
MOTECH Documentation

Release 0.24-SNAPSHOT

Grameen Foundation

October 05, 2014

1	About MOTECH	3
1.1	MOTECH Architecture Overview	3
1.2	What can MOTECH do?	3
2	Creating Applications	5
2.1	Introduction	5
2.2	Installing and Upgrading MOTECH	7
2.3	Configuration System	7
2.4	Modeling Data with MOTECH Data Services	10
2.5	Configuring Messaging: SMS	11
2.6	Configuring Messaging: IVR	11
2.7	Connecting MOTECH to OpenMRS	11
2.8	Integrating MOTECH with CommCare	11
2.9	Using the Tasks Module	11
2.10	Configuring Pill Reminders	11
2.11	Configuring Your MOTECH App to be HIPAA Compliant	11
2.12	Tour of MOTECH UI	11
3	Architecture and Technical Overviews	13
3.1	Core Architecture	13
3.2	Event and Scheduler Architecture	15
3.3	Modules Architecture	18
3.4	Data Services Architecture	18
3.5	Security Model	18
4	Modules	19
4.1	<i>Alerts</i>	19
4.2	<i>Appointments</i>	19
4.3	<i>Batch</i>	19
4.4	<i>CMS Lite</i>	19
4.5	<i>CommCare</i>	19
4.6	<i>Data Services Module</i>	19
4.7	<i>Email</i>	19
4.8	<i>Event Logging</i>	20
4.9	<i>Hindi Transliteration</i>	20
4.10	<i>Hub</i>	20
4.11	<i>Message Campaign</i>	20
4.12	<i>mTraining</i>	20

4.13	<i>OpenMRS</i>	20
4.14	<i>Pill Reminder</i>	20
4.15	<i>Schedule Tracking</i>	20
4.16	<i>Scheduler</i>	20
4.17	<i>IVR</i>	21
4.18	<i>SMS</i>	21
4.19	<i>Tasks</i>	21
5	Developing the MOTECH Platform	23
5.1	Setting up a Development Environment	23
5.2	Developing and Submitting a Patch	36
5.3	MOTECH Code Repositories	37
5.4	Commit Message Format	38
5.5	Coding Conventions	39
5.6	Code Review Workflow and Checklist	39
5.7	Authoring Documentation	39
5.8	Project Management	40
5.9	Release Strategy and Procedure	40
5.10	Browser Compatibility	40
6	Demos	41
6.1	Demo: Hello World	41
6.2	IVR Demos	48
6.3	Demo: Create a Message Campaign	62
6.4	Demo: Create a Care Schedule	63
6.5	Demo: Modeling a New System with Data Services	63
6.6	Demo: OpenMRS Schedule Tracking	63
7	Contribute	65
7.1	Development	65
7.2	Documentation	66
7.3	Translation	66
8	Release Notes	67
8.1	Current Version	67
8.2	Older Versions	67
9	Roadmap	73
9.1	Version 0.25 - Fall 2014	73
9.2	Version 0.26 - Late 2014	74
9.3	Version 1.0 - 2015	74
10	MOTECH Mailing Lists	75
10.1	MOTECH Developers	75
10.2	MOTECH Implementers	75
11	Javadoc	77
11.1	org.motechproject.admin.domain	77
11.2	org.motechproject.admin.messages	82
11.3	org.motechproject.admin.service	84
11.4	org.motechproject.bundle.extender	86
11.5	org.motechproject.commons.api	88
11.6	org.motechproject.commons.api.json	96
11.7	org.motechproject.commons.api.model	97
11.8	org.motechproject.commons.date	97

11.9	org.motechproject.commons.date.model	97
11.10	org.motechproject.commons.date.util	100
11.11	org.motechproject.commons.date.util.datetime	105
11.12	org.motechproject.commons.date.valueobjects	106
11.13	org.motechproject.commons.sql.service	107
11.14	org.motechproject.config.core	108
11.15	org.motechproject.config.core.constants	108
11.16	org.motechproject.config.core.domain	111
11.17	org.motechproject.config.core.filestore	118
11.18	org.motechproject.config.core.filters	119
11.19	org.motechproject.config.core.service	120
11.20	org.motechproject.config.domain	121
11.21	org.motechproject.config.monitor	123
11.22	org.motechproject.config.service	124
11.23	org.motechproject.email.builder	131
11.24	org.motechproject.email.contract	135
11.25	org.motechproject.email.domain	136
11.26	org.motechproject.email.service	142
11.27	org.motechproject.event	145
11.28	org.motechproject.event.listener	148
11.29	org.motechproject.event.listener.annotations	149
11.30	org.motechproject.event.queue	152
11.31	org.motechproject.mds.annotations	154
11.32	org.motechproject.mds.builder	157
11.33	org.motechproject.mds.config	161
11.34	org.motechproject.mds.domain	166
11.35	org.motechproject.mds.dto	210
11.36	org.motechproject.mds.enhancer	246
11.37	org.motechproject.mds.ex	247
11.38	org.motechproject.mds.ex.rest	258
11.39	org.motechproject.mds.filter	259
11.40	org.motechproject.mds.javassist	261
11.41	org.motechproject.mds.jdo	268
11.42	org.motechproject.mds.lookup	274
11.43	org.motechproject.mds.performance.domain	275
11.44	org.motechproject.mds.performance.service	275
11.45	org.motechproject.mds.performance.service.impl	278
11.46	org.motechproject.mds.query	280
11.47	org.motechproject.mds.repository	289
11.48	org.motechproject.mds.rest	296
11.49	org.motechproject.mds.service	299
11.50	org.motechproject.mds.service.impl	315
11.51	org.motechproject.mds.service.impl.history	324
11.52	org.motechproject.mds.util	329
11.53	org.motechproject.osgi.web	355
11.54	org.motechproject.osgi.web.domain	374
11.55	org.motechproject.osgi.web.exception	375
11.56	org.motechproject.osgi.web.ext	376
11.57	org.motechproject.osgi.web.service	378
11.58	org.motechproject.osgi.web.settings	380
11.59	org.motechproject.osgi.web.util	381
11.60	org.motechproject.scheduler.builder	383
11.61	org.motechproject.scheduler.contract	385
11.62	org.motechproject.scheduler.exception	396

11.63	org.motechproject.scheduler.factory	397
11.64	org.motechproject.scheduler.service	398
11.65	org.motechproject.scheduler.service.impl	402
11.66	org.motechproject.security.annotations	407
11.67	org.motechproject.security.authentication	407
11.68	org.motechproject.security.builder	410
11.69	org.motechproject.security.constants	412
11.70	org.motechproject.security.domain	416
11.71	org.motechproject.security.ex	425
11.72	org.motechproject.security.model	427
11.73	org.motechproject.security.repository	435
11.74	org.motechproject.security.service	441
11.75	org.motechproject.server.api	447
11.76	org.motechproject.server.config	454
11.77	org.motechproject.server.config.domain	456
11.78	org.motechproject.server.config.service	464
11.79	org.motechproject.server.startup	465
11.80	org.motechproject.server.ui	466
11.81	org.motechproject.server.ui.ex	467
11.82	org.motechproject.server.web.controller	467
11.83	org.motechproject.server.web.form	472
11.84	org.motechproject.server.web.helper	478
11.85	org.motechproject.server.web.validator	480
11.86	org.motechproject.tasks.annotations	484
11.87	org.motechproject.tasks.contract	485
11.88	org.motechproject.tasks.domain	492
11.89	org.motechproject.tasks.ex	536
11.90	org.motechproject.tasks.json	537
11.91	org.motechproject.tasks.repository	539
11.92	org.motechproject.tasks.service	539
11.93	org.motechproject.tasks.util	549

The topics below will give you an introduction to MOTECH, an open source mHealth platform developed by the Grameen Foundation. Some of these topics are quite technical in nature (i.e. aimed at software developers), while others are more accessible to people who aren't familiar with software code.

About MOTECH

Mobile Technology for Community Health (MOTECH) is a modular, extensible open source software project originally designed for mobile health (mHealth), which can also be used outside of the health domain. It allows organizations to use mobile technology to communicate information to patients, collect data about patients, alert caregivers to patients' status, and schedule caregivers' work. The modular system allows organizations to choose among multiple mHealth technologies and to enable data sharing for users of the system.

1.1 MOTECH Architecture Overview

MOTECH consists of the core platform and several modules, each providing use of a technology such as SMS or email, or access to an external system such as CommCare. Organizations can choose to install one or more modules, and developers can extend MOTECH by writing new modules. MOTECH is written in Java. It depends on open source systems including Apache Tomcat, Apache ActiveMQ, and Quartz. For more information about MOTECH architecture, see *Core Architecture* and *Modules Architecture*.

1.2 What can MOTECH do?

The MOTECH Platform can be used for setting appointments, tracking any scheduled activity, and managing workers deployed in the field. Its initial implementations have been for mHealth projects that improve health by sending messages to patients and caregivers based on an evaluation of the recommended schedule of care compared to the patient's health-related actions. Some features of typical MOTECH-based applications are:

Communicate information to patients via voice or SMS according to a schedule of care defined for the patient's condition, e.g.:

- Reminders for ANC appointments, lab visits, etc.
- Reminders to take medication on schedule, e.g., DOTS, ART, etc.
- Reminder notices to take children for scheduled immunization services

Collect data from patients or caregivers, e.g.:

- Patients report their symptoms prior to treatment or during treatment (adverse events)
- Patients give feedback on service delivery
- Caregivers report what service was delivered to a patient and on what date

Alert caregivers of the status of their patients, e.g.:

- Notify Community Health Worker when patient has not taken ART, DOTS or other drugs

- Notify nurse when patient has not kept a scheduled appointment (e.g., ANC visit)

Facilitate communication between patients, caregivers, and/or health administrators, e.g.:

- Establish secure peer networks for patients who share similar health concerns
- Initiate conversations between patients and caregivers in a way that allows the caregiver to manage the workload effectively

Creating Applications

This section is intended to help implementers of MOTECH get started creating custom solutions. The topics below cover some of the common features that need to be configured for a MOTECH app.

