode() (Java method), 154  
getSmsDirection() (Java method), 496, 500  
getSmsDirections() (Java method), 502  
getSord() (Java method), 125  
getSourceDirectory() (Java method), 211, 256  
getSpecializations() (Java method), 237  
getSpecialties() (Java method), 248  
getSplitFooter() (Java method), 507  
getSplitHeader() (Java method), 507  
getStages() (Java method), 269  
getStartDate() (Java method), 216, 439, 440, 444, 447  
getStartHour() (Java method), 437  
getStartingMilestoneName() (Java method), 469, 481  
getStartKey() (Java method), 312  
getStartMinute() (Java method), 438  
getStartOfDueWindow() (Java method), 478  
getStartOfEarliestWindow() (Java method), 478  
getStartOfLateWindow() (Java method), 478  
getStartOfMaxWindow() (Java method), 478  
getStartOfSchedule() (Java method), 455  
getStartOfWindowForCurrentMilestone(WindowName) (Java method), 455  
getStartTime() (Java method), 361, 363, 383  
getStartValue() (Java method), 312  
getState() (Java method), 383, 387  
getStateProvince() (Java method), 406  
getStatus() (Java method), 84, 91, 108, 192, 211, 256, 283, 284, 286, 359, 376, 455, 479  
getStatusCode() (Java method), 417  
getStatusFieldMapString() (Java method), 345  
getStoredBy() (Java method), 284  
getStreamContent(long) (Java method), 120, 122  
getStreamContent(String, String) (Java method), 120, 122  
getStreamContent(String, String, HttpServletResponse) (Java method), 129  
getStringContent(long) (Java method), 121, 122  
getStringContent(String, String) (Java method), 120, 122  
getStringContent(String, String, HttpServletResponse) (Java method), 129  
getSubElements() (Java method), 165  
getSubject() (Java method), 307

getSubjects() (Java method), 314  
getSubscriptionStatusCode() (Java method), 332  
getSubscriptionStatusId() (Java method), 332  
getSuffix() (Java method), 233, 245  
getSurname() (Java method), 233, 245  
getSystemId() (Java method), 415  
getTemplate(String) (Java method), 351, 520, 521  
getTemplateName() (Java method), 342, 507  
getTimeEnd() (Java method), 192  
getTimeOffset() (Java method), 363, 369  
getTimePeriod() (Java method), 216  
getTimePeriodMultiplier() (Java method), 216  
getTimeStamp() (Java method), 99, 307, 326  
getTimestamp() (Java method), 496, 500  
getTimestampRange() (Java method), 502  
getTimeStart() (Java method), 192  
getTo() (Java method), 342  
getTopicUrl() (Java method), 333  
getToStatus() (Java method), 93  
getTotal() (Java method), 133, 289, 498  
getTotalCount() (Java method), 155, 156  
getTrackedEntities() (Java method), 296, 305  
getTrackedEntity() (Java method), 269, 290, 293  
getTrackedEntityAttribute() (Java method), 292  
getTrackedEntityAttributeByHref(String) (Java method), 296  
getTrackedEntityAttributes() (Java method), 296  
getTrackedEntityByHref(String) (Java method), 296  
getTrackedEntityInstance() (Java method), 284, 287, 293  
getType() (Java method), 131, 149, 151, 209, 221, 278  
getUiversion() (Java method), 152  
getUpdated() (Java method), 256, 288  
getUpdateElement() (Java method), 185  
getUpdateTask() (Java method), 143  
getUrl() (Java method), 149  
getUrn() (Java method), 221  
getUserByUserName(String) (Java method), 434  
getUserByUuid(String) (Java method), 434  
getUserData() (Java method), 157  
getUserID() (Java method), 192  
getUserId() (Java method), 139, 141, 143, 145, 172, 185, 415  
getUserName() (Java method), 415  
getUsername() (Java method), 134, 147, 157, 168, 298, 320, 345  
getUuid() (Java method), 268–272, 399, 405  
getValue() (Java method), 118, 159, 161, 162, 165, 210, 214, 242, 278, 281, 347, 398, 409, 509  
getValues() (Java method), 168  
getVersion() (Java method), 139, 141, 149, 152  
getVisitedDate() (Java method), 92  
getWindowDuration(WindowName) (Java method), 459  
getWindowEnd(WindowName) (Java method), 459  
getWindowName() (Java method), 472

getWindowStart(WindowName) (Java method), 459  
getXformIds() (Java method), 139, 141  
getXml() (Java method), 266  
getXmlContent() (Java method), 262  
getXmlns() (Java method), 143, 186, 187, 189, 192  
getXmlUrl() (Java method), 216  
GreaterOrEqual (Java class), 80  
GreaterOrEqual(String, T) (Java constructor), 80  
GridSettings (Java class), 124

## H

handle(MotechEvent) (Java method), 489  
handleCMSLiteException(Exception) (Java method), 129  
handleConfigNotFoundException(ConfigNotFoundException) (Java method), 267  
handleContentNotFoundException(Exception) (Java method), 130  
handleCreate(MotechEvent) (Java method), 273  
handleCreateAndEnroll(MotechEvent) (Java method), 273  
handleDataValue(MotechEvent) (Java method), 273  
handleDataValueSet(MotechEvent) (Java method), 273  
handleEnrollment(MotechEvent) (Java method), 274  
handleException(Exception) (Java method), 266, 267  
handleFault(MessageContext, Object) (Java method), 267  
handleFileChanged(MotechEvent) (Java method), 514, 521  
handleRequest(MessageContext, Object) (Java method), 267  
handleResponse(MessageContext, Object) (Java method), 267  
handleStageUpdate(MotechEvent) (Java method), 274  
HAS\_MANAGE\_CSD\_PERMISSION (Java field), 206  
HAS\_MANAGE\_SMS\_ROLE (Java field), 521  
HAS\_VIEW\_SMS\_LOGS\_ROLE (Java field), 521  
hasConfig() (Java method), 517  
hasConfig(String) (Java method), 350, 511, 513, 514  
hasConfigs() (Java method), 513, 514  
hasDefault() (Java method), 137  
hasDeliveryTime() (Java method), 517  
hasExpiredSince(DateTime, String) (Java method), 462  
hashCode() (Java method), 84, 116, 118, 131, 148, 208–211, 213, 214, 216, 218, 219, 221, 224, 226–228, 230, 233, 234, 237, 239–242, 244, 245, 248, 250, 251, 253, 254, 256–258, 278, 280, 281, 284, 286–288, 290–293, 343, 345, 347, 363, 364, 398, 400, 401, 406, 409, 411, 413, 415, 447, 460, 462, 464, 500, 518  
hasMotechId() (Java method), 517  
hasRegistration() (Java method), 269, 270  
hasTemplate(String) (Java method), 351  
havingCurrentMilestone(String) (Java method), 484  
havingExternalId(String) (Java method), 484  
havingMetadata(String, String) (Java method), 484



- ul style="list-style-type: none; padding-left: 0;">
- havingSchedule(String) (Java method), 484
- havingState(CampaignEnrollmentStatus) (Java method), 376
- havingState(EnrollmentStatus) (Java method), 484
- havingWindowEndingDuring(WindowName, DateTime, DateTime) (Java method), 485
- havingWindowStartingDuring(WindowName, DateTime, DateTime) (Java method), 485
- HEADERS (Java field), 321
- HIGH (Java field), 86
- HTTP\_REQUEST (Java field), 322
- HttpAgent (Java interface), 322
- HttpException (Java class), 417
- HttpException(String, HttpStatus) (Java constructor), 417
- HttpException(String, Throwable) (Java constructor), 417
- HttpMethod (Java enum), 347
- HubDistributionContent (Java class), 328
- HubDistributionContent() (Java constructor), 328
- HubDistributionContent(String, String) (Java constructor), 328
- HubDistributionContentMDSService (Java interface), 333
- HubDistributionStatus (Java class), 329
- HubDistributionStatus() (Java constructor), 329
- HubDistributionStatus(Integer, String) (Java constructor), 329
- HubDistributionStatusMDSService (Java interface), 333
- HubError (Java class), 326
- HubError() (Java constructor), 326
- HubErrors (Java interface), 326
- HubException (Java class), 327
- HubException(HubErrors) (Java constructor), 327
- HubException(HubErrors, String) (Java constructor), 327
- HubException(HubErrors, Throwable) (Java constructor), 327
- HubException(HubErrors, Throwable, String) (Java constructor), 327
- HubPublisherTransaction (Java class), 329
- HubPublisherTransaction() (Java constructor), 330
- HubPublisherTransaction(Integer, Integer, DateTime) (Java constructor), 330
- HubPublisherTransactionMDSService (Java interface), 333
- HubSettings (Java class), 335
- HubSubscriberTransaction (Java class), 330
- HubSubscriberTransaction() (Java constructor), 330
- HubSubscriberTransaction(Integer, Integer, Integer) (Java constructor), 330
- HubSubscriberTransactionMDSService (Java interface), 334
- HubSubscription (Java class), 331
- HubSubscription() (Java constructor), 331
- HubSubscription(Integer, Integer, String) (Java constructor), 331
- HubSubscription(Integer, Integer, String, Integer, String) (Java constructor), 331
- HubSubscriptionMDSService (Java interface), 334
- HubSubscriptionStatus (Java class), 332
- HubSubscriptionStatus() (Java constructor), 332
- HubSubscriptionStatus(Integer, String) (Java constructor), 332
- HubSubscriptionStatusMDSService (Java interface), 334
- HubTopic (Java class), 333
- HubTopic() (Java constructor), 333
- HubTopic(String) (Java constructor), 333
- HubTopicMDSService (Java interface), 334
- I
- id(String) (Java method), 415
- ImportCountDto (Java class), 288
- importTemplate(Template) (Java method), 520, 521
- importTemplates(List) (Java method), 520, 521
- INACTIVE (Java field), 111, 360
- Inactive (Java field), 387
- INBOUND (Java field), 344, 495
- INBOUND\_SMS (Java field), 523
- inboundEvent(String, String, String, String, String, DateTime) (Java method), 524
- IndexSubElement (Java class), 190
- IndexSubElement(String, String, String) (Java constructor), 190
- IndexTask (Java class), 167
- IndexTask(List) (Java constructor), 167
- initiateCall(String, Map) (Java method), 351
- INPROGRESS (Java field), 384
- INTENT\_FAILED (Java field), 336
- INTENT\_VERIFIED (Java field), 336
- interval() (Java method), 464
- InWindowCriterion (Java class), 466
- InWindowCriterion(List) (Java constructor), 467
- isAbsoluteSchedule() (Java method), 464
- isActive() (Java method), 360, 455
- isArchived() (Java method), 152
- isAttribute(String) (Java method), 182
- isAuthRequired() (Java method), 345
- isBasedOnAbsoluteWindows() (Java method), 462
- isClose() (Java method), 146
- isClosed() (Java method), 139, 141
- isCompleted() (Java method), 455
- isCorrect(String, String, String) (Java method), 125
- isDataSetComplete() (Java method), 283
- isDead() (Java method), 413
- isDescending() (Java method), 125
- isEmpty() (Java method), 511
- isFirstReminderFor(Dosage) (Java method), 448
- isFloating() (Java method), 453, 464
- isFollowup() (Java method), 284, 287
- isForwardCases() (Java method), 135

isForwardForms() (Java method), 135  
isForwardSchema() (Java method), 135  
isForwardStubs() (Java method), 135  
isIncludeArchived() (Java method), 189  
isJsonRequest() (Java method), 345  
isJsonResponse() (Java method), 345  
isLoggableEvent(MotechEvent) (Java method), 314  
isOpenFlag() (Java method), 234  
isOptional() (Java method), 312  
isSchedulerEnabled() (Java method), 216  
isSingleEvent() (Java method), 269  
isStartingMilestoneSpecified() (Java method), 481  
isStream() (Java method), 125  
isStreamContentAvailable(String, String) (Java method), 121, 122  
isString() (Java method), 125  
isStringContentAvailable(String, String) (Java method), 121, 122  
isTodaysDosageResponseCaptured() (Java method), 445  
isValue(String) (Java method), 182  
IvrTemplateException (Java class), 348  
IvrTemplateException(String) (Java constructor), 348  
IvrTemplateException(String, HttpStatus) (Java constructor), 348