As the platform matures, most of the features below will be usable by implementers without developing any software code. As of now, a number of these features do require some coding in order to use (the topics below provide sample code where appropriate).

2.1 Introduction

Motech provides [Maven archetypes](#) in its [Nexus repository](#) which you can use to create a new Motech module. The archetypes supply basic source code needed for a new module, as well as configuration files for packaging the module as a bundle to be loaded into a Motech server.

The first archetype is the *minimal bundle archetype*. This supplies just enough source code and configuration to make a “Hello World” module.

Additional archetypes can add functionality to the minimal archetype:

- The *http bundle archetype* adds a servlet to respond to HTTP requests, and a simple web user interface.
- The *repository bundle archetype* adds a repository layer for storing and retrieving data from *MOTECH's data services*.
- The *settings bundle archetype* adds a properties file to store custom module configuration, and exposes the configuration through Motech's web user interface

Any combination of these additional archetypes may be added to the minimal archetype.

2.1.1 Minimal Bundle Archetype

To create a new minimal bundle from the minimal bundle archetype, use the following command:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

This will create a new Maven project in your current directory.

This is a long command. Here is an explanation of the parameters:

<i>parameter</i>	<i>value</i>	<i>explanation</i>
-DinteractiveMode	false	no need to wait for user input
-DarchetypeRepository	http://nexus.motechproject.org/content/repositories/releases	where to find the archetype
-DarchetypeGroupId	org.motechproject	group name for the archetype
-DarchetypeArtifactId	minimal-bundle-archetype	which archetype to use
-DarchetypeVersion	0.25-SNAPSHOT	Motech version to use with the new module
-DgroupId	org.motechproject	group name for the new module
-DartifactId	motech-test-module	artifact name for the new module
-Dpackage	archetype.test	Java package for new module classes
-Dversion	0.1-SNAPSHOT	version of the new module itself
-DbundleName	"Archetype Test Module"	name of the new module

2.1.2 HTTP Bundle Archetype

To create a new bundle that has HTTP support, use the following two commands from the same directory.

Create a minimal bundle with configuration modified for HTTP:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

Note the new parameter:

-Dhttp	true
--------	------

Add new source files from the HTTP archetype:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

Note the new archetype Id:

-DarchetypeArtifactId	http-bundle-archetype
-----------------------	-----------------------

2.1.3 Repository Bundle Archetype

To create a new bundle that has support for *MOTeCH's data services*, use the following two commands from the same directory.

Create a minimal bundle with configuration modified for repository:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

Add new source files from the repository archetype:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

2.1.4 Settings Bundle Archetype

To create a new bundle that has module settings support, use the following two commands from the same directory.

Create a minimal bundle with configuration modified for settings:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

Add new source files from the settings archetype:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

2.1.5 Combined Bundle Archetype

The minimal bundle archetype can be supplemented with any combination of additional archetypes. To create a bundle that uses them all, use all the following commands from the same directory.

Create a minimal bundle with configuration modified for all additional archetypes:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

Add source files from all the additional archetypes:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

2.1.6 Using Archetypes Locally

You can also use the archetypes locally, without the Motech Nexus repository. First, you must build the archetypes locally. You can either follow the `developer` guidelines to set up your developer environment, or to build locally without committing:

```
git clone https://code.google.com/p/motech/
cd motech
mvn clean install
```

Then you can use the archetypes from your Maven local catalog:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeCatalog=local -DarchetypeGroupId=org.motech
```

Note the new parameter:

-DarchetypeCatalog	local
--------------------	-------

2.2 Installing and Upgrading MOTECH

There will be more text here.

2.3 Configuration System

This document describes the MOTECH configuration system.

2.3.1 Step 1: Specify Config Locations

Default Behavior

By default, all configuration files are loaded from one of the following locations:

```
....${user.home}/.motech/config/
```

For example, if Motech runs under the user `motech` in Linux, configuration files will be searched in `“/home/motech/.motech/config”` folder.

Or:

```
..../etc/motech/
```

If the configuration files are not found in the above mentioned location, files will be searched in `“/etc/motech”` folder.

Overriding Default Behavior

If you want to override the location where configuration files are loaded from, you have to add `config.locations.properties` to `${user.home}/.motech` or to `${user.home}/` directory. Otherwise the property file in classpath will be picked up. Default `config.locations.properties` file:

```
....config.location= ${sys:user.home}/.motech/config/,/etc/motech/.
```

where `“${sys:user.home}”` is used to specify the home directory.

Note that, multiple locations can be specified for `config.location` property, and `motech-settings.properties` (which contains platform core config) config file will be searched starting from the first location and then falling back to next specified location, if it is not found. The directory in which it is found, is considered as the current config location and all other config files will be looked only in that particular location. For e.g., in the above sample config, if you have `motech-settings.properties` in `/etc/motech/`, then all config files will be searched in `/etc/motech/` location only. You cannot have files in different locations.

2.3.2 Step 2: Bootstrap Configuration

There are certain properties which are essential for the system to start up. These properties can be defined either in `bootstrap.properties` file or by environment variables. The properties are:

- `db.url` – The database connection url, e.g.: `localhost:5984`
- `db.username` – The user who has access to the database.
- `db.password` – Password to connect to the database.
- `tenant.id` – Optional. Default value is default.
- `config.source` – Optional. The source from which MOTECH core configuration and all module configurations should be read. Valid config values can be either one of `FILE` or `UI`. Default value is `UI`.

During startup, the system will look for these configurations in the following locations, falling back in order (that is when the properties are found in any of the following locations, search look up will stop):

1. **Config Directory Environment Variable:** If `MOTECH_CONFIG_DIR` environment variable is defined, then the system will load properties from `${MOTECH_CONFIG_DIR}/bootstrap.properties`.
2. **Config Environment Variables: Config is loaded from one or more environment variables:**
 - `MOTECH_DB_URL` – specifies value for `db.url`

- MOTECHE_DB_USERNAME - specifies value for db.username
- MOTECHE_DB_PASSWORD – specifies value for db.password
- MOTECHE_TENANT_ID – specifies value for tenant.id
- MOTECHE_CONFIG_SOURCE – specifies value for config.source

3. Location from Property file: bootstrap.properties file is loaded from any one of the locations specified in config-locations.properties file described above.

We are working on a feature in which, if bootstrap.properties is not found in any of the above mentioned locations, a UI will be presented to user after startup, prompting for bootstrap properties.

2.3.3 Step 3: MOTECHE Core Config

There are some system configurations and activemq configurations which are needed to get MOTECHE up and running.

- **System configurations:**

- system.language - Can take en(English), pl(Polski), es(Spanish), fr(French), it(Italian), sw(Swahili) as values(although only English and Polski are implemented as of now). Optional, default value is en.
- statusmsg.timeout - Represents the expiration time(in seconds) of messages and notifications in admin UI. Optional, default value is 60.
- login.mode - Can be repository or openId (case insensitive).
- provider.name - OpenId? provider name, mandatory in case login mode is openId.
- provider.url - OpenId? provider url, mandatory in case login mode is openId.

- **Activemq configurations:**

- jms.queue.for.events - Queue name to hold motech event messages. Optional, default value is Queue-ForEvents?.
- jms.broker.url - JMS broker URL. Can take failover URLs also. Sample values: tcp://localhost:61616, failover:(tcp://192.168.32.1:61616,tcp://192.168.32.2:61616)?randomize=false
- jms.maximumRedeliveries - Maximum number of redeliveries in case of any exceptions and a message consumption fails. Optional, default value is 0.
- jms.redeliveryDelayInMillis - Delay(in seconds) between successive re-deliveries of messages. If delay=d and first exception was raised at time=t, then successive redelivery times are calculated using exponential backoff . i.e. t+d, t+(d*2), t+(d*4), t+(d*8), t+(d*16) and so on, till maximum redelivery count is reached. Optional, default value is 2000.
- jms.concurrentConsumers - Optional, default value is 1.
- jms.maxConcurrentConsumers - Optional, default value is 10.
- jms.session.cache.size - Optional, default value is 10.
- jms.cache.producers - Optional, default value is false.

Case 1: When ConfigSource is FILE

Define motech-settings.properties file in any one of the locations defined in config-locations.properties with the above mentioned properties.

Case 2: When ConfigSource is UI

After server startup, if core settings are not configured already, you will be presented with a startup page which asks for System Language, Queue URL for events, Login Mode and user setup based on login mode. Other activemq settings can be changed in Settings tab after logging in.

2.3.4 Step 4: Module Configurations

Case 1: When ConfigSource is FILE

Module specific property files can be added to:

```
....<config-location-dir>/<module-symbolic-name>/ directory
```

and any JSON templates/configurations to:

```
....<config-location-dir>/<module-symbolic-name>/raw/ directory.
```

A typical example of a motech's module symbolic name:

```
....<module-name>
```

prefixed with “org.motechproject.motech-”.

All these files are monitored for changes. So, any change to these config files at runtime would be detected and saved in DB. Restart the module if required using Manage Modules tab in UI. We are enhancing the config monitor to raise an event in case of config change. This event can be listened by interested modules and take appropriate actions.

Case 2: When ConfigSource is UI

After server startup, you can find each module having settings UI associated with it in the Manage Modules tab, where you can edit the properties for the module. Also, restart the module if required. We are enhancing the config monitor to raise an event in case of config change.

2.4 Modeling Data with MOTECH Data Services

There will be more text here.

2.4.1 Javadoc

org.motechproject.mds.service

org.motechproject.mds.annotations

org.motechproject.mds.builder

org.motechproject.mds.config

org.motechproject.mds.domain

org.motechproject.mds.dto

org.motechproject.mds.enhancer

org.motechproject.mds.ex

org.motechproject.mds.filter
org.motechproject.mds.jdo
org.motechproject.mds.repository
org.motechproject.mds.util
/org/motechproject/mds/web/package-index

2.5 Configuring Messaging: SMS

There will be more text here.

2.6 Configuring Messaging: IVR

There will be more text here.

2.7 Connecting MOTECH to OpenMRS

There will be more text here.

2.8 Integrating MOTECH with CommCare

There will be more text here.

2.9 Using the Tasks Module

There will be more text here.

2.10 Configuring Pill Reminders

There will be more text here.

2.11 Configuring Your MOTECH App to be HIPAA Compliant

There will be more text here.

2.12 Tour of MOTECH UI

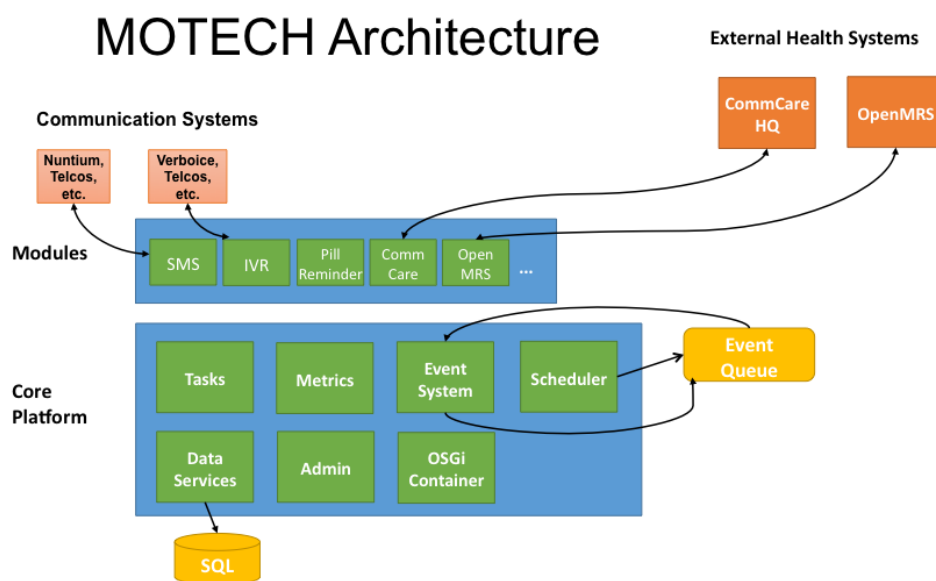
There will be more text here.

Architecture and Technical Overviews

3.1 Core Architecture

3.1.1 Architecture Overview

MOTECH can logically be broken into the core platform and modules. The core platform wraps several well-known open source systems, and augments and exposes their features to the other components. The main functions of the core are to wrap ActiveMQ (which provides the message queue) and present an internal pub/sub like event interface to the module and implementation layers. The core also provides a module loading environment (OSGi), an interface to the Scheduler, and access to the database.



Modules within MOTECH are self-contained bits of functionality that are loaded into the server via the OSGi host. Typically a module provides one type of functionality, such as SMS or access to an external health system. For more information, see [Modules Architecture](#). For a list of current and planned modules, see [Modules](#).

MOTECH is designed to be horizontally scalable with multiple MOTECHs all acting as workers connecting to the same message queue.

3.1.2 Design Philosophy

Stateless

A core design principle of the MOTECH platform is that the server should be stateless across requests to allow for horizontal scalability. It is expected that code running within the MOTECH server should perform a single action per request and then return. The module should never persist any state in memory or local disk and expect that state to be available to later requests.

Events

To aid in the development of stateless services, the MOTECH engine provides a pub/sub like event system. (The event system follows the publish-subscribe pattern but does not implement the standard Java pub/sub protocol.) It helps to decouple emitters of events from the modules that wish to consume them. Any module can emit an event by calling the `EventRelay` and passing it a `MotechEvent` and a subject. To register for an event, a module just needs to annotate a method with the list of event subjects of interest.

For more information, see [Event and Scheduler Architecture](#).

Scheduled Events & Timers

To assist in the development of a stateless event-based server, the MOTECH platform provides access to a flexible scheduling system. Using the open source Quartz engine, MOTECH can easily schedule events for future consumption. For more information, see [Event and Scheduler Architecture](#).

3.1.3 Subsystems

Tasks System

The Tasks system allows you to connect modules without code by using tasks. Each task consists of three parts:

1. Trigger: an event raised by Module A (or the Scheduler)
2. Filter: a conditional statement specifying whether the task should run
3. Action: an action executed by Module B in response

In between the trigger and the action, tasks may use data loaders to look up data from other modules that are registered as data sources.

Data Services

MOTECH Data Services is a flexible data modeling system that allows users to define and share custom schemas without code, and provides auditing and revision tracking. It is a JDBC-based user configurable database abstraction layer on top of a standard SQL database. It provides generated POJOs and OSGi service interfaces for the data objects, generated CRUD events, and generated user interface for data browsing and editing. In a future release it will also support auto-generation of REST APIs.

3.1.4 Dependencies on Third-Party Systems

Quartz Scheduler

Quartz is an open source job scheduling engine that enables MOTECHE modules to schedule events for future consumption.

Tomcat

Apache Tomcat provides the application container for MOTECHE.

ActiveMQ

Apache ActiveMQ is an open source message broker that provides the message queue.

OSGi

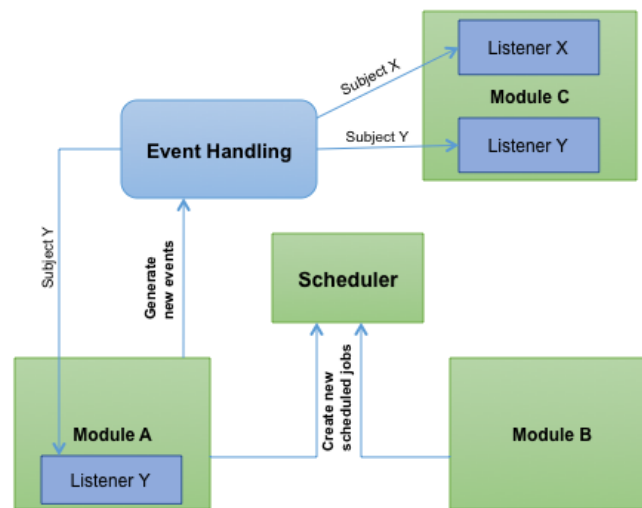
Each MOTECHE module is an OSGi bundle. Using OSGi allows the platform to manage the bundle lifecycle (adding, removing, starting, and stopping modules), and allows modules to expose service interfaces. For more information, see *Modules Architecture*.

3.2 Event and Scheduler Architecture

3.2.1 Event Handling and Scheduling among Modules

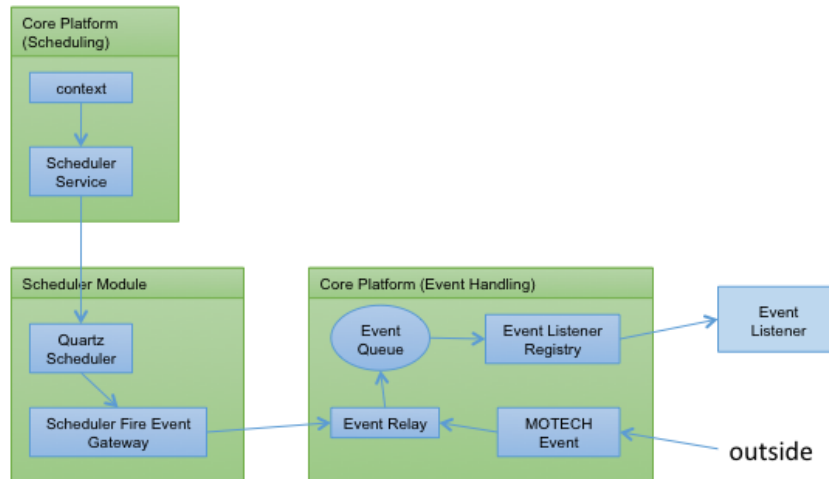
The following diagram provides three examples of Motech modules:

- Motech module A publishes events, schedules new jobs and listens for events. An example of a Motech platform that has these three responsibilities is the Message Campaign module.
- Motech module B schedules new jobs.
- Motech module C listens to events of subject X and Y: listener X listens to events of subject X, and listener Y listens to events of subject Y.



3.2.2 Event Handling and Scheduling Architecture

To schedule a job in MOTECH, the core platform exposes the `MotechSchedulerService`. Clients of the service have an instance of it injected into the class that uses it. This service employs Quartz to schedule `MotechScheduledJobs`. When triggered, `MotechScheduledJobs` raise events that are sent through an event relay to the event queue. These messages are dequeued and then received by a consumer, the event listener registry, which in turn discovers all of the listeners on the event and invokes the appropriate method that was listening on that event.



Notes

- The scheduler service retrieves the Quartz Scheduler from the SchedulerFactoryBean. The service can only schedule MotechScheduledJobs, which all must include a MotechEvent.
- When the trigger for the scheduled job is satisfied, the job is executed and the accompanying Motech event is sent to the SchedulerFireEventGateway interface. This gateway is defined in schedulerFiredEventChannelAdapter.xml and a proxy is generated by Spring at runtime.
- The SchedulerFireEventGateway serves as an outbound message gateway to shuttle messages to the event relay, which then sends JMS messages to the event queue.
- The event queue dequeues messages and routes them to the event listener registry. The core platform has one consumer for events, a channel that routes messages to an event listener registry.
- The event listener registry retrieves the listeners that are listening on the motech event it is relaying (listening is based on the value bound to the motech event's subject key).
- The core platform scans new bundles for any method annotated as a MotechListener, and registers those methods with the listener registry.
- If the event has a specific destination or only one listener, the listener's handle method is called. This will invoke the method that was annotated as a MotechListener. These listener classes can be project specific and will reside outside of the core platform.
- If the event has multiple listeners, a separate event is raised for each listener. The original event object is not handled by a listener in this case.
- The EventListener's handle method will invoke the method that was annotated as a MotechListener. The MotechEvent will be passed to that method as a parameter.

3.3 Modules Architecture

Modules within MOTECH are self-contained bundles of functionality that are loaded into the server via the OSGi host. Modules interact with the core platform through its APIs and with other modules, either through their APIs or by consuming their events. Modules can expose service interfaces of their own as well as emit their own events. Modules may also register servlet controllers, which allow them to respond to HTTP requests.

Through MOTECH Data Services, a modules may expose entities from its data model. This allows a module to provide a data editor, REST APIs, record-level security, and field-level auditing. Via the Tasks system, modules can expose triggers, data and actions to be orchestrated by other modules.

See [Modules](#) for the list of current and planned modules.

Reasons to create a module include developing application-specific UI, business logic, and data models. Another reason is to develop generic reusable functionality to share with the MOTECH community.

3.4 Data Services Architecture

There will be text here.

3.5 Security Model

There will be text here.

4.1 Alerts

Collects alerts for users in an inbox-like container

4.2 Appointments

Provides appointment scheduling and reminders

4.3 Batch

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

4.4 CMS Lite

Provides basic content storage and retrieval

4.5 CommCare

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

4.6 Data Services Module

Integrates data from external data sources and provides sharable data schemas

4.7 Email

Sends and logs email messages

4.8 *Event Logging*

Allows MOTECH modules to easily see each others' events

4.9 *Hindi Transliteration*

Supports transliteration of English strings to Hindi using ITRANS encoding

4.10 *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

4.11 *Message Campaign*

Enrolls users in message campaigns with flexible content-scheduling rules

4.12 *mTraining*

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

4.13 *OpenMRS*

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

4.14 *Pill Reminder*

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

4.15 *Schedule Tracking*

Enrolls users for alerts based on complex scheduling rules

4.16 *Scheduler*

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

4.17 *IVR*

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

4.18 *SMS*

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

4.19 *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

4.19.1 Data Services Module

There will be more text here.

4.19.2 Email Module

There will be more text here.

4.19.3 Scheduler Module

There will be more text here.

4.19.4 Tasks Module

There will be more text here.

Developing the MOTECH Platform

This section of the documentation is aimed at developers and maintainers of the MOTECH Platform. These docs help MOTECH devs get up and running, from configuring a machine, to getting the code, and committing and reviewing changes. If you find any errors in the topics below, or you have any other questions about MOTECH development, please contact us via the [mailing list](#).

5.1 Setting up a Development Environment

The topics below will walk you through installing MOTECH on a Mac or Linux machine. Note that the Docker-based setup is much faster than the “official” method but the instructions are in beta. If you have any trouble with either approach, please feel free to contact us via motech-dev@googlegroups.com.

5.1.1 Installing MOTECH for Developers (“Official” Method)

Table of Contents

- Installing MOTECH for Developers (“Official” Method)
 - Installing on Ubuntu
 - Installing on a Macintosh
 - Building and Installing MOTECH
 - Installing the IDE, IntelliJ IDEA Community Edition & open MOTECH project

Installing on Ubuntu

The versions below may change, most likely the latest stable release will work for your purposes. If they do not, please feel free to send in feedback.

1. **Install Ubuntu Desktop 12.04.2 LTS 64bit** [Installation instructions](#)

Note: 64-bit is required for Motech’s installation

2. Install Maven, Git, Curl, ActiveMQ, and mysql

(a) In terminal, type