## J

JOB\_NOT\_FOUND (Java field), 98  
JOB\_TRIGGER\_FAILED (Java field), 99  
JobExecutionHistoryListDTO (Java class), 111  
JobService (Java interface), 112  
JobStatusLookup (Java enum), 111  
JobTriggerService (Java interface), 113

## K

KeyValue (Java class), 311  
KeyValue(String, Object, String, Object, boolean) (Java constructor), 311

## L

LAST\_CAPTURE\_DATE (Java field), 435  
LAST\_MODIFIED (Java field), 207  
late (Java field), 463  
late() (Java method), 465  
LATE\_DATE\_TIME (Java field), 473  
Lesson (Java class), 387  
Lesson(String, CourseUnitState, String) (Java constructor), 387  
LessonDataService (Java interface), 390  
LessOrEqual (Java class), 80  
LessOrEqual(String, T) (Java constructor), 80  
loadAfterInit() (Java method), 374  
loadCampaigns() (Java method), 378, 381  
loadCampaigns(InputStream) (Java method), 374  
loadCampaigns(String) (Java method), 374

loadCampaignsJson() (Java method), 381  
loadSingleCampaign(InputStream) (Java method), 374  
loadSingleCampaign(String) (Java method), 374  
LOCATION (Java field), 275  
log(MotechEvent) (Java method), 308–310  
log(String) (Java method), 310  
logEvent(MotechEvent) (Java method), 316, 319, 320  
LoggableEvent (Java class), 313  
LoggableEvent(List, List) (Java constructor), 313  
LogMappings (Java class), 312  
LogMappings(List, List, List) (Java constructor), 312  
LOW (Java field), 86

## M

MALFORMED\_CASE\_EXCEPTION (Java field), 179  
MalformedCaseXmlException (Java class), 181  
MalformedCaseXmlException(Exception, String) (Java constructor), 181  
MalformedCaseXmlException(String) (Java constructor), 181  
MalformedFormStatusMessageEvent (Java class), 175  
MalformedFormStatusMessageEvent(String) (Java constructor), 175  
MANAGE\_CSD (Java field), 206  
MANAGE\_SMS\_PERMISSION (Java field), 521  
map(CampaignEnrollment) (Java method), 376  
map(Enrollment) (Java method), 490  
mapFromExternalId(String) (Java method), 301  
MappingsJson (Java class), 314  
MapQueryExecution (Java class), 475  
MapQueryExecution(String, String) (Java constructor), 475  
mapStatusField(String) (Java method), 346  
mapWithDates(Enrollment) (Java method), 490  
MARK\_ALERT\_READ\_SUBJECT (Java field), 75  
marshall(Object, URL, Class) (Java method), 266  
MarshallUtils (Java class), 266  
max (Java field), 463  
max() (Java method), 465  
Medicine (Java class), 446  
Medicine() (Java constructor), 446  
Medicine(String, LocalDate, LocalDate) (Java constructor), 446  
MEDICINE\_END\_DATE\_CANNOT\_BE\_BEFORE\_START\_DATE (Java field), 446  
MedicineRequest (Java class), 439  
MedicineRequest(String, LocalDate, LocalDate) (Java constructor), 439  
MedicineResponse (Java class), 440  
MedicineResponse(String, LocalDate, LocalDate) (Java constructor), 440  
MEDIUM (Java field), 86  
merge(Schedule) (Java method), 462  
MESSAGE (Java field), 177, 522

- MESSAGE\_CAMPAIGNS\_JSON\_FILENAME (Java field), 378
- MESSAGE\_FORMATS (Java field), 353
- MESSAGE\_KEY (Java field), 353
- MESSAGE\_LANGUAGES (Java field), 353
- MESSAGE\_NAME\_KEY (Java field), 353
- MessageCampaignException (Java class), 373
- MessageCampaignException(String, String, List, Throwable) (Java constructor), 373
- MessageCampaignException(String, String, Throwable) (Java constructor), 373
- MessageCampaignService (Java interface), 377
- MessageCampaignServiceImpl (Java class), 380
- MessageCampaignServiceImpl(EnrollmentService, CampaignEnrollmentDataService, CampaignEnrollmentRecordMapper, CampaignSchedulerFactory, CampaignRecordService, CampaignMessageRecordService, EventRelay, MotechSchedulerService) (Java constructor), 380
- MessageKey (Java class), 373
- MessageKey(String, List) (Java constructor), 373
- META\_APP\_VERSION (Java field), 177
- META\_BASE\_URL (Java field), 177
- META\_DEVICE\_ID (Java field), 177
- META\_DOMAIN (Java field), 177
- META\_INSTANCE\_ID (Java field), 177
- META\_TIME\_END (Java field), 177
- META\_TIME\_START (Java field), 177
- META\_USER\_ID (Java field), 178
- META\_USERNAME (Java field), 177
- METADATA (Java field), 473
- MetadataCriterion (Java class), 467
- MetadataCriterion(String, String) (Java constructor), 467
- metadataUpdater (Java field), 482
- MetadataValue (Java class), 167
- MetadataValue(Collection) (Java constructor), 167
- MetadataValue(String) (Java constructor), 167
- MetaElement (Java class), 191
- MetaElement(String, String, String, String, String) (Java constructor), 191
- Method (Java enum), 325
- METHOD (Java field), 321
- Milestone (Java class), 458
- Milestone() (Java constructor), 458
- Milestone(String, Period, Period, Period, Period) (Java constructor), 458
- MILESTONE\_ALERT (Java field), 474
- MILESTONE\_DATA (Java field), 473
- MILESTONE\_DEFAULTED (Java field), 474
- MILESTONE\_NAME (Java field), 473
- MilestoneAlert (Java class), 459
- MilestoneAlert(String, DateTime, DateTime, DateTime, DateTime) (Java constructor), 459
- MilestoneAlerts (Java class), 485
- MilestoneAlerts() (Java constructor), 485
- MilestoneCriterion (Java class), 467
- MilestoneCriterion(String) (Java constructor), 467
- MilestoneDefaultedEvent (Java class), 470
- MilestoneDefaultedEvent(Long, String, String) (Java constructor), 470
- MilestoneDefaultedEvent(MotechEvent) (Java constructor), 470
- MilestoneEvent (Java class), 470
- MilestoneEvent(Enrollment, MilestoneAlert, MilestoneWindow) (Java constructor), 471
- MilestoneEvent(MotechEvent) (Java constructor), 471
- MilestoneEvent(String, String, MilestoneAlert, String, DateTime, Map) (Java constructor), 470
- MilestoneFulfillment (Java class), 460
- MilestoneFulfillment() (Java constructor), 460
- MilestoneFulfillment(String, DateTime) (Java constructor), 460
- MilestoneRangeQueryExecution (Java class), 475
- MilestoneRangeQueryExecution(DateTime, DateTime) (Java constructor), 475
- MilestoneRecord (Java class), 464
- milestoneRecords() (Java method), 465
- MilestoneWindow (Java class), 460
- MilestoneWindow() (Java constructor), 460
- MilestoneWindow(WindowName, Period) (Java constructor), 461
- MISSED (Java field), 95
- Modes (Java enum), 335
- MONDAY (Java field), 220
- Monday (Java field), 367
- MOTech\_ID (Java field), 421, 522
- MTrainingService (Java interface), 393
- ## N
- Name (Java class), 231
- NAME (Java field), 87
- Name() (Java constructor), 232
- name() (Java method), 398, 464, 465
- Name(List) (Java constructor), 232
- Name(List, String, String, Set, String, String) (Java constructor), 232
- name(String) (Java method), 81
- NAME\_UPDATER (Java field), 86
- NameDataService (Java interface), 260
- nameInUse(String) (Java method), 137
- NEW (Java field), 86
- newAlerts() (Java method), 90
- newOrReadAlerts() (Java method), 90
- nextPage() (Java method), 189
- NONE (Java field), 95, 384
- numberOfAlertsToSchedule() (Java method), 454

numberOfTimesPillRemindersSentFor(Dosage) (Java method), 449

## O

OBSERVATION\_CONCEPT\_NAME (Java field), 421

OBSERVATION\_DATE (Java field), 421

OBSERVATION\_VALUE (Java field), 421

ObservationNotFoundException (Java class), 418

ObservationNotFoundException(String) (Java constructor), 418

ObservationNotFoundException(String, Throwable) (Java constructor), 418

OFFSET (Java field), 365

offset() (Java method), 464

OffsetCampaign (Java class), 368

OffsetCampaign() (Java constructor), 368

OffsetCampaign(String, List) (Java constructor), 368

OffsetCampaign(String, List, Period) (Java constructor), 368

OffsetCampaignMessage (Java class), 369

OffsetCampaignMessage(CampaignMessageRecord) (Java constructor), 369

OffsetCampaignMessage(String, List, List, String, Time, Period) (Java constructor), 369

OffsetCampaignMessage(Time) (Java constructor), 369

OffsetCampaignMessage(Time, Period) (Java constructor), 369

OneTimeJobScheduleParams (Java class), 111

OpenMRSAtribute (Java class), 398

OpenMRSAtribute(String, String) (Java constructor), 398

OpenMRSEncounter (Java class), 399

OpenMRSEncounter() (Java constructor), 399

OpenMRSEncounter(OpenMRSEncounterName) (Java constructor), 399

OpenMRSEncounterName (Java class), 400

OpenMRSEncounterName(Concept.ConceptName) (Java constructor), 401

OpenMRSEncounterName(String) (Java constructor), 400

OpenMRSEncounterName(String, String, String) (Java constructor), 401

OpenMRSEncounterService (Java interface), 423

OpenMRSEncounter (Java class), 402

OpenMRSEncounter() (Java constructor), 402

OpenMRSEncounter(OpenMRSEncounter, OpenMRSEncounterUser, OpenMRSEncounterFacility, Date, OpenMRSEncounterPatient, Set, String) (Java constructor), 402

OpenMRSEncounterBuilder (Java class), 403

OpenMRSEncounterService (Java interface), 425

OpenMRSEncounterType (Java class), 404

OpenMRSEncounterException (Java class), 418

OpenMRSEncounterException(String) (Java constructor), 418

OpenMRSEncounterException(String, Throwable) (Java constructor), 418

OpenMRSEncounterException(Throwable) (Java constructor), 418

OpenMRSEncounterFacility (Java class), 405

OpenMRSEncounterFacility(String) (Java constructor), 405

OpenMRSEncounterFacility(String, String, String, String, String) (Java constructor), 405

OpenMRSEncounterFacility(String, String, String, String, String, String) (Java constructor), 406

OpenMRSEncounterFacilityService (Java interface), 426

OpenMRSEncounterObservation (Java class), 407

OpenMRSEncounterObservation(Date, String, String, T) (Java constructor), 407

OpenMRSEncounterObservation(Date, String, T) (Java constructor), 407

OpenMRSEncounterObservation(String, Date, String, String, T) (Java constructor), 408

OpenMRSEncounterObservation(String, Date, String, T) (Java constructor), 408

OpenMRSEncounterObservationService (Java interface), 428

OpenMRSEncounterPatient (Java class), 409

OpenMRSEncounterPatient() (Java constructor), 409

OpenMRSEncounterPatient(String) (Java constructor), 410

OpenMRSEncounterPatient(String, OpenMRSEncounterPerson, OpenMRSEncounterFacility) (Java constructor), 410

OpenMRSEncounterPatient(String, String, OpenMRSEncounterPerson, OpenMRSEncounterFacility) (Java constructor), 410

OpenMRSEncounterPatientService (Java interface), 429

OpenMRSEncounterPerson (Java class), 411

OpenMRSEncounterPerson() (Java constructor), 411

OpenMRSEncounterPerson(String) (Java constructor), 411

OpenMRSEncounterPersonService (Java interface), 431

OpenMRSEncounterProvider (Java class), 414

OpenMRSEncounterProvider() (Java constructor), 414

OpenMRSEncounterProvider(OpenMRSEncounterPerson) (Java constructor), 414

OpenMRSEncounterProviderService (Java interface), 432

OpenMRSEncounterUser (Java class), 415

OpenMRSEncounterUserService (Java interface), 433

OpenMRSEncounterResponse (Java class), 192

OpenMRSEncounterResponseParser (Java class), 184

OperatingHours (Java class), 233

OperatingHours() (Java constructor), 234

OperatingHours(boolean, DateTime) (Java constructor), 234

OperatingHours(boolean, List, DateTime, DateTime, DateTime, DateTime) (Java constructor), 234

OperatingHoursDataService (Java interface), 260

org.motechproject.alerts (package), 74

org.motechproject.alerts.contract (package), 75

org.motechproject.alerts.domain (package), 82

org.motechproject.alerts.service (package), 87

org.motechproject.alerts.web (package), 90

org.motechproject.appointments.domain (package), 90

org.motechproject.appointments.repository (package), 95

org.motechproject.appointments.service (package), 96



- org.motechproject.batch.exception (package), 98
- org.motechproject.batch.mds (package), 103
- org.motechproject.batch.mds.service (package), 106
- org.motechproject.batch.model (package), 107
- org.motechproject.batch.service (package), 112
- org.motechproject.cmslite.model (package), 114
- org.motechproject.cmslite.service (package), 119
- org.motechproject.cmslite.web (package), 124
- org.motechproject.commcare.builder (package), 133
- org.motechproject.commcare.config (package), 133
- org.motechproject.commcare.domain (package), 138
- org.motechproject.commcare.events (package), 170
- org.motechproject.commcare.events.constants (package), 176
- org.motechproject.commcare.exception (package), 179
- org.motechproject.commcare.parser (package), 181
- org.motechproject.commcare.request (package), 184
- org.motechproject.commcare.response (package), 192
- org.motechproject.commcare.service (package), 193
- org.motechproject.commcare.tasks (package), 205
- org.motechproject.csd.constants (package), 206
- org.motechproject.csd.domain (package), 207
- org.motechproject.csd.exception (package), 259
- org.motechproject.csd.mds (package), 259
- org.motechproject.csd.service (package), 261
- org.motechproject.csd.util (package), 265
- org.motechproject.csd.web (package), 266
- org.motechproject.dhis2.domain (package), 267
- org.motechproject.dhis2.event (package), 273
- org.motechproject.dhis2.repository (package), 276
- org.motechproject.dhis2.rest.domain (package), 277
- org.motechproject.dhis2.rest.service (package), 294
- org.motechproject.dhis2.service (package), 296
- org.motechproject.dhis2.tasks (package), 301
- org.motechproject.dhis2.web (package), 304
- org.motechproject.eventlogging.converter (package), 305
- org.motechproject.eventlogging.converter.impl (package), 306
- org.motechproject.eventlogging.domain (package), 307
- org.motechproject.eventlogging.loggers (package), 308
- org.motechproject.eventlogging.loggers.impl (package), 309
- org.motechproject.eventlogging.matchers (package), 310
- org.motechproject.eventlogging.service (package), 316
- org.motechproject.eventlogging.service.impl (package), 318
- org.motechproject.http.agent.domain (package), 320
- org.motechproject.http.agent.service (package), 322
- org.motechproject.hub.exception (package), 325
- org.motechproject.hub.mds (package), 328
- org.motechproject.hub.mds.service (package), 333
- org.motechproject.hub.model (package), 335
- org.motechproject.hub.service (package), 336
- org.motechproject.hub.service.impl (package), 338
- org.motechproject.ivr.domain (package), 340
- org.motechproject.ivr.exception (package), 348
- org.motechproject.ivr.repository (package), 349
- org.motechproject.ivr.service (package), 349
- org.motechproject.messagecampaign (package), 352
- org.motechproject.messagecampaign.contract (package), 354
- org.motechproject.messagecampaign.dao (package), 355
- org.motechproject.messagecampaign.domain.campaign (package), 357
- org.motechproject.messagecampaign.exception (package), 371
- org.motechproject.messagecampaign.loader (package), 374
- org.motechproject.messagecampaign.service (package), 375
- org.motechproject.mtraining.domain (package), 382
- org.motechproject.mtraining.repository (package), 389
- org.motechproject.mtraining.service (package), 391
- org.motechproject.openmrs19.domain (package), 398
- org.motechproject.openmrs19.exception (package), 417
- org.motechproject.openmrs19.service (package), 419
- org.motechproject.pillreminder (package), 434
- org.motechproject.pillreminder.contract (package), 436
- org.motechproject.pillreminder.dao (package), 442
- org.motechproject.pillreminder.domain (package), 442
- org.motechproject.pillreminder.service (package), 450
- org.motechproject.scheduletracking.domain (package), 452
- org.motechproject.scheduletracking.domain.json (package), 463
- org.motechproject.scheduletracking.domain.search (package), 465
- org.motechproject.scheduletracking.events (package), 469
- org.motechproject.scheduletracking.events.constants (package), 472
- org.motechproject.scheduletracking.repository (package), 474
- org.motechproject.scheduletracking.repository.dataservices (package), 476
- org.motechproject.scheduletracking.service (package), 477
- org.motechproject.scheduletracking.service.impl (package), 489
- org.motechproject.sms.audit (package), 493
- org.motechproject.sms.configs (package), 507
- org.motechproject.sms.service (package), 512
- org.motechproject.sms.util (package), 521
- org.motechproject.transliteration.hindi.service (package), 525
- ORG\_UNIT (Java field), 303
- OrganisationUnitDto (Java class), 288
- Organization (Java class), 235

Organization() (Java constructor), 236  
Organization(String, Set, String, Record) (Java constructor), 236  
Organization(String, Set, String, Set, Set, String, Set, Set, Set, Set, Set, Set, Record) (Java constructor), 236  
OrganizationContact (Java class), 238  
OrganizationContact() (Java constructor), 238  
OrganizationContact(Person) (Java constructor), 239  
OrganizationContact(String) (Java constructor), 239  
OrganizationContactDataService (Java interface), 260  
OrganizationDataService (Java interface), 260  
OrganizationDirectory (Java class), 239  
OrganizationDirectory() (Java constructor), 240  
OrganizationDirectory(Set) (Java constructor), 240  
OrganizationService (Java interface), 263  
OrgUnit (Java class), 268  
OrgUnit() (Java constructor), 268  
OrgUnit(String, String) (Java constructor), 268  
OrgUnitDataService (Java interface), 276  
OrgUnitService (Java interface), 297  
OtherID (Java class), 240  
OtherID() (Java constructor), 241  
OtherID(String) (Java constructor), 241  
OtherID(String, String) (Java constructor), 241  
OtherIDDataService (Java interface), 260  
OtherName (Java class), 241  
OtherName() (Java constructor), 242  
OtherName(String) (Java constructor), 242  
OtherName(String, String) (Java constructor), 242  
OtherNameDataService (Java interface), 261  
OUTBOUND (Java field), 344, 495  
OutboundCallService (Java interface), 350  
outboundEvent(String, String, List, String, String, String, Integer, String, DateTime, Map) (Java method), 524  
OutgoingSms (Java class), 514  
OutgoingSms() (Java constructor), 514  
OutgoingSms(List, String) (Java constructor), 516  
OutgoingSms(List, String, DateTime) (Java constructor), 515  
OutgoingSms(MotechEvent) (Java constructor), 515  
OutgoingSms(String, List, String) (Java constructor), 515  
OutgoingSms(String, List, String, DateTime) (Java constructor), 515  
OutgoingSms(String, String) (Java constructor), 516  
OutgoingSms(String, String, DateTime) (Java constructor), 516  
OutgoingSms(String, String, String) (Java constructor), 515  
OutgoingSms(String, String, String, DateTime) (Java constructor), 515  
OWNER\_ID (Java field), 178

## P

PagedResourceDto (Java class), 289  
PagerDto (Java class), 289  
ParametersPresentEventFlag (Java class), 315  
ParametersPresentEventFlag() (Java constructor), 315  
ParametersPresentEventFlag(Map) (Java constructor), 315  
parse() (Java method), 184  
parseCase() (Java method), 182  
parseCase(Document) (Java method), 182  
parseResponse(String) (Java method), 184  
passesFlags(MotechEvent) (Java method), 311, 315, 316  
Password (Java class), 416  
PASSWORD (Java field), 321  
Password(Integer) (Java constructor), 417  
PASSWORD\_KEY (Java field), 433  
PATIENT\_DECEASED\_SUBJECT (Java field), 421  
PATIENT\_ID (Java field), 422  
PatientNotFoundException (Java class), 418  
PatientNotFoundException(Exception) (Java constructor), 418  
PatientNotFoundException(String) (Java constructor), 418  
PENDING (Java field), 493, 523  
Pending (Java field), 387  
PERIOD (Java field), 275, 303  
Person (Java class), 242  
Person() (Java constructor), 243  
person(OpenMRSPerson) (Java method), 415  
Person(Set) (Java constructor), 243  
Person(Set, Set, Set, String, DateTime) (Java constructor), 243  
PERSON\_ADDRESS (Java field), 422  
PERSON\_AGE (Java field), 422  
PERSON\_BIRTH\_DATE\_ESTIMATED (Java field), 422  
PERSON\_DATE\_OF\_BIRTH (Java field), 422  
PERSON\_DEAD (Java field), 422  
PERSON\_DEATH\_DATE (Java field), 422  
PERSON\_FIRST\_NAME (Java field), 422  
PERSON\_GENDER (Java field), 422  
PERSON\_ID (Java field), 422  
PERSON\_LAST\_NAME (Java field), 422  
PERSON\_MIDDLE\_NAME (Java field), 422  
PERSON\_PREFERRED\_NAME (Java field), 422  
PersonDataService (Java interface), 261  
PersonName (Java class), 244  
PersonName() (Java constructor), 244  
PersonName(List) (Java constructor), 244  
PersonName(List, String, String, Set, String, String, String) (Java constructor), 244  
PersonNameDataService (Java interface), 261  
PILL\_REGIMEN\_ID (Java field), 436  
PillRegimen (Java class), 447  
PillRegimen() (Java constructor), 447