```
sudo apt-get install curl git maven activemq mysql-server
```

- (b) On a fresh Ubuntu installation, you may need to run the following first

```
sudo apt-get update
```

3. Configure ActiveMQ

Run the following

```
sudo ln -s /etc/activemq/instances-available/main /etc/activemq/instances-enabled/main
```

Note: For ActiveMQ scheduled delivery to work, you must set the attribute: **schedulerSupport="true"** for the broker element in your activemq.xml config file. This file should be located at (active-mq-folder)/conf/activemq.xml. See ActiveMQ docs.

4. Install JDK 7

- (a) Go to [The Java JDK Download Page](#)
- (b) Accept License Agreement
- (c) Click on `jdk-7u51-linux-x64.tar.gz` (or latest stable version)
- (d) Extract the file into your home directory, ie: `/home/*<user>*/jdk1.7.0_51`
- (e) Set the proper Java environment and change maven options:

i. Start a new terminal session

ii. Edit your `.profile` file

```
nano ~/.profile
```

iii. append the following at the end of the file:

```
export PATH="$HOME/jdk1.7.0_21/bin:$PATH"
export JAVA_HOME=$HOME/jdk1.7.0_21
export MAVEN_OPTS="-Xmx512m -XX:MaxPermSize=128m"
export CATALINA_OPTS="-Xms1024m -Xmx2048m -XX:MaxPermSize=1024m"
```

iv. Save the changes (Ctrl+X) and quit

v. Confirm the settings are right

vi. Log out & log back in & start a new terminal

vii. Type

```
java -version && env | grep "\ (MAVEN_OPTS\|CATALINA_OPTS\)"
```

You should see something like:

```
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b11)
Java HotSpot(TM) 64-Bit Server VM (build 23.21-b01, mixed mode)
MAVEN_OPTS=-Xmx512m -XX:MaxPermSize=128m
CATALINA_OPTS=-Xms1024m -Xmx2048m -XX:MaxPermSize=1024m
```

5. Install Tomcat7

- (a) Go to [Tomcat's download page](#)
- (b) Under 7.0.52 (or the latest stable version) - Binary Distributions - Core, click on tar.gz
- (c) Once downloaded, expand the file to your home directory, i.e.:
`/home/*<user>*/apache-tomcat-7.0.52`
- (d) Edit the `tomcat-users.xml` file (located under `\etc\tomcat7\conf\`) to add an admin user:
- (e) In the terminal type

```
nano ~/apache-tomcat-7.0.52/conf/tomcat-users.xml
```

- (f) Insert a line similar to the following before the closing `</tomcat-users>` tag:

```
<user username="*<username>*" password="*<password>*" roles="manager-gui"/>
```

- (g) Save the changes (Ctrl+X) then quit
- (h) Now edit `~/ .bashrc` to setup tomcat's environment variable

```
nano ~/ .bashrc
```

- (i) Append the following line:

```
export CATALINA_HOME=$HOME/apache-tomcat-7.0.52
```

- (j) Save the changes (Ctrl+X) then quit
- (k) Start a new terminal session or type

```
source ~/ .bashrc
```

6. Install CouchDB

- (a) Download the latest stable sources from <http://couchdb.apache.org> and then follow the instructions in the INSTALL.Unix file.

For Ubuntu 13.10, just run:

```
sudo apt-get install couchdb
```

- (b) Once this is done, navigate to <http://localhost:5984> to verify that the installation completed successfully.

```
{ "couchdb": "Welcome", "uuid": "52068def93b82a2653dcf352a4f9273a", "version": "1.4.0", "vendor": {
```

7. Install CouchDB-Lucene

- (a) Follow [these instructions](#) also be sure to follow the [proxy handler instructions](#)
- (b) Once the proxy has been configured, restart couchdb with:

```
sudo service couchdb restart
```

- (c) After restarting couchdb, navigate to http://localhost:5984/_fti and you should see something like this:

```
{ "couchdb-lucene": "Welcome", "version": "0.10.0-SNAPSHOT" }
```

8. Setup MySQL

- (a) In your motech source root directory, type in the terminal:

```
$ mysql -u root -p
```

(b) then type:

```
sql> create database motechquartz;
sql> create database motech_data_services;
sql> create user 'quartz'@'localhost' identified by 'quartz2123';
sql> grant all privileges on motechquartz.* to 'quartz'@'localhost';
sql> exit;
```

(c) then type:

```
mysql -u root -p motechquartz < <motech-dir>/modules/scheduler/sql/create_db_schema_quar
```

9. Start Tomcat

(a) In terminal, type:

```
~/apache-tomcat-7.0.52/bin/catalina.sh jpda start
```

(b) You should see messages similar to:

```
Using CATALINA_BASE:   /home/*<user>*/apache-tomcat-7.0.52
Using CATALINA_HOME:   /home/*<user>*/apache-tomcat-7.0.52
Using CATALINA_TMPDIR: /home/*<user>*/apache-tomcat-7.0.52/temp
Using JRE_HOME:        /home/*<user>*/jdk1.7.0_51
Using CLASSPATH:       /home/*<user>*/apache-tomcat-7.0.52/bin/bootstrap.jar:/home/*<use
```

(c) You can also confirm tomcat was started by going to <http://localhost:8080> in a browser

10. Jump to the [Building and Installing MOTECH](#) section to install MOTECH

Installing on a Macintosh

1. Installing Prerequisites for MOTECH

(a) Installing [HomeBrew](#)

To install Homebrew, run the following in the terminal

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/homebrew/go/install)"
```

(b) Use Homebrew to install git, erlang, ActiveMQ, and Apache Tomcat:

```
brew install git
brew install activemq
brew install tomcat
brew install maven
```

(c) Homebrew installations are located in `/usr/local/Cellar`` with symlinks in ```/usr/local/bin`, which should already be part of your `$PATH` environment variable.