- PillRegimen(String, Set, DailyScheduleDetails) (Java constructor), 448
  - PillRegimenDataService (Java interface), 442
  - PillRegimenJobScheduler (Java class), 450
  - PillRegimenJobScheduler(MotechSchedulerService) (Java constructor), 450
  - PillRegimenResponse (Java class), 440
  - PillRegimenResponse(Long, String, int, int, int, List) (Java constructor), 440
  - PILLREMINDER\_REMINDER\_EVENT\_SUBJECT (Java field), 435
  - PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_DOSAGE\_STATUS\_UNKNOWN (Java field), 435
  - PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_SCHEDULER (Java field), 435
  - PILLREMINDER\_REMINDER\_EVENT\_SUBJECT\_UNSUBSCRIBE (Java field), 435
  - PILLREMINDER\_RETRY\_INTERVAL (Java field), 435
  - PILLREMINDER\_TIMES\_SENT (Java field), 435
  - PILLREMINDER\_TOTAL\_TIMES\_TO\_SEND (Java field), 435
  - PillReminderService (Java interface), 450
  - PillReminderServiceImpl (Java class), 452
  - PillReminderServiceImpl(PillRegimenDataService, PillRegimenJobScheduler) (Java constructor), 452
  - POST (Java field), 325, 347
  - PREFERRED\_ALERT\_TIME (Java field), 473
  - PREFIX\_ATTRIBUTE (Java field), 159
  - PREFIX\_SEARCH\_FROM\_ROOT (Java field), 159
  - PREFIX\_SEARCH\_RELATIVE (Java field), 159
  - PREFIX\_VALUE (Java field), 159
  - previousPage() (Java method), 189
  - PRIORITY (Java field), 87
  - priority(int) (Java method), 81
  - PRIORITY\_UPDATER (Java field), 87
  - Program (Java class), 269
  - PROGRAM (Java field), 275
  - PROGRAM\_ENROLLMENT (Java field), 303
  - PROGRAM\_NAME (Java field), 303
  - ProgramActionBuilder (Java class), 304
  - ProgramDataService (Java interface), 276
  - ProgramDto (Java class), 290
  - ProgramService (Java interface), 298
  - ProgramStageDataElementDto (Java class), 291
  - ProgramStageDto (Java class), 291
  - ProgramTrackedEntityAttributeDto (Java class), 292
  - Provider (Java class), 246
  - Provider() (Java constructor), 247
  - Provider(String, Set, Record, Person) (Java constructor), 247
  - Provider(String, Set, Set, Record, Set, Person, Set, Set, Set, Set, Set) (Java constructor), 247
  - PROVIDER\_ID (Java field), 422
  - PROVIDER\_MESSAGE\_ID (Java field), 522
  - PROVIDER\_STATUS (Java field), 522
  - ProviderDataService (Java interface), 261
  - ProviderDirectory (Java class), 249
  - ProviderDirectory() (Java constructor), 250
  - ProviderDirectory(Set) (Java constructor), 250
  - ProviderFacility (Java class), 250
  - ProviderFacility() (Java constructor), 251
  - ProviderFacility(String) (Java constructor), 251
  - ProviderFacility(String, Set, Set, Set) (Java constructor), 251
  - ProviderFacilityDataService (Java interface), 261
  - ProviderFacilityService (Java class), 252
  - ProviderFacilityService() (Java constructor), 252
  - ProviderFacilityService(String) (Java constructor), 252
  - ProviderFacilityService(String, Set, String, Set, Set, String, Set) (Java constructor), 252
  - ProviderFacilityServiceDataService (Java interface), 261
  - ProviderOrganization (Java class), 254
  - ProviderOrganization() (Java constructor), 254
  - ProviderOrganization(String) (Java constructor), 254
  - ProviderOrganization(String, Set, Set, Set) (Java constructor), 254
  - ProviderOrganizationDataService (Java interface), 261
  - ProviderService (Java interface), 264
  - PUBLISH (Java field), 335
  - PUT (Java field), 325
- ## Q
- Question (Java class), 387
  - Question() (Java constructor), 388
  - Question(String, String) (Java constructor), 388
  - Quiz (Java class), 388
  - Quiz(String, CourseUnitState, String) (Java constructor), 388
  - Quiz(String, CourseUnitState, String, List, double) (Java constructor), 388
  - QuizDataService (Java interface), 390
- ## R
- READ (Java field), 86
  - readAlerts() (Java method), 90
  - readJson(String) (Java method), 182
  - readListJson(String) (Java method), 183
  - RECEIVED (Java field), 493
  - RECEIVED\_ON (Java field), 178
  - RECIPIENT (Java field), 522
  - RECIPIENTS (Java field), 522
  - Record (Java class), 255
  - Record() (Java constructor), 255
  - Record(DateTime, DateTime, String, String) (Java constructor), 255
  - Record(DateTime, String, DateTime) (Java constructor), 255
  - recursivelyParse(String, JsonElement) (Java method), 183

- REFERENCE\_DATE (Java field), 353, 473
  - REFERENCE\_TIME (Java field), 473
  - referenceDate() (Java method), 354
  - RegexEventFlag (Java class), 315
  - register(CampaignEnrollment) (Java method), 377
  - registerDefaultService() (Java method), 319
  - REGISTERED\_CASE\_NAME (Java field), 178
  - REGISTERED\_CASE\_TYPE (Java field), 178
  - registerEventLoggingService(EventLoggingService) (Java method), 317, 319
  - REGISTRATION (Java field), 275
  - remove(String) (Java method), 451, 452, 487, 492
  - removeAppointment(String) (Java method), 97
  - removeAppointments(List) (Java method), 97
  - removeContent(String, String, String) (Java method), 130
  - REMOVED (Java field), 95
  - removeLoggableEvents(List) (Java method), 309
  - removeStreamContent(String, String) (Java method), 121, 122
  - removeStringContent(String, String) (Java method), 121, 123
  - renew(DailyPillRegimenRequest) (Java method), 451, 452
  - REPEAT\_INTERVAL (Java field), 365
  - REPEATING\_DEFAULT\_START\_OFFSET (Java field), 352
  - REPEATING\_PROGRAM\_24HRS\_MESSAGE\_DISPATCH\_STRATEGY (Java field), 352
  - RepeatIntervalCampaign (Java class), 369
  - RepeatIntervalCampaign() (Java constructor), 369
  - RepeatIntervalCampaign(String, List) (Java constructor), 369
  - RepeatIntervalCampaign(String, List, Period) (Java constructor), 370
  - RepeatIntervalCampaignMessage (Java class), 370
  - RepeatIntervalCampaignMessage(CampaignMessageRecord) (Java constructor), 370
  - RepeatIntervalCampaignMessage(String, List, List, String, Time, Period) (Java constructor), 370
  - RepeatIntervalCampaignMessage(Time, Period) (Java constructor), 370
  - rescheduleJob(String, String) (Java method), 112
  - rescheduleMessageJob(Long) (Java method), 378, 381
  - resolveConceptUuidFromConceptName(String) (Java method), 424
  - ResourceComparator (Java class), 126
  - ResourceComparator(GridSettings) (Java constructor), 127
  - ResourceController (Java class), 127
  - ResourceDto (Java class), 130
  - ResourceDto(Content) (Java constructor), 130
  - ResourceDto(String, String, String) (Java constructor), 130
  - ResourceFilter (Java class), 131
  - Resources (Java class), 132
  - Resources(GridSettings, List) (Java constructor), 132
  - REST (Java field), 215
  - restart(String, Integer) (Java method), 114
  - RestException (Java class), 102, 328
  - RestException(BatchException, List) (Java constructor), 102
  - RestException(BatchException, String) (Java constructor), 102
  - RestException(HubException, List) (Java constructor), 328
  - RestException(HubException, String) (Java constructor), 328
  - retrieveAll() (Java method), 475
  - retrieveForm(String) (Java method), 201
  - retrieveForm(String, String) (Java method), 201
  - retrieveFormJson(String) (Java method), 202
  - retrieveFormJson(String, String) (Java method), 201
  - retrieveFormList(FormListRequest) (Java method), 202
  - retrieveFormList(FormListRequest, String) (Java method), 202
  - retrieveStreamContentData(StreamContent) (Java method), 122, 123
  - RETRY\_COUNT (Java field), 321
  - RETRY\_INTERVAL (Java field), 321
  - RETRYING (Java field), 493, 523
  - HttpStrategyStatus(Integer) (Java method), 508
  - retryOrAbortSubject(Integer) (Java method), 508
- ## S
- SATURDAY (Java field), 220
  - Saturday (Java field), 367
  - saveCampaign(CampaignRecord) (Java method), 379, 381
  - saveConfig(Config) (Java method), 137, 199
  - saveFromXml(String) (Java method), 262
  - Schedule (Java class), 461
  - Schedule() (Java constructor), 461
  - Schedule(String) (Java constructor), 461
  - SCHEDULE\_JOB\_ID\_KEY (Java field), 353
  - SCHEDULE\_NAME (Java field), 474
  - scheduleAlertsForCurrentMilestone(Enrollment) (Java method), 489
  - ScheduleCriterion (Java class), 468
  - ScheduleCriterion(String) (Java constructor), 468
  - SCHEDULED (Java field), 493, 523
  - scheduleDailyJob(PillRegimen) (Java method), 450
  - scheduledAlertStartDate() (Java method), 454
  - ScheduleDataService (Java interface), 477
  - ScheduleFactory (Java class), 462
  - ScheduleFactory() (Java constructor), 463
  - scheduleJob(CronJobScheduleParam) (Java method), 112
  - scheduleJobsForEnrollment(CampaignEnrollment) (Java method), 379, 381



- `scheduleJobToCaptureDefaultment(Enrollment)` (Java method), 490
- `scheduleOneTimeJob(OneTimeJobScheduleParams)` (Java method), 113
- `ScheduleRecord` (Java class), 464
- `ScheduleTrackingService` (Java interface), 486
- `ScheduleTrackingServiceImpl` (Java class), 491
- `ScheduleTrackingServiceImpl(EnrollmentsQueryService, ScheduleDataService, EnrollmentService, EnrollmentRecordMapper, EnrollmentDataService, SettingsFacade)` (Java constructor), 491
- `ScheduletrackingTasksActionFacade` (Java interface), 488
- `ScheduletrackingTasksActionFacadeImpl` (Java class), 492
- `ScheduleWindowsRecord` (Java class), 465
- `scheduleWindowsRecord()` (Java method), 464
- `SchedulingException` (Java class), 373
- `SchedulingException(String, Throwable)` (Java constructor), 373
- `SCHEMA_CHANGE_EVENT` (Java field), 179
- `schemaChange(MotechEvent)` (Java method), 171
- `search(AlertCriteria)` (Java method), 77, 88, 89
- `search(CampaignEnrollmentsQuery)` (Java method), 377, 379, 381
- `search(EnrollmentsQuery)` (Java method), 488, 491, 492
- `search(String)` (Java method), 166, 424
- `search(String, String)` (Java method), 430
- `searchFirst(String)` (Java method), 166
- `searchWithWindowDates(EnrollmentsQuery)` (Java method), 488, 492
- `securityRole(String)` (Java method), 416
- `send(OutgoingSms)` (Java method), 519
- `SEND_DATA_VALUE` (Java field), 275, 303
- `SEND_DATA_VALUE_SET` (Java field), 276, 303
- `SEND_MESSAGE` (Java field), 353
- `SEND_SMS` (Java field), 523
- `sendDataValueSet(DataValueSetDto)` (Java method), 296
- `SENDER` (Java field), 522
- `serializeModules()` (Java method), 148
- `SERVER_MODIFIED_ON` (Java field), 178
- `Service` (Java class), 256
- `Service()` (Java constructor), 257
- `Service(String, CodedType, Record)` (Java constructor), 257
- `Service(String, CodedType, Set, Record)` (Java constructor), 257
- `ServiceDataService` (Java interface), 261
- `ServiceDirectory` (Java class), 257
- `ServiceDirectory()` (Java constructor), 258
- `ServiceDirectory(Set)` (Java constructor), 258
- `ServiceService` (Java interface), 264
- `setAccountConfig(AccountConfig)` (Java method), 135
- `setAction(String)` (Java method), 145, 172
- `setAddress(String)` (Java method), 413
- `setAddresses(Set)` (Java method), 224, 237, 244, 255
- `setAddressLines(Set)` (Java method), 209
- `setAge(Integer)` (Java method), 413
- `setAlertFilter(AlertFilter)` (Java method), 89
- `setAlerts(List)` (Java method), 461
- `setAlertsDataService(AlertsDataService)` (Java method), 89
- `setAlertService(AlertService)` (Java method), 88
- `setAlertType(AlertType)` (Java method), 84
- `setAllEventMappings(AllEventMappings)` (Java method), 319
- `setAllMessageCampaigns(CampaignRecordService)` (Java method), 374
- `setAltitude(double)` (Java method), 230
- `setAnswer(String)` (Java method), 388
- `setApiKey(String)` (Java method), 145, 172
- `setApplication(String)` (Java method), 99, 326
- `setApplicationName(String)` (Java method), 148
- `setAppointmentDate(DateTime)` (Java method), 92, 94
- `setAppointmentId(String)` (Java method), 94
- `setAppVersion(String)` (Java method), 189
- `setArchived(boolean)` (Java method), 152
- `setAssigningAuthorityName(String)` (Java method), 241
- `setAttribute(String)` (Java method), 278
- `setAttributeOptionCombo(String)` (Java method), 282
- `setAttributes(List)` (Java method), 269, 287, 293, 413
- `setAttributes(Map)` (Java method), 166
- `setAuthRequired(boolean)` (Java method), 346
- `setBasedOnAbsoluteWindows(boolean)` (Java method), 462
- `setBaseUrl(String)` (Java method), 134
- `setBatchJobDtoList(List)` (Java method), 109
- `setBatchJobId(Integer)` (Java method), 105
- `setBatchJobStatusId(Integer)` (Java method), 104
- `setBeginEffectiveDate(DateTime)` (Java method), 235
- `setBeginningHour(DateTime)` (Java method), 235
- `setBirthDateEstimated(Boolean)` (Java method), 413
- `setBufferOverDosageTimeInMinutes(int)` (Java method), 437, 442, 443
- `setBundleContext(BundleContext)` (Java method), 265
- `setCallbackUrl(String)` (Java method), 332
- `setCampaign(CampaignRecord)` (Java method), 361, 363
- `setCampaignName(String)` (Java method), 355, 360
- `setCampaignType(CampaignType)` (Java method), 364
- `setCaseDataXmlns(String)` (Java method), 145, 172
- `setCaseId(String)` (Java method), 139, 141, 143, 145, 172, 186, 191
- `setCaseIds(List)` (Java method), 162
- `setCaseInfoList(List)` (Java method), 146
- `setCaseName(String)` (Java method), 139, 145, 159, 170, 173, 188
- `setCaseProperties(List)` (Java method), 156