Note: Homebrew provides instructions about how to run these applications, as well as how to have launchd start them automatically on system startup.

(d) Configuring Tomcat:

Edit the `tomcat-users.xml` file to add an admin user. Insert a line similar to the following before the closing `</tomcat-users>` tag:

```
<user username="motech" password="motech" roles="manager-gui"/>
```

(e) Installing JDK 7:

Mac OS includes JDK6 by default, however JDK 7 is required for MOTECH. Use [these instructions](#) to install the latest version of the JDK.

(f) Installing MySQL:

- i. Before installing MySQL, you will need Xcode from the App Store. This can take a while; it's a big download.
- ii. Next start Xcode from the Launchpad (rocketship icon in the dock) and select Install. Then you can quit Xcode; you don't need to keep it running.

Note: (Command Line Tools using Xcode are included in OS X Mavericks, but not previous OS versions. If you are running Mountain Lion, you can follow [these instructions](#).)

- iii. Go to <http://dev.mysql.com/downloads/mysql/> and download the appropriate DMG archive. Open it, double-click on the installer, and follow directions.
- d. Once mysql has finished installing, double-click the MySQL preferences pane in the DMG and follow instructions. For more details see [these instructions](#).

Note: Homebrew can be used to install MySQL, however Homebrew will not install the Mysql System Preferences control panel.

2. Setting up Symbolic Link and Environment Variables

- (a) Create a symbolic link from the Tomcat directory (Homebrew installs into `/usr/local/Cellar/tomcat/<version number>/libexec`) to `/usr/local/tomcat`:

```
ln -s /usr/local/Cellar/tomcat/`brew info tomcat | grep stable | awk '{print $3}'` | sed
```

- (b) Edit your `~/ .bash_profile` to set environment variables (catalina is Tomcat):

```
export JAVA_HOME="/Library/Java/Home"
export MAVEN_OPTS="-Xmx512m -XX:MaxPermSize=128m"
export CATALINA_HOME="/usr/local/tomcat"
export CATALINA_OPTS="-Xms1024m -Xmx2048m -XX:MaxPermSize=1024m"
export PATH="/usr/local/mysql/bin:$PATH"
```

- (c) **When you're done editing:**

```
source ~/.bash_profile
```

3. Jump to the [Building and Installing MOTECH](#) section to install MOTECH

Building and Installing MOTECH

1. Getting the MOTECH code

List of MOTECH repositories

Generic developer git workflow

2. Building MOTECH

- (a) Assuming you issued the git clone command in your home directory root, in the terminal

```
$ cd ~/motech
$ mvn install
```

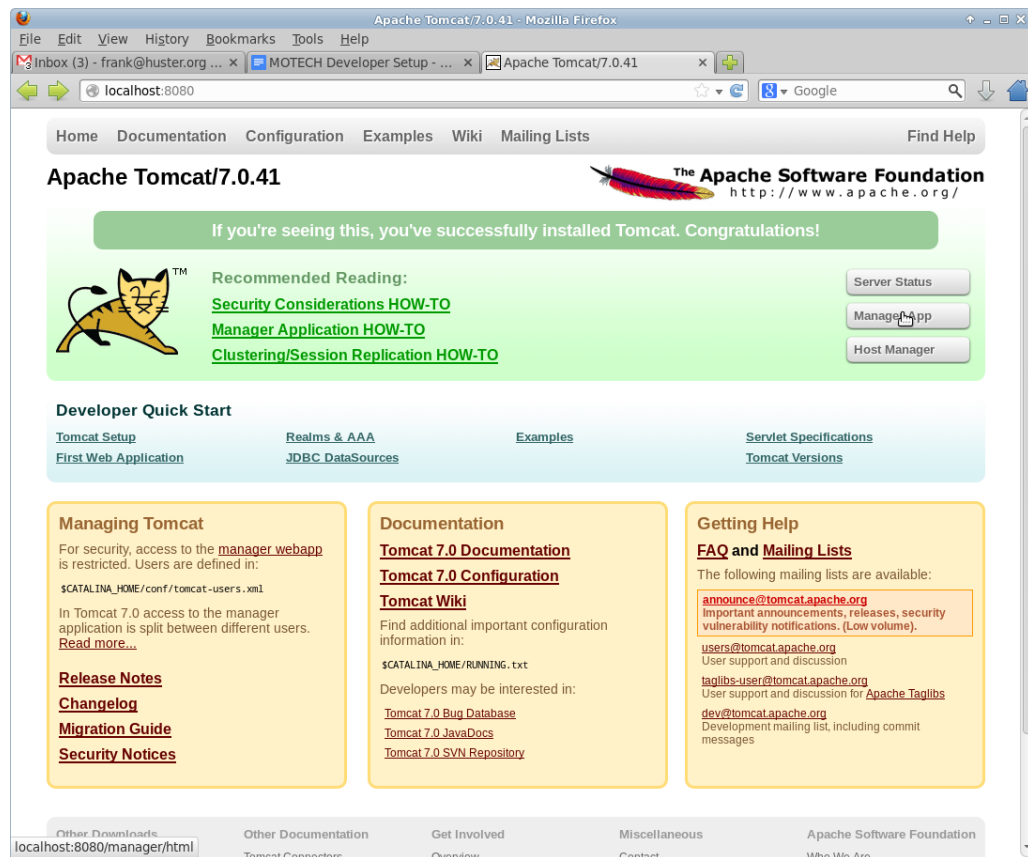
- b.) It takes some time to build MOTECH, but eventually you should see:

```
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 29:19.284s
[INFO] Finished at: Fri Jun 07 12:12:43 PDT 2013
[INFO] Final Memory: 152M/378M
[INFO] -----
```

Note: Should you get a `java.lang.OutOfMemoryError` exception, it may be because you forgot to set `MAVEN_OPTS` as described in [3.5]. But you may need to increase `-Xmx`. So something like `-Xmx1024m` might work.

3. Install MOTECH

- (a) In a browser, go to <http://localhost:8080>



- (b) Click on Manager App

- (c) Type the user/password you used in tomcat-users.xml

temporary hack you need to remove `~/motech/config/motech-settings.conf` to allow the create initial user wizard.

- (d) In the Tomcat Web Application Manager, scroll down to the Deploy section and the WAR file to deploy subsection, click on Browse and select or navigate to `~/motech/platform/server/target/motech-platform-server.war` then click on Deploy

The screenshot shows the Tomcat Web Application Manager interface. The page has a header with the Apache Software Foundation logo and the title "Tomcat Web Application Manager". Below the header is a message box. The main content area is divided into sections: "Manager" with links for "List Applications", "HTML Manager Help", "Manager Help", and "Server Status"; "Applications" with a table of running applications; and "Deploy" with fields for "Context Path (required)", "XML Configuration file URL", and "WAR or Directory URL". The table of applications includes entries for "/", "/docs", "/examples", "/host-manager", "/manager", and "/motech-platform-server". The "/motech-platform-server" entry is highlighted, and its "WAR or Directory URL" field is populated with "localhost:8080/motech-platform-server/".

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/motech-platform-server	None specified		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

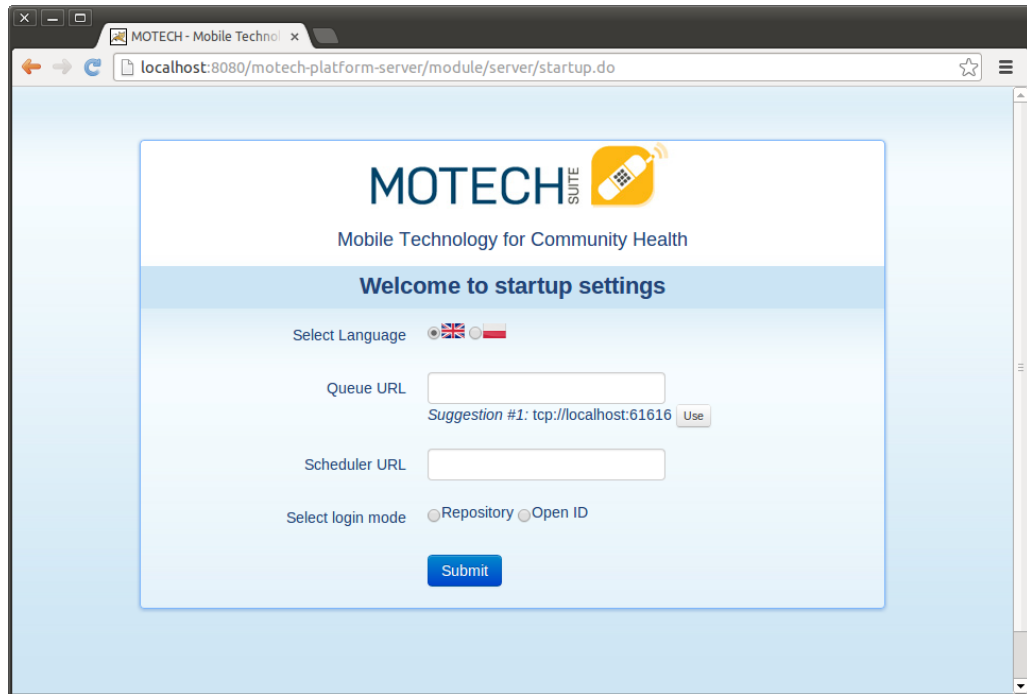
Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

- (e) Depending on your machine it could take a while for motech-platform-server to deploy

- (f) In the Tomcat Web Application Manager page, click on `/motech-platform-server`, you get the MOTECH initial user screen

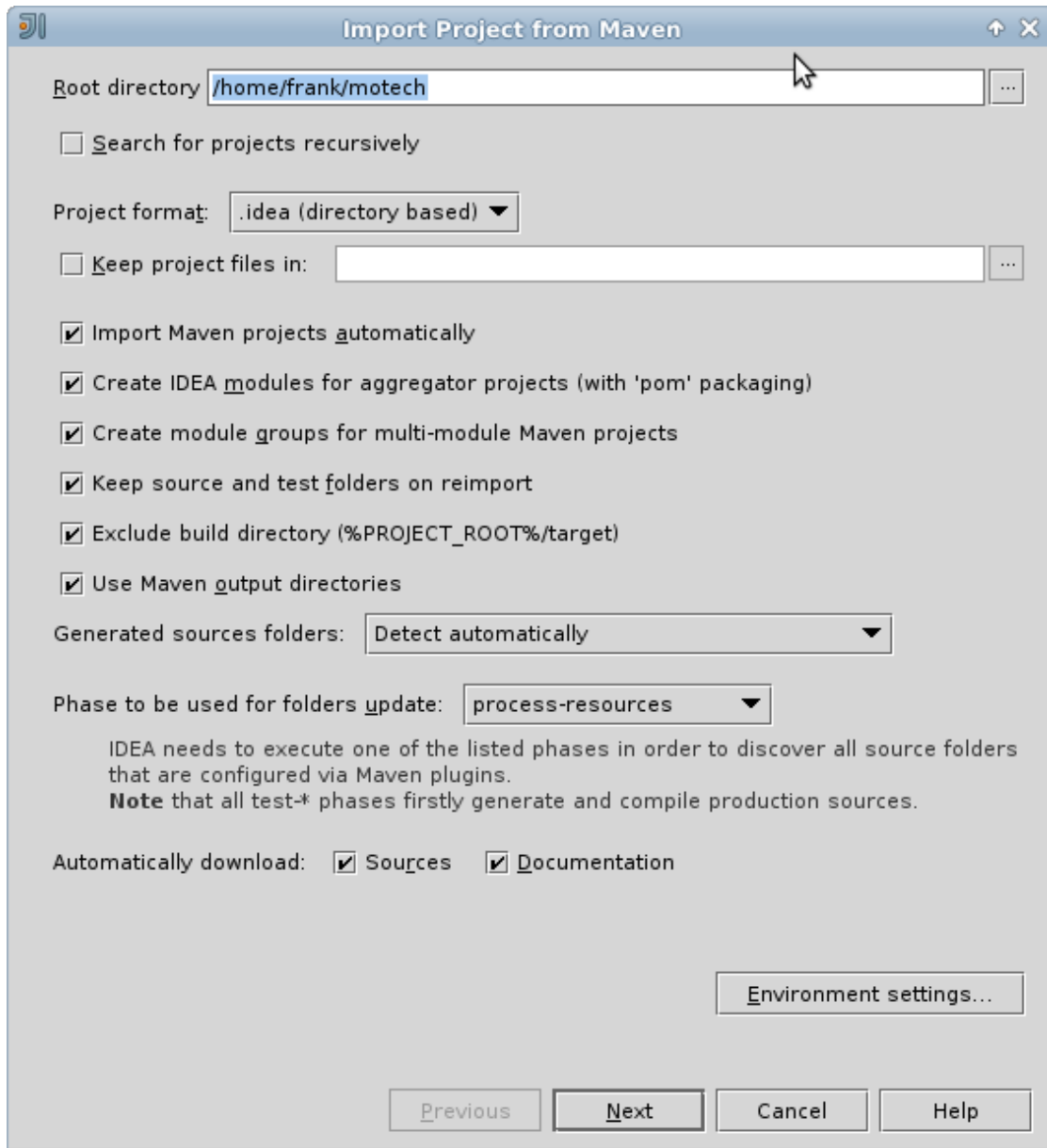


Installing the IDE, IntelliJ IDEA Community Edition & open MOTECH project

1. Go to the [Jetbrains home page](#) and click on Download Now in the Community Edition box, then expand the file to your home directory.
2. From a terminal, assuming you extracted IntelliJ to ~/idea-IC-129.713, start IntelliJ

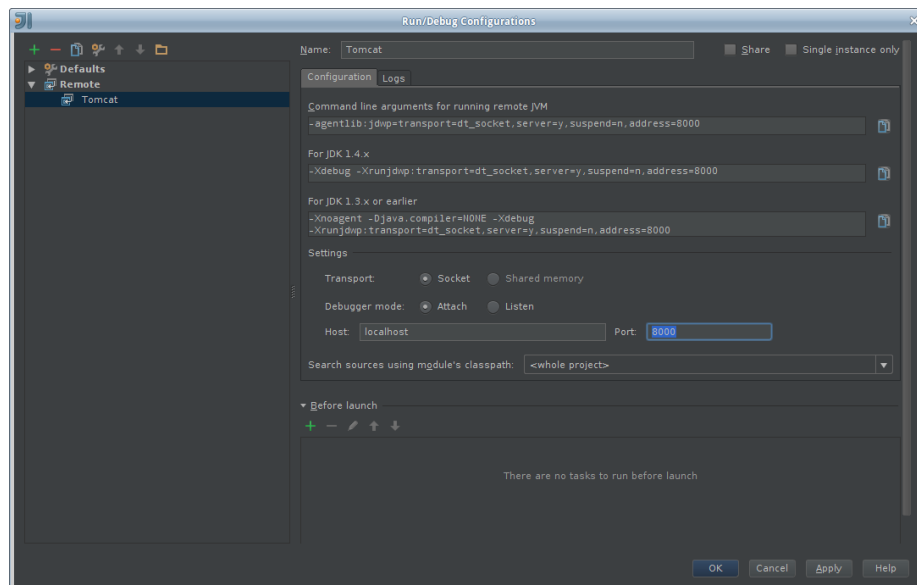
```
$ ~/idea-IC-129.713/bin/idea.sh
```

3. Select Import Project

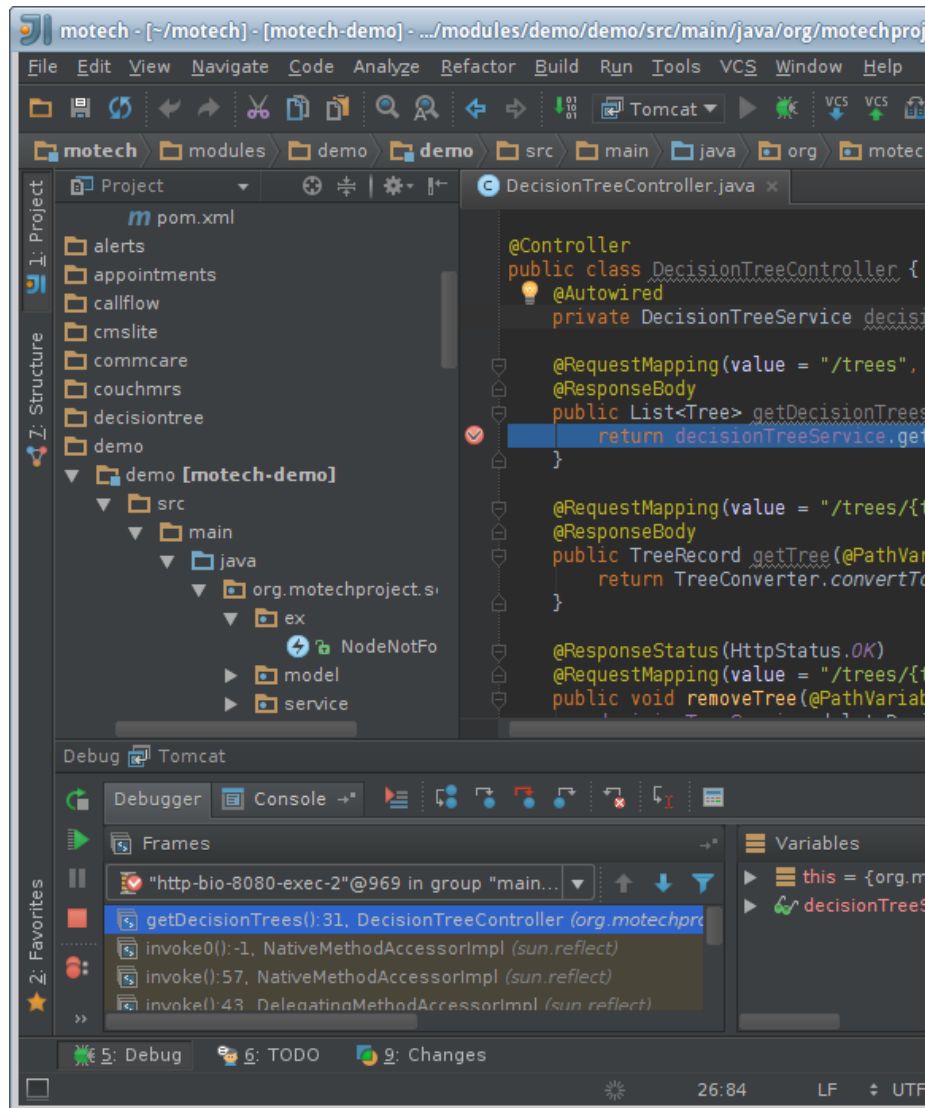


1. Select ~/motech/pom.xml, a dialog box will appear. Set the options as shown:
2. Click Next
3. In Select Profiles, do not select any profile, click Next
4. In Select Maven projects to Import, there should only be one project: org.motechproject:motech:0.20-SNAPSHOT, click Next
5. In Please select project SDK, if the 1.7.0_21 is present, select it, otherwise add it:
6. Click +
7. Select JDK
8. Select /home/frank/jdk1.7.0_21, then click OK
9. Click Next