setCaseType(String) (Java method), 139, 145, 156, 159, 170, 173, 188, 191

setCategoryOptionCombo(String) (Java method), 281, 282

setCertificate(String) (Java method), 218

setChapterIdentifier(String) (Java method), 384

setChapterName(String) (Java method), 383

setChapters(List) (Java method), 386

setChecksum(String) (Java method), 117

setClose(boolean) (Java method), 146

setClosed(boolean) (Java method), 139, 141

setCloseElement(CloseElement) (Java method), 186

setCloseTask(CloseTask) (Java method), 143

setCode(String) (Java method), 214, 241, 278, 279

setCodedType(CodedType) (Java method), 218, 219, 257

setCodedTypes(Set) (Java method), 224, 237, 248

setCodingScheme(String) (Java method), 214

setCommcareAppId(String) (Java method), 148

setCommcareBaseUrl(String) (Java method), 147, 169

setCommcareDomain(String) (Java method), 147, 169

setComment(String) (Java method), 281, 282

setCommonNames(List) (Java method), 233, 245

setCommunicationProtocol(CommunicationProtocol) (Java method), 216

setCompleteDate(String) (Java method), 282

setCompletionTime(DateTime) (Java method), 383

setComponent(String) (Java method), 210

setConceptClass(String) (Java method), 400

setConceptName(String) (Java method), 409

setConceptNameType(String) (Java method), 401

setConfig(String) (Java method), 496, 500, 518

setConfigName(String) (Java method), 139, 148, 150, 152, 173, 343

setConfigs(List) (Java method), 137, 511

setConfigService(ConfigService) (Java method), 267

setContact(Object) (Java method), 239

setContactPoints(Set) (Java method), 224, 237, 244, 255

setContacts(Set) (Java method), 224, 237

setContent(Byte[]) (Java method), 117

setContent(String) (Java method), 329, 387

setContentId(Integer) (Java method), 330, 331

setContentTypes(String) (Java method), 117, 329

setCoordinate(CoordinateDto) (Java method), 285

setCoordinateSystem(String) (Java method), 230

setCount(int) (Java method), 453, 505

setCountry(String) (Java method), 406

setCountyDistrict(String) (Java method), 407

setCourseIdentifier(String) (Java method), 384

setCourseName(String) (Java method), 383

setCreated(DateTime) (Java method), 256

setCreated(String) (Java method), 279, 293

setCreatedBy(String) (Java method), 108

setCreateElement(CreateElement) (Java method), 186

setCreateTask(CreateTask) (Java method), 143

setCreateTime(DateTime) (Java method), 108

setCreationDate(DateTime) (Java method), 207

setCreator(OpenMRSUser) (Java method), 403

setCredentialIssueDate(DateTime) (Java method), 219

setCredentialRenewalDate(DateTime) (Java method), 219

setCredentials(Set) (Java method), 237, 248

setCron(String) (Java method), 363, 366

setCronExpression(String) (Java method), 104, 108, 110

setCSD(CSD) (Java method), 231

setCurrentMilestoneName(String) (Java method), 455, 479

setCustomParams(Map) (Java method), 518

setData(Map) (Java method), 84, 459

setDataElement(DataElementDto) (Java method), 291

setDataElement(String) (Java method), 281

setDataElements(List) (Java method), 270

setDataSet(String) (Java method), 282

setDataSetComplete(boolean) (Java method), 283

setDataType(String) (Java method), 400

setDataValueCount(ImportCountDto) (Java method), 283

setDataValues(List) (Java method), 282, 285

setDataXmlns(String) (Java method), 186

setDate(DateTime) (Java method), 403, 409

setDate(LocalDate) (Java method), 358, 363

setDate(String) (Java method), 112

setDateClosed(String) (Java method), 139, 141

setDateModified(String) (Java method), 141, 143, 145, 173, 186

setDateOfBirth(DateTime) (Java method), 244, 413

setDateOfEnrollment(String) (Java method), 287

setDateOfIncident(String) (Java method), 287

setDateOpened(String) (Java method), 140, 170

setDateTime(DateTime) (Java method), 84

setDaysOfTheWeek(List) (Java method), 235

setDaysOfWeek(List) (Java method), 368

setDbEventLoggers(List) (Java method), 319

setDeathDate(DateTime) (Java method), 413

setDefault(String) (Java method), 199

setDefaultConfigName(String) (Java method), 137, 511

setDefaultDbEventLogger(DbEventLogger) (Java method), 319

setDefaultDbToLogConverter(DefaultDbToLogConverter) (Java method), 319

setDefaultPhoneNumber(String) (Java method), 157

setDeleted(int) (Java method), 288

setDeliverTime(Time) (Java method), 360

setDeliveryStatus(DeliveryStatus) (Java method), 500

setDeliveryStatus(String) (Java method), 496

setDeliveryTime(DateTime) (Java method), 518

setDemographic(Person) (Java method), 249

setDependentObservations(Set) (Java method), 409

setDescription(String) (Java method), 84, 283, 405

setDhis2Name(String) (Java method), 272

- setDhis2Uuid(String) (Java method), 272
- setDisplay(String) (Java method), 400
- setDisplayName(String) (Java method), 278
- setDistributionServiceDelegate(DistributionServiceDelegate) (Java method), 339
- setDistributionStatusCode(String) (Java method), 329
- setDistributionStatusId(Integer) (Java method), 329
- setDomain(String) (Java method), 134, 140, 141, 149
- setDosages(Set) (Java method), 449
- setDosageTime(Time) (Java method), 445
- setDueDate(String) (Java method), 285
- setDueStart(DateTime) (Java method), 479
- setEarliestStart(DateTime) (Java method), 479
- setElementName(String) (Java method), 167
- setEmail(String) (Java method), 157
- setEncounterId(String) (Java method), 403
- setEncounterType(String) (Java method), 403
- setEndDate(LocalDate) (Java method), 447
- setEndEffectiveDate(DateTime) (Java method), 235
- setEndingHour(DateTime) (Java method), 235
- setEndKey(String) (Java method), 312
- setEndValue(Object) (Java method), 312
- setEnrolledOn(DateTime) (Java method), 456
- setEnrollment(String) (Java method), 285
- setEnrollmentDate(LocalDate) (Java method), 481
- setEnrollmentDateTime(DateTime) (Java method), 479
- setEnrollmentStatus(String) (Java method), 285
- setEnrollmentTime(Time) (Java method), 481
- setEntityID(String) (Java method), 208, 211, 259
- setEquipment(String) (Java method), 218
- setErrorCode(String) (Java method), 100, 326
- setErrorMessage(String) (Java method), 100, 326, 496, 500
- setEvent(String) (Java method), 285
- setEventDate(String) (Java method), 285
- setEventStrategy(String) (Java method), 135, 169
- setEventSubjects(List) (Java method), 314
- setExcludeLastFooter(Boolean) (Java method), 508
- setExcludes(List) (Java method), 314
- setExclusions(List) (Java method), 313
- setExtensions(Set) (Java method), 219, 224, 227, 228, 238, 249, 251, 253, 255, 257
- setExternalId(String) (Java method), 85, 92, 94, 355, 360, 383, 385, 449, 456, 479, 481
- setExternalName(String) (Java method), 272
- setFacilities(Set) (Java method), 226
- setFacility(OpenMRSFacility) (Java method), 403, 411
- setFacilityDirectory(FacilityDirectory) (Java method), 213
- setFacilityId(String) (Java method), 407
- setFacilityOrganizations(Set) (Java method), 224
- setFacilityOrganizationSet(Set) (Java method), 224
- setFailureCount(Integer) (Java method), 518
- setField(String, String) (Java method), 343
- setFields(Map) (Java method), 150
- setFieldValues(Map) (Java method), 140, 145, 170, 173
- setFirstName(String) (Java method), 157, 413
- setFixtureType(String) (Java method), 150
- setFlags(List) (Java method), 314, 315
- setFloating(boolean) (Java method), 453
- setFollowup(boolean) (Java method), 285, 287
- setForename(String) (Java method), 233, 245
- setForm(FormValueElement) (Java method), 152
- setFormats(List) (Java method), 361, 363
- setFormId(String) (Java method), 162
- setFormNames(Map) (Java method), 160
- setFormSchemas(List) (Java method), 156
- setForwardAppStructure(boolean) (Java method), 169
- setForwardCases(boolean) (Java method), 135, 169
- setForwardForms(boolean) (Java method), 135, 169
- setForwardFormStubs(boolean) (Java method), 169
- setForwardSchema(boolean) (Java method), 135
- setForwardStubs(boolean) (Java method), 135
- setFreeBusyURI(String) (Java method), 228, 253
- setFrom(String) (Java method), 343
- setFromStatus(AppointmentStatus) (Java method), 94
- setFulfillmentDateTime(DateTime) (Java method), 460
- setFulfillments(List) (Java method), 456
- setGender(String) (Java method), 244, 413
- setGeocode(Geocode) (Java method), 225
- setGroups(List) (Java method), 157
- setHonorific(String) (Java method), 233, 245
- setHostName(String) (Java method), 100, 326
- setHref(String) (Java method), 279
- setHttpAgentImpl(HttpAgent) (Java method), 339, 340
- setHubBaseUrl(String) (Java method), 335, 336
- setHubDistributionStatusId(Integer) (Java method), 331
- setHubSubscriptionId(Integer) (Java method), 331
- setHubSubscriptionStatusId(Integer) (Java method), 332
- setHubTopicId(Integer) (Java method), 330, 332
- setId(Long) (Java method), 85, 207, 360, 363, 364, 445, 449, 456
- setId(String) (Java method), 149, 150, 152, 158, 279, 400, 407, 413
- setIdentifier(String) (Java method), 414
- setIgnored(int) (Java method), 288
- setIgnoredStatusFields(List) (Java method), 346
- setImportCount(ImportCountDto) (Java method), 286
- setImportCountDto(ImportCountDto) (Java method), 283
- setImported(int) (Java method), 288
- setIncludeArchived(Boolean) (Java method), 189
- setIncludes(List) (Java method), 315
- setInclusions(List) (Java method), 313
- setIndex(int) (Java method), 453
- setIndexElement(IndexTask) (Java method), 186
- setIndexNodeName(String) (Java method), 191
- setIndexTask(IndexTask) (Java method), 143
- setIndices(List) (Java method), 167

setIndices(Map) (Java method), 140, 142  
 setInterval(Period) (Java method), 453  
 setIsDead(Boolean) (Java method), 413  
 setIssuingAuthority(String) (Java method), 219  
 setJobContent(Byte[]) (Java method), 104  
 setJobExecutionHistoryList(List) (Java method), 111  
 setJobId(long) (Java method), 108  
 setJobName(String) (Java method), 104, 108, 110  
 setJsonExtraParamsList(List) (Java method), 346  
 setJsonRequest(boolean) (Java method), 346  
 setJsonResponse(boolean) (Java method), 346  
 setKeyValuePairsPresent(Map) (Java method), 315  
 setLabel(String) (Java method), 161  
 setLang(String) (Java method), 242, 245  
 setLanguage(String) (Java method), 117, 119  
 setLanguages(List) (Java method), 361, 363  
 setLanguages(Set) (Java method), 225, 229, 238, 249, 253  
 setLanguages(String) (Java method), 125  
 setLastModified(String) (Java method), 216  
 setLastModified(XMLGregorianCalendar) (Java method), 231  
 setLastName(String) (Java method), 158, 413  
 setLastUpdated(DateTime) (Java method), 108  
 setLastUpdated(String) (Java method), 279  
 setLastUpdatedBy(String) (Java method), 108  
 setLateStart(DateTime) (Java method), 479  
 setLatitude(Double) (Java method), 280  
 setLatitude(double) (Java method), 230  
 setLeaseSeconds(Integer) (Java method), 332  
 setLessonIdentifier(String) (Java method), 385  
 setLessonName(String) (Java method), 383  
 setLessons(List) (Java method), 385  
 setLimit(int) (Java method), 155  
 setLocale(String) (Java method), 401  
 setLongitude(Double) (Java method), 280  
 setLongitude(double) (Java method), 230  
 setMappings(List) (Java method), 313, 315  
 setMappings(LogMappings) (Java method), 311  
 setMaxDuration(Period) (Java method), 358  
 setMaxDuration(String) (Java method), 358, 364  
 setMaxRetries(Integer) (Java method), 508  
 setMaxStart(DateTime) (Java method), 479  
 setMd5(String) (Java method), 152  
 setMedicines(Set) (Java method), 445  
 setMessage(String) (Java method), 518  
 setMessageCampaignsJsonFile(String) (Java method), 375  
 setMessageContent(String) (Java method), 496, 500  
 setMessageKey(String) (Java method), 361, 363  
 setMessageNature(String) (Java method), 192  
 setMessageRecords(List) (Java method), 359  
 setMessages(List) (Java method), 359, 365  
 setMessageText(String) (Java method), 192  
 setMessageType(CampaignType) (Java method), 363  
 setMeta(CommcareMetadataJson) (Java method), 153  
 setMetadata(Map) (Java method), 117, 119, 152, 456, 479, 481  
 setMetadataInfo(CommcareMetadataInfo) (Java method), 146  
 setMiddleName(String) (Java method), 413  
 setMilestone(Milestone) (Java method), 461  
 setMilestoneName(String) (Java method), 460  
 setMilestoneWindow(MilestoneWindow) (Java method), 453  
 setModificationDate(DateTime) (Java method), 207  
 setModules(List) (Java method), 148  
 setMotechId(String) (Java method), 411, 497, 500, 518  
 setMotechTimestamp(String) (Java method), 343  
 setName(OpenMRSConceptName) (Java method), 400  
 setName(String) (Java method), 85, 117, 119, 125, 135, 268–272, 279, 346, 348, 359, 361, 363, 365, 387, 398, 401, 405, 407, 447, 508, 510  
 setName(WindowName) (Java method), 461  
 setNames(List) (Java method), 400  
 setNames(Set) (Java method), 229, 244, 253  
 setNewPasswordForUser(String) (Java method), 434  
 setNextPage(String) (Java method), 289  
 setNextPageQueryString(String) (Java method), 155  
 setNotificationTime(DateTime) (Java method), 330  
 setNumber(String) (Java method), 220  
 setObjects(List) (Java method), 153  
 setObservationId(String) (Java method), 409  
 setObservations(Set) (Java method), 403  
 setOffset(int) (Java method), 155  
 setOffset(Period) (Java method), 453  
 setOpenFlag(boolean) (Java method), 235  
 setOperatingHours(Set) (Java method), 225, 229, 251, 253  
 setOptional(boolean) (Java method), 312  
 setOptions(List) (Java method), 160  
 setOrganisationUnits(List) (Java method), 290  
 setOrganization(UniqueID) (Java method), 253  
 setOrganizationDirectory(OrganizationDirectory) (Java method), 213  
 setOrganizations(Set) (Java method), 240  
 setOrgUnit(String) (Java method), 281, 282, 285, 293  
 setOrgUnitName(String) (Java method), 285  
 setOtherIDs(Set) (Java method), 225, 238, 249  
 setOtherNames(Set) (Java method), 225, 233, 238, 245  
 setOutgoingCallMethod(HttpMethod) (Java method), 346  
 setOutgoingCallUriTemplate(String) (Java method), 346  
 setOwnerId(String) (Java method), 140, 145, 159, 170, 173, 188  
 setPage(int) (Java method), 289  
 setPage(Integer) (Java method), 126  
 setPageCount(int) (Java method), 290  
 setPageNumber(Integer) (Java method), 190  
 setPager(PagerDto) (Java method), 289



- setPageSize(Integer) (Java method), 190
- setParameterName(String) (Java method), 106
- setParameters(Map) (Java method), 109, 307
- setParameterValue(String) (Java method), 106
- setParamsMap(Map) (Java method), 110, 111
- setParent(UniqueID) (Java method), 238
- setParentOrganization(String) (Java method), 238
- setPassPercentage(double) (Java method), 388
- setPassword(String) (Java method), 134, 147, 169, 298, 321, 346
- setPatient(OpenMRSPatient) (Java method), 403
- setPatientId(String) (Java method), 409, 411
- setPeriod(Period) (Java method), 461
- setPeriod(String) (Java method), 281, 282
- setPerson(OpenMRSPerson) (Java method), 411, 414, 416
- setPerson(Person) (Java method), 239
- setPersonId(String) (Java method), 414
- setPhoneNumber(String) (Java method), 497, 501
- setPhoneNumbers(List) (Java method), 158
- setPillWindowInHours(int) (Java method), 443
- setPreferredAlertTime(Time) (Java method), 456, 479, 481
- setPreferredName(String) (Java method), 414
- setPreviousPageQueryString(String) (Java method), 155
- setPrevPage(String) (Java method), 290
- setPrimaryName(String) (Java method), 225, 238
- setPriority(Long) (Java method), 85
- setProgram(ProgramDto) (Java method), 292
- setProgram(String) (Java method), 270, 285, 287
- setProgramStage(ProgramStageDto) (Java method), 291
- setProgramStage(String) (Java method), 285
- setProgramStageDataElements(List) (Java method), 292
- setProgramStages(List) (Java method), 290
- setProgramTrackedEntityAttributes(List) (Java method), 291
- setProgress(Map) (Java method), 385
- setProps(List) (Java method), 508
- setProvider(OpenMRSPProvider) (Java method), 403
- setProviderDirectory(ProviderDirectory) (Java method), 213
- setProviderEntityID(String) (Java method), 239
- setProviderFacilities(Set) (Java method), 249
- setProviderFacilitySet(Set) (Java method), 249
- setProviderId(String) (Java method), 414, 497, 501, 518
- setProviderOrganization(String) (Java method), 253
- setProviderOrganizations(Set) (Java method), 249
- setProviderOrganizationSet(Set) (Java method), 249
- setProviders(Set) (Java method), 250
- setProviderStatus(String) (Java method), 497, 501
- setProviderTimestamp(String) (Java method), 343
- setPurpose(String) (Java method), 218
- setQuestion(String) (Java method), 388
- setQuestionLabel(String) (Java method), 161
- setQuestionRepeat(String) (Java method), 161
- setQuestions(List) (Java method), 160, 389
- setQuestionTag(String) (Java method), 161
- setQuestionValue(String) (Java method), 161
- setQuiz(Quiz) (Java method), 385
- setQuizName(String) (Java method), 383
- setQuizScore(double) (Java method), 383
- setReason(String) (Java method), 102, 328
- setReceivedOn(String) (Java method), 152, 162
- setReceivedOnEnd(DateTime) (Java method), 190
- setReceivedOnStart(DateTime) (Java method), 190
- setRecipients(List) (Java method), 519
- setRecord(Record) (Java method), 211
- setRecords(List) (Java method), 505
- setReference(String) (Java method), 286
- setReferenceDate(LocalDate) (Java method), 355, 360, 481
- setReferenceDateTime(DateTime) (Java method), 479
- setReferenceTime(Time) (Java method), 482
- setRegion(String) (Java method), 407
- setRegistration(boolean) (Java method), 269, 270, 291
- setReminderInterval(Period) (Java method), 92
- setReminderStartTime(DateTime) (Java method), 92
- setRepeatEvery(String) (Java method), 363
- setRepeatInterval(Period) (Java method), 370
- setRepeatInterval(String) (Java method), 370
- setRepeatIntervalInMinutes(int) (Java method), 443
- setRepeatOn(List) (Java method), 363
- setResources(String, List) (Java method), 289
- setResourceUi(String) (Java method), 158
- setResourceUri(String) (Java method), 148, 149, 152
- setResponseLastCapturedDate(LocalDate) (Java method), 445
- setRetryCount(String) (Java method), 339, 340
- setRetryInterval(String) (Java method), 339, 340
- setRows(Integer) (Java method), 126
- setSchedule(Schedule) (Java method), 456, 459
- setScheduleDetails(DailyScheduleDetails) (Java method), 449
- setScheduleName(String) (Java method), 456, 479, 482
- setSchedulerEnabled(boolean) (Java method), 216
- setSecret(String) (Java method), 332
- setSecurityRole(String) (Java method), 416
- setSendReminders(boolean) (Java method), 92
- setSerializedModules(String) (Java method), 148
- setServerDateModified(String) (Java method), 140, 142
- setServerDateOpened(String) (Java method), 140, 142
- setServerModifiedOn(String) (Java method), 145, 173
- setServerURI(String) (Java method), 298
- setServiceDirectory(ServiceDirectory) (Java method), 213
- setServices(Set) (Java method), 227, 251, 258
- setServicesMapString(String) (Java method), 346
- setSettings(SettingsFacade) (Java method), 375