10. Click Finish
11. Background processes will take a long time
12. You can also create a menu launcher, so you can start IntelliJ from the gui:
 - (a) From the Tools menu select Create Desktop Entry
 - (b) A IntelliJ menu item will be created in the Development application group
 - (c) Debug demo module in IntelliJ
 - (d) Start IntelliJ (from the command line, or from launcher icon if you created one)
 - (e) It'll automatically open the motech project (if it was the last project you worked on)
 - (f) From the Run menu select Edit Configurations
 - (g) Click on the green +
 - (h) Select Remote
 - (i) Give a name to your Run/Debug configuration and change the port to 8000 as:



- (j) Hit OK
- (k) Set a breakpoint somewhere in the demo module code, i.e.:
 - (l) From the Run menu, select Debug 'Tomcat' where Tomcat is the name of your configuration.
- (m) In the browser go to the place that will hit the breakpoint, i.e.: if you setup a breakpoint as in the previous screen, then in the Demo module, click the Decision Trees tab, and you should hit the breakpoint!



5.1.2 Installing MOTECH Using Docker (“Beta”)

Note: These instructions assume you’re running on Ubuntu. This setup is also possible on Mac OSX but the steps are slightly different. We hope to provide OSX instructions in future.

This document provides instructions for creating a MOTECH environment using [Docker](#) containers. These instructions are “in beta” (the *official* installation guide is still the one [here](#)), but many members of the MOTECH dev team have been following this approach with success. This installation method is much faster than the official route.

There are two supported ways to install MOTECH with Docker:

1. As an implementer - follow this approach if you want to install a released version of MOTECH.
2. As a developer - follow this approach if you will be developing MOTECH and want to build the platform and modules from source code.

Get Docker, Fig and motech-docker

Whether you're installing as an implementer or a developer, you'll need Docker and Fig:

Docker

1. Follow the instructions on the [Docker website](#) to get the latest version of Docker.
2. Execute the following to configure Docker to work for non-root users:

```
sudo groupadd docker
sudo gpasswd -a ${USER} docker (logout and re-login)
sudo service docker restart
```

Fig

Execute the following to install Fig:

```
sudo apt-get install python-pip python-dev build-essential
sudo pip install -U fig
```

motech-docker

Clone the [motech-docker](#) project from GitHub or download it as a zip file and extract it. You'll need to run all Fig commands from the motech-docker directory.

Implementer Setup

Go to your motech-docker directory. To setup as an implementer (everything is automagically installed):

```
./setup_as_imp.sh
```

Type the following to start MOTECH in the background:

```
fig up -d
```

Voila! MOTECH has started. Wait a little bit (about 30s) then direct your browser to: <http://localhost:8080/motech-platform-server>

Note: 'fig up' ERASES ALL YOUR DATA (well not really all, but pretend it does). You have to run it at least once to setup MOTECH. If you run it again, it'll erase everything you did in MOTECH. It's useful to start afresh, but remember: it nukes everything!

Developer Setup

Go to your motech-docker directory. To setup as a dev:

```
./setup_as_dev.sh
```

Type the following to start all the pieces that MOTECH needs to run in the background:


```
fig up -d
```

Once you start the containers with the `fig up -d` command above and *before* you build MOTECH for the first time, you need to copy the MOTECH binaries into the container's `/root/.motech/bundles` directory.

Conveniently, the container's `/root/.motech/bundles` directory is exposed as the `docker-motech-bundles` directory (with `a-rw` access) in your home directory (also note that the container's `/root/.motech/config` dir is also exposed as `~/docker-motech-config`). So, you can either manually copy the necessary binaries, or you can create a symbolic link to `~/docker-motech-bundles` from `~/motech/bundles`.

Assuming the latter, and that you never built MOTECH before, you'd run the following commands:

```
# go to your home dir
cd
# create the .motech dir
mkdir .motech
# create the symlink
ln -s ~/docker-motech-bundles .motech/bundles
```

If you built MOTECH before, you can just delete the `bundles` directory and create the symlink using the command above.

Build, deploy and run MOTECH: see `:doc:dev_install`.

Some Useful Fig Commands

Stop MOTECH

```
fig stop
```

Restart MOTECH

```
fig start
```

Watching logs

To watch all the logs (very verbose):

```
fig logs
```

To watch only the tomcat logs:

```
fig logs tomcat
```

See the sections in the generated `fig.yml` to see what other logs you can watch.

5.1.3 Configuring MOTECH

There will be more text here. This doc may belong in “Setting up a Development Environment” or in “Getting Started”.

5.2 Developing and Submitting a Patch

We use a web-based code review system called [Gerrit](#). Using this system, anyone can comment on a proposed change before it is merged to our Git repository. It's pretty easy to use; the instructions below will get you started.

5.2.1 Create a Gerrit Account

1. Navigate to <http://review.motechproject.org/>
2. Click sign in on top-right
3. Select Open ID provider (only Google accounts are enabled)
4. Select user name
5. Upload your SSH public key

5.2.2 Configuring Your Git Client to Use Gerrit

Follow these steps once for each *MOTECH code repository* that you clone.

1. Get source code

```
git clone ssh://<userid>@review.motechproject.org:29418/motech
```

2. Set up review branch

```
cd motech
git config remote.origin.push refs/heads/*:refs/for/*
```

3. Install change-id generation hook

```
scp -p -P 29418 <userid>@review.motechproject.org:hooks/commit-msg .git/hooks/
```

5.2.3 Development Workflow

1. Checkout to feature branch

```
git checkout -b newfeature
```

2. Make changes/test/multiple commits

3. When ready to submit changes: update master, squash commits and merge feature branch

```
git checkout master && git pull --rebase
git merge --squash newfeature
git gui
```

4. Edit commit message using the proper *commit message format*

5. Push changes

```
git push origin
```

5.2.4 Submitting Changes (Patch Set) to Incorporate Review Comments

If you've received some code review feedback and you'd like to make some changes, follow the steps below to add your changes as a new "patch set" to the existing Gerrit code review.

1. **Checkout patch from gerrit change:**
 - (a) Navigate to <http://review.motechproject.org/#/c/<change id>>
 - (b) Copy pull url under patch set section and run
2. Make changes
3. Copy change ID from Gerrit (top section in Gerrit change page)
4. Amend change ID in commit message
5. [Squash commits](#)
6. Push changes

5.2.5 Pushing to Remote Branches (Not for Review)

This practice enables developers to share in-progress feature work with others without actually submitting the changes for review.

1. Use branch namespace dev

```
git checkout -b dev/newfeature
git add . && git commit -m "message"
git push -u origin dev/newfeature:dev/newfeature
```
2. Once done with feature, squash commits and merge with master. Submit for review as mentioned above.

5.2.6 Additional Information

- <http://review.motechproject.org/Documentation/user-upload.html>
- <http://review.motechproject.org/Documentation/user-changeid.html>

5.3 MOTeCH Code Repositories

Each repository can be cloned from either GitHub/Google Code or Gerrit. If you're only interested in a copy of the code and will not be contributing, use the GitHub repo (no sign-in required). Otherwise use the Gerrit repo (sign-in required) where each commit will trigger a Jenkins build and be submitted for code review. Jenkins is our continuous integration (CI) system. Once your change is approved and merged by an authorized Gerrit reviewer, it will show up on the GitHub/Google Code repo.

5.3.1 MOTeCH Platform

The platform repo contains the motech-platform-server Tomcat servlet. In addition it also contains the essential Admin, Config, Tasks, Motech Data Services, Email, and Scheduler modules.

- Google Code Repository

```
git clone https://code.google.com/p/motech
```

- Gerrit Repository

```
git clone ssh://<userid>@review.motechproject.org:29418/motech
```

5.3.2 MOTECH Modules

This repo is the home of all optional MOTECH modules.

- GitHub Repository

```
git clone https://github.com/motech/modules
```

- Gerrit Repository

```
git clone ssh://<userid>@review.motechproject.org:29418/modules
```

5.4 Commit Message Format

To ensure that our commit messages are both concise and informative, all MOTECH committers are asked to follow the git commit message format outlined below. For reference, Linus Torvalds provides a description of a good commit message [here](#).

5.4.1 Format

1. First line is the Jira issue ID + subject (max 50 chars)
2. Leave a blank line (if not, the entire message is interpreted as subject)
3. Summary (ensure it is wrapped to a reasonable number of characters, e.g. 72, because git log does not handle wrapping).

The description of the change should be both brief and informative. The Linux kernel documentation has this to say about good commit summaries:

...the “summary” must be no more than 70-75 characters, and it must describe both what the patch changes, as well as why the patch might be necessary. It is challenging to be both succinct and descriptive, but that is what a well-written summary should do.

5.4.2 Author/Committer

There is no need to include the names of the developers who developed the change in the subject line. If there is more than one person working on a change, the committer is asked to include the `–author` parameter in his/her git commit to specify the second person’s name.

5.4.3 Example

Here is an example of a MOTECH commit message:

MOTECH-678 Moves campaign modules to github

Disables pillreminder, message-campaign and scheduletracking modules from main pom.xml as they've moved to a new repo on github - <https://github.com/motech/platform-campaigns>

Change-Id: I5964249887160c868fa9598c413aebb93a49fa32

5.5 Coding Conventions

Text to come

5.6 Code Review Workflow and Checklist

Text to come

5.7 Authoring Documentation

This document provides information specific to setting up and authoring Sphinx documentation for MOTECH. The [Sphinx Documentation Guide](#) is also a good resource.

Each MOTECH repository contains a *docs* directory populated with reStructured Text (reST) files. These files are built by Sphinx and hosted at <http://readthedocs.org>. A Sphinx plugin, called Javaspinx, builds Javadoc for the MOTECH codebase.

5.7.1 Installing Sphinx and Javaspinx

If you are working on MOTECH documentation, it is helpful to build the docs locally and review changes in advance of checkin. The steps below will get your environment configured to build Sphinx docs.

Install python utils

```
sudo apt-get install python-setuptools
sudo apt-get install python-dev
sudo apt-get install python-pip
```

Install sphinx

```
sudo easy_install sphinx
```

Install javaspinx and motechjavaspinx

```
sudo apt-get install libxslt-dev
sudo apt-get install libxml2-dev
sudo apt-get install zlib1g-dev
cd docs
sudo pip install -r requirements.txt
```

5.7.2 Building Docs

ReadTheDocs.org automatically builds Sphinx projects, but it is a good idea to build your documents and make sure they render correctly before you push your changes up to the code repository.

To build Sphinx docs, go to the docs directory and type:

```
make html
```

You can then use a web browser to view docs/build/html/index.html and make sure everything looks good.

5.7.3 Authoring and Submitting New Documents

To create a new documentation topic, simply create a file with extension *.rst* under the docs directory. Create a table-of-contents entry for your new doc in the appropriate index.rst file. The [reStructuredText Primer](#) is good to have handy as you write your doc.

When your document is ready for review, follow the instructions for [creating and submitting a patch](#) to submit it for code review.

5.8 Project Management

There will be text here.

5.9 Release Strategy and Procedure

Text to come

5.10 Browser Compatibility

Text to come

6.1 Demo: Hello World

Table of Contents

- Demo: Hello World
 - Overview
 - Generating the Module from Archetypes
 - * Generating the Minimal Bundle
 - * Tour of the Minimal Bundle
 - * Adding the HTTP Archetype
 - * Adding the Repository Archetype
 - * Building and Deploying the Module
 - An Event-Driven Hello World
 - * Declaring the Event System Dependency
 - * Event Subjects and Event Parameters
 - * Sending an Event
 - * Listening for Events
 - * Task Triggers and Task Actions
 - * Creating a Task in the User Interface
 - * Final Walkthrough

6.1.1 Overview

This “Hello World” tutorial aims to get you started with Motech development. The tutorial is separated into two phases. In the first, we generate the Hello World module incrementally from a series of archetypes provided by the Motech platform. We briefly tour the project code, configuration, and layout that make up a minimal Motech module, and then add in additional archetypes to support web requests and data services. Finally, we build and deploy the module to our Motech server.

In the second phase, we introduce an essential feature of the Motech platform: the Event system. The Event system allows our module to communicate with other modules by emitting and listening for events. We also introduce the Tasks module, which allows us to wire up Motech events using a graphical user interface. Finally, we introduce Motech Data Services, which we use to persist entities. Using these tools, we modify the archetype-generated code to save a new record whenever users request a URL defined by our module.

This tutorial assumes you are at least somewhat familiar with Java, [Maven](#), and the [Spring Framework](#), and have completed the instructions to *set up your development machine*. We use [IntelliJ IDEA 13 Community Edition](#) as the

integrated development environment, but this is not a requirement.

This tutorial was written with Motech 0.24.

6.1.2 Generating the Module from Archetypes

Motech is a modular system. Modules are units of functionality that encapsulate application-specific business logic in a reusable package. The Open Service Gateway initiative specification (OSGi) provides the framework to describe this modular architecture: each Motech module is also an OSGi bundle. The OSGi host manages the lifecycle of a module (adding, starting, stopping, and removing), and allows a module to expose services for use by other modules.

To mitigate some of the complexity of configuring OSGi bundles, the Motech platform provides a number of Maven archetypes. The archetypes are also modular: each new module begins from the minimal bundle archetype, and additional capabilities—serving web requests, managing data, and more—are enabled by adding their respective archetypes to the minimal bundle’s foundation.

Generating the Minimal Bundle

The minimal bundle archetype generates the basic project layout and configuration sufficient to begin module development in Motech. To generate the bundle, open a terminal from the directory you wish to contain the project, then enter the following command:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

The flags given with the Maven command instruct Maven to generate a minimal bundle from the 0.24-SNAPSHOT branch of the Motech repository. Once the process completes, the generated project resides in a folder named after the artifact ID, in this case “helloworld.”

Tour of the Minimal Bundle

First, import the module into your favorite IDE. If using IntelliJ IDEA, from the “Welcome to IntelliJ IDEA” splash screen, select *Import Project*, navigate to the *helloworld* folder, and click *OK*. Select *Import project from external model*, select *Maven*, and accept the defaults by clicking “Next” and finally “Finish” through the remaining options.

The minimal bundle archetype generates a standard Maven project layout. In the Java folder, the archetype created a service package under our top level `org.motechproject.helloworld` package. Inside the package, the archetype created the interface and implementation of a very simple Spring service. We’ll modify this service later in the tutorial to make it a bit more interesting.

As for configuration, the *resources/META-INF* directory contains folders for Motech and Spring XML files. In the *motech* folder, the *applicationContext.xml* file enables Spring component scanning of our base Java package, and declares details of our module as a `ModuleRegistrationData` bean. In the *spring* folder, the *blueprint.xml* file exposes our `HelloWorldService` as an OSGi service. Since we activated repository support when generating the minimal bundle, the `HelloWorldRecordService` is also exposed. This code will be added later in the tutorial, when we add the repository archetype. Similarly, we get an OSGi reference to the `HelloWorldRecordsDataService`, also to be added later. Any additional references to Motech platform services will be added to this blueprint file.

Adding the HTTP Archetype

Now that we have some perspective on the basic module layout, let’s add the HTTP archetype into the mix. With a terminal open at the same folder you issued the first Maven command, enter the following:


```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

This archetype pulls in additional dependencies to interact with and integration test Spring web servlets. Returning to the IDE, notice that our top level package now contains an additional subpackage called `web`, with a simple Spring controller. The controller makes our module a little more interesting by injecting our previously generated `HelloWorldService` and providing a URL route to exercise the service's public API.

The HTTP archetype also creates a `webapp` folder in the `resources` directory. This folder contains the module's static files, including HTML partials, stylesheets, JavaScripts, and internationalized messages. Motech modules use the [AngularJS](#) framework to drive the front-end client, so the archetype created the top-level module, controllers, directives, and services necessary for a simple Angular client in the `js` folder.

Adding the Repository Archetype

As the final step in setting up our basic Hello World module, let's generate the repository archetype code with the following Maven command:

```
mvn archetype:generate -DinteractiveMode=false -DarchetypeRepository=http://nexus.motechproject.org/
```

The repository archetype created two new packages, `domain` and `repository`, which contain a simple data model and repository service, respectively. In addition, the archetype added an interface and implementation to the service package, so we can interact with the data layer.

Taking a closer look at the `domain` and `repository` packages, the `HelloWorldRecord` is a typical Java bean that models a record with name and message fields. The class-level `@Entity` annotation identifies the record as a data type to be managed and persisted by the core [Motech Data Services \(MDS\)](#) module. The MDS `@Field` annotations provide object-relational mappings between the bean's fields and columns in the database. The `HelloWorldRecordsDataService` interface extends the base `MotechDataService` interface, inheriting functionality to provide basic CRUD operations for our `HelloWorldRecord` objects. Using the MDS `@Lookup` annotation, we provide a custom method by which to find a `HelloWorldRecord`, in this case by name. Additional custom lookups can be defined here.

In the service package, the `HelloWorldRecordService` injects the data service and exposes a public interface by which to retrieve and persist records.

Building and Deploying the Module

To build the project in Maven, create a new run configuration in IntelliJ by clicking *Run -> Edit Configurations...*. Click the green plus sign to add a new configuration, and select Maven. Name the configuration "Maven clean install", enter "clean install" in the command line field, and click *OK*. Finally, click the *Run* button in the upper right hand corner to package the module.

Alternatively, from the command line, change directory to the `helloworld` module's folder and type "mvn clean install".

Once the build succeeds, the project directory will contain our module packaged as a jar in the `/target` subfolder. With your Tomcat server running, open a browser and navigate to <http://localhost:8080/motech-platform-server/>, then log in with your administrator credentials. Click the *Admin* tab, then click *Manage Modules* from the sidebar navigation. In the dropdown labeled *Install module from*, choose *File*, then click the *Select File* and choose our packaged `helloworld-0.1-SNAPSHOT.jar`. Click the *Start on Install* button to toggle the icon to a check mark, and then finally click *Install or Update*.

If all goes well, our *Hello World Module* should appear alongside the other installed modules in the user interface. The state should be "Active."

As a small test of our web controller, navigate your browser to <http://localhost:8080/motech-platform-server/module/helloworld/sayHello>. The controller should respond with the JSON string `{"message": "Hello World"}`.

6.1.3 An Event-Driven Hello World

In this second phase of the tutorial, we implement a feature that creates a new record whenever a user requests the `sayHello` URL route defined by our module. To accomplish this, we emit an event when the URL is accessed, listen for an event instructing