setSidx(String) (Java method), 126  
setSingleEvent(boolean) (Java method), 269, 291  
setSmsDirection(SmsDirection) (Java method), 501  
setSmsDirection(String) (Java method), 497  
setSord(String) (Java method), 126  
setSourceDirectory(String) (Java method), 211, 256  
setSpecializations(Set) (Java method), 238  
setSpecialties(Set) (Java method), 249  
setSplitFooter(String) (Java method), 509  
setSplitHeader(String) (Java method), 509  
setStages(List) (Java method), 269  
setStartDate(LocalDate) (Java method), 447  
setStartDate(String) (Java method), 216  
setStartingMilestoneName(String) (Java method), 482  
setStartKey(String) (Java method), 312  
setStartOfSchedule(DateTime) (Java method), 456  
setStartTime(DateTime) (Java method), 383  
setStartTime(String) (Java method), 361, 364  
setStartTime(Time) (Java method), 355, 361  
setStartValue(Object) (Java method), 312  
setState(ActivityState) (Java method), 383  
setState(CourseUnitState) (Java method), 387  
setStateProvince(String) (Java method), 407  
setStatus(AlertStatus) (Java method), 85  
setStatus(AppointmentStatus) (Java method), 92  
setStatus(CampaignEnrollmentStatus) (Java method), 360  
setStatus(DhisStatus) (Java method), 283, 286  
setStatus(EnrollmentStatus) (Java method), 456  
setStatus(int) (Java method), 192  
setStatus(String) (Java method), 109, 211, 256, 285, 479  
setStatusCode(HttpStatus) (Java method), 418  
setStatusFieldMapString(String) (Java method), 346  
setStoredBy(String) (Java method), 285  
setStream(Boolean) (Java method), 126  
setStreamContentService(StreamContentService) (Java method), 123  
setString(Boolean) (Java method), 126  
setStringContentService(StringContentService) (Java method), 123  
setSubElements(Multimap) (Java method), 167  
setSubject(String) (Java method), 308  
setSubjects(List) (Java method), 315  
setSubscriptionStatusCode(String) (Java method), 332  
setSubscriptionStatusId(Integer) (Java method), 332  
setSuffix(String) (Java method), 233, 246  
setSurname(String) (Java method), 233, 246  
setSystemId(String) (Java method), 416  
setTemplateName(String) (Java method), 343, 509  
setTimeOffset(Period) (Java method), 369  
setTimeOffset(String) (Java method), 364, 369  
setTimePeriod(String) (Java method), 216  
setTimePeriodMultiplier(String) (Java method), 216  
setTimeStamp(Date) (Java method), 100, 326  
setTimeStamp(DateTime) (Java method), 308  
setTimestamp(DateTime) (Java method), 501  
setTimestamp(String) (Java method), 497  
Settings (Java class), 298  
Settings() (Java constructor), 298  
Settings(String, String, String) (Java constructor), 298  
SettingsDTO (Java class), 336  
SettingsDto (Java class), 168  
SettingsDTO() (Java constructor), 336  
SettingsDto() (Java constructor), 168  
SettingsService (Java interface), 299  
setTo(String) (Java method), 344  
setTopicUrl(String) (Java method), 333  
setToStatus(AppointmentStatus) (Java method), 94  
setTotal(int) (Java method), 290  
setTotalCount(int) (Java method), 155  
setTrackedEntity(String) (Java method), 293  
setTrackedEntity(TrackedEntity) (Java method), 270  
setTrackedEntity(TrackedEntityDto) (Java method), 291  
setTrackedEntityAttribute(TrackedEntityAttributeDto) (Java method), 292  
setTrackedEntityInstance(String) (Java method), 285, 288, 293  
setType(String) (Java method), 149, 152, 209, 221, 278  
setUiversion(String) (Java method), 152  
setUpdated(DateTime) (Java method), 256  
setUpdated(int) (Java method), 288  
setUpdateElement(UpdateTask) (Java method), 186  
setUpdateTask(UpdateTask) (Java method), 143  
setUrl(String) (Java method), 149  
setUrn(String) (Java method), 221  
setUserData(Map) (Java method), 158  
setUserId(String) (Java method), 140, 142, 143, 145, 173, 186, 416  
setUserName(String) (Java method), 416  
setUsername(String) (Java method), 134, 147, 158, 169, 299, 321, 346  
setUuid(String) (Java method), 268–272, 400, 405  
setValue(String) (Java method), 119, 161, 167, 210, 214, 242, 278, 281, 348, 398, 510  
setValue(T) (Java method), 409  
setValues(Collection) (Java method), 168  
setVersion(String) (Java method), 140, 142, 149, 152  
setVisitedDate(DateTime) (Java method), 92  
setXformIds(List) (Java method), 140, 142  
setXmlns(String) (Java method), 143, 186, 190  
setXmlUrl(String) (Java method), 216  
shouldForwardAppStructure() (Java method), 169  
shouldForwardCases() (Java method), 169  
shouldForwardForms() (Java method), 169  
shouldForwardFormStubs() (Java method), 169  
shouldIgnoreField(String) (Java method), 347  
SmsAuditService (Java interface), 493  
SmsAuditServiceImpl (Java class), 494

- SmsAuditServiceImpl(SmsRecordsDataService) (Java constructor), 494
  - SmsDirection (Java enum), 494
  - SmsEventParams (Java class), 521
  - SmsEvents (Java class), 524
  - SmsEventSubjects (Java class), 523
  - SmsLoggingRecord (Java class), 495
  - SmsLoggingRecord(SmsRecord) (Java constructor), 495
  - SmsLoggingRecords (Java class), 497
  - SmsLoggingRecords(Integer, Integer, Long, SmsRecords) (Java constructor), 497
  - SmsRecord (Java class), 498
  - SmsRecord() (Java constructor), 498
  - SmsRecord(String, SmsDirection, String, String, DateTime, DeliveryStatus, String, String, String, String) (Java constructor), 498
  - SmsRecords (Java class), 504
  - SmsRecords() (Java constructor), 505
  - SmsRecords(int, List) (Java constructor), 505
  - SmsRecordsDataService (Java interface), 505
  - SmsRecordSearchCriteria (Java class), 501
  - SmsService (Java interface), 519
  - SmsServiceImpl (Java class), 519
  - SmsServiceImpl(EventRelay, MotechSchedulerService, TemplateService, ConfigService, SmsRecordsDataService) (Java constructor), 519
  - SOAP (Java field), 215
  - Stage (Java class), 270
  - STAGE (Java field), 275
  - STAGE\_EVENT (Java field), 303
  - StageActionBuilder (Java class), 304
  - StageDataService (Java interface), 277
  - StageService (Java interface), 299
  - START\_TIME (Java field), 353
  - STARTED (Java field), 384
  - StartOfWindowCriterion (Java class), 468
  - StartOfWindowCriterion(WindowName, DateTime, DateTime) (Java constructor), 468
  - STATUS (Java field), 87
  - status(AlertStatus) (Java method), 81
  - STATUS\_UPDATER (Java field), 87
  - StatusCriterion (Java class), 468
  - StatusCriterion(EnrollmentStatus) (Java constructor), 468
  - stopAll(CampaignEnrollmentsQuery) (Java method), 379, 381
  - stopAll(CampaignEnrollmentsQuery, boolean) (Java method), 379, 381
  - StreamContent (Java class), 115
  - StreamContent() (Java constructor), 115
  - StreamContent(String, String, Byte[], String, String) (Java constructor), 116
  - StreamContentService (Java interface), 123
  - StringContent (Java class), 117
  - StringContent() (Java constructor), 117
  - StringContent(String, String, String) (Java constructor), 118
  - StringContent(String, String, String, Map) (Java constructor), 118
  - StringContentService (Java interface), 123
  - SUB\_ELEMENTS (Java field), 178
  - SUBSCRIBE (Java field), 335
  - subscribe(String, Modes, String, String, String) (Java method), 337, 340
  - SUBSCRIPTION\_NOT\_FOUND (Java field), 325
  - SubscriptionService (Java interface), 337
  - SubscriptionServiceImpl (Java class), 339
  - SubscriptionServiceImpl(HubTopicMDSService, HubSubscriptionMDSService, HttpAgent) (Java constructor), 340
  - SubscriptionStatusLookup (Java enum), 336
  - SUCCESS (Java field), 286, 335
  - SUNDAY (Java field), 220
  - Sunday (Java field), 367
  - sync() (Java method), 299, 305
  - SYNC\_HTTP\_REQUEST\_RET\_TYPE (Java field), 322
  - SyncService (Java interface), 299
  - systemId(String) (Java method), 416
- ## T
- TasksChannelServiceInstance (Java class), 205
  - TasksChannelServiceInstance(Object, ChannelRequestBuilder) (Java constructor), 205
  - TasksService (Java interface), 300
  - Template (Java class), 347
  - Template(String, String) (Java constructor), 347
  - TemplateNotFoundException (Java class), 349
  - TemplateNotFoundException(String) (Java constructor), 349
  - TemplateNotFoundException(String, Throwable) (Java constructor), 349
  - TemplateService (Java interface), 351, 519
  - TemplateServiceImpl (Java class), 520
  - TemplateServiceImpl(SettingsFacade) (Java constructor), 520
  - THURSDAY (Java field), 220
  - Thursday (Java field), 367
  - timesPillRemainderWillBeSent() (Java method), 449
  - TIMESTAMP (Java field), 523
  - TIMESTAMP\_DATETIME (Java field), 523
  - TIMESTAMP\_TIME (Java field), 523
  - toCampaign() (Java method), 365
  - toDate() (Java method), 76
  - todayDosageTime() (Java method), 445
  - toEnrollment() (Java method), 456
  - toggleReminders(String, boolean) (Java method), 97
  - toMotechEvent() (Java method), 174–176, 470, 472
  - toMotechEventWithData() (Java method), 173
  - toMotechEventWithoutData() (Java method), 173

TOPIC\_NOT\_FOUND (Java field), 325  
toString() (Java method), 82, 85, 117, 119, 131, 133, 208–211, 213, 215, 218, 220, 221, 225–227, 229, 230, 233, 235, 238–242, 244, 246, 249, 250, 252, 253, 255–258, 344, 347, 348, 355, 364, 365, 409, 460, 497, 498, 501, 502, 519  
TRACKED\_ENTITY\_TYPE (Java field), 303  
TrackedEntity (Java class), 271  
TrackedEntity() (Java constructor), 271  
TrackedEntity(String, String) (Java constructor), 271  
TrackedEntityAttribute (Java class), 271  
TrackedEntityAttribute() (Java constructor), 271  
TrackedEntityAttribute(String, String) (Java constructor), 271  
TrackedEntityAttributeDataService (Java interface), 277  
TrackedEntityAttributeDto (Java class), 292  
TrackedEntityAttributeService (Java interface), 300  
TrackedEntityDataService (Java interface), 277  
TrackedEntityDto (Java class), 292  
TrackedEntityInstanceDto (Java class), 293  
TrackedEntityInstanceMapping (Java class), 272  
TrackedEntityInstanceMapping() (Java constructor), 272  
TrackedEntityInstanceMapping(String, String) (Java constructor), 272  
TrackedEntityInstanceMappingDataService (Java interface), 277  
TrackedEntityInstanceMappingException (Java class), 300  
TrackedEntityInstanceMappingException(String) (Java constructor), 300  
TrackedEntityInstanceMappingException(String, Throwable) (Java constructor), 300  
TrackedEntityInstanceMappingService (Java interface), 300  
TrackedEntityService (Java interface), 301  
TrackedSchedulesJsonReader (Java interface), 475  
TrackedSchedulesJsonReaderImpl (Java class), 476  
TrackedSchedulesJsonReaderImpl() (Java constructor), 476  
transliterate(String) (Java method), 525  
TransliterationService (Java interface), 525  
TransliterationServiceImpl (Java class), 525  
triggerJob(String) (Java method), 114  
TUESDAY (Java field), 220  
Tuesday (Java field), 367

## U

UI\_CONFIG (Java field), 352  
understands(SoapHeaderElement) (Java method), 267  
unenroll(Enrollment) (Java method), 482, 491  
unenroll(String, List) (Java method), 488, 492  
unenroll(String, String) (Java method), 379, 381, 488, 492  
UNENROLL\_USER\_SUBJECT (Java field), 353  
UNENROLLED (Java field), 458

UNENROLLED\_USER\_SUBJECT (Java field), 353  
UnenrolledUserEvent (Java class), 472  
UnenrolledUserEvent(Map) (Java constructor), 472  
UnenrolledUserEvent(String, String) (Java constructor), 472  
UniqueID (Java class), 258  
UniqueID() (Java constructor), 258  
UniqueID(String) (Java constructor), 258  
UNKNOWN (Java field), 344  
UnknownCampaignTypeException (Java class), 374  
UnknownCampaignTypeException(String) (Java constructor), 374  
unmarshall(String, URL, Class) (Java method), 266  
unregister(CampaignEnrollment) (Java method), 377  
unregister(String, String) (Java method), 377  
UNSCHEDULE\_JOB\_FAILED (Java field), 99  
unscheduleAllAlerts(Enrollment) (Java method), 489  
UNSCHEDULED (Java field), 95  
unscheduleJob(String) (Java method), 113  
unscheduleJobs(PillRegimen) (Java method), 450  
unscheduleJobsForEnrollment(CampaignEnrollment) (Java method), 379, 381  
unscheduleMessageJob(CampaignMessageRecord) (Java method), 379, 381  
unscheduleMilestoneDefaultedJob(Enrollment) (Java method), 490  
UNSUBSCRIBE (Java field), 335  
update(CSD) (Java method), 262  
update(Facility) (Java method), 263  
update(Long, UpdateCriteria) (Java method), 78, 89  
update(Organization) (Java method), 264  
update(Provider) (Java method), 264  
update(Service) (Java method), 265  
update(Set) (Java method), 263–265  
update(String, UpdateCriteria) (Java method), 78, 89  
update(T) (Java method), 297  
UPDATE\_DATE (Java field), 207  
UPDATE\_PROGRAM\_STAGE (Java field), 276  
updateActivity(ActivityRecord) (Java method), 392  
updateAppointment(Appointment) (Java method), 98  
updateAppointments(List) (Java method), 98  
updateBookmark(Bookmark) (Java method), 393  
updateChannel() (Java method), 300  
updateChapter(Chapter) (Java method), 397  
updateConcept(OpenMRSConcept) (Java method), 424  
updateConfig(Config) (Java method), 137  
updateConfigs(Configs) (Java method), 513, 514  
updateConfigs(List) (Java method), 262, 267, 350  
updateCourse(Course) (Java method), 397  
UpdateCriteria (Java class), 80  
UpdateCriteria() (Java constructor), 80  
UpdateCriterion (Java enum), 87  
UPDATED\_CONCEPT\_SUBJECT (Java field), 422  
UPDATED\_ENCOUNTER\_SUBJECT (Java field), 423



- UPDATED\_FACILITY\_SUBJECT (Java field), 423
  - UPDATED\_PATIENT\_SUBJECT (Java field), 423
  - UPDATED\_PERSON\_SUBJECT (Java field), 423
  - UPDATED\_PROVIDER\_SUBJECT (Java field), 423
  - updateDataForwardingRule(CommcareDataForwardingEndpointConfig) (Java method), 200
  - updateEnrollment(CampaignRequest, Long) (Java method), 380, 381
  - updateEnrollment(String, String, UpdateCriteria) (Java method), 488, 492
  - updateEnrollments(Long) (Java method), 380, 382
  - updateEventLoggingService(EventLoggingService) (Java method), 317, 319
  - updateFacility(OpenMRSFacility) (Java method), 427
  - updateFrom(CampaignRecord) (Java method), 365
  - updateJobProperty(String, Map) (Java method), 113
  - updateLesson(Lesson) (Java method), 397
  - updatePatient(OpenMRSPatient) (Java method), 431
  - updatePatient(OpenMRSPatient, String) (Java method), 430
  - updatePerson(OpenMRSPerson) (Java method), 432
  - updateQuiz(Quiz) (Java method), 397
  - updateResponseLastCapturedDate(LocalDate) (Java method), 445
  - updateSettings(Settings) (Java method), 299, 305
  - updateStatus(MotechEvent) (Java method), 88
  - UpdateTask (Java class), 169
  - updateTaskChannel() (Java method), 205
  - updateTasksInfo() (Java method), 205
  - updateTemplates(List) (Java method), 352
  - updateUser(OpenMRSUser) (Java method), 434
  - updateWithoutObs(OpenMRSEncounter) (Java method), 403
  - uploadCase(CaseTask) (Java method), 197
  - uploadCase(CaseTask, String) (Java method), 197
  - URL (Java field), 321
  - USER\_ENROLLED (Java field), 474
  - USER\_ID (Java field), 178, 423
  - USER\_KEY (Java field), 433
  - USER\_UNENROLLED (Java field), 474
  - UserAlreadyExistsException (Java class), 419
  - UserAlreadyExistsException(String) (Java constructor), 419
  - UserDeleteException (Java class), 419
  - UserDeleteException(String, Throwable) (Java constructor), 419
  - USERNAME (Java field), 321
  - userName(String) (Java method), 416
  - ValidationException (Java class), 450
  - ValidationException(String) (Java constructor), 450
  - VALUE (Java field), 178, 275, 303
  - value() (Java method), 398
  - VelocityResourceLoader (Java class), 205
  - verifyConfig(Config) (Java method), 199
  - VIEW\_SMS\_LOGS\_PERMISSION (Java field), 521
  - VISITED (Java field), 95
  - VOIDED\_OBSERVATION\_SUBJECT (Java field), 423
  - voidObservation(OpenMRSObservation, String) (Java method), 429
- ## W
- WEDNESDAY (Java field), 220
  - Wednesday (Java field), 367
  - window() (Java method), 464
  - WINDOW\_NAME (Java field), 474
  - windowElapsed(WindowName, DateTime) (Java method), 459
  - WindowName (Java enum), 463
  - withCampaignName(String) (Java method), 377
  - withConfig(String) (Java method), 503
  - withCreator(OpenMRSUser) (Java method), 404
  - withCreatorId(String) (Java method), 404
  - withCurrentMilestoneName(String) (Java method), 457
  - withDate(Date) (Java method), 404
  - withDeliverystatuses(Set) (Java method), 503
  - withEncounterType(String) (Java method), 404
  - withEnrolledOn(DateTime) (Java method), 457
  - withErrorMessage(String) (Java method), 503
  - withExternalId(String) (Java method), 377, 457
  - withFacility(OpenMRSFacility) (Java method), 404
  - withFacilityId(String) (Java method), 404
  - withFulfillments(List) (Java method), 457
  - withId(Long) (Java method), 457
  - withId(String) (Java method), 404
  - withMessageContent(String) (Java method), 503
  - withMetadata(Map) (Java method), 457
  - withMotechId(String) (Java method), 503
  - withObservations(Set) (Java method), 404
  - withPatient(OpenMRSPatient) (Java method), 404
  - withPatientId(String) (Java method), 404
  - withPhoneNumber(String) (Java method), 503
  - withPreferredAlertTime(Time) (Java method), 457
  - withProvider(OpenMRSPProvider) (Java method), 404
  - withProviderId(String) (Java method), 404, 504
  - withProviderStatus(String) (Java method), 504
  - withQueryParams(QueryParams) (Java method), 504
  - withSchedule(Schedule) (Java method), 457
  - withScheduleName(String) (Java method), 457
  - withSmsDirections(Set) (Java method), 504
  - withStartOfSchedule(DateTime) (Java method), 457
  - withStatus(EnrollmentStatus) (Java method), 457
  - withTimestamp(DateTime) (Java method), 504

`withTimestampRange(Range)` (Java method), [504](#)

## X

`XML_URL` (Java field), [207](#)

`XMLNS_ELEMENT` (Java field), [183](#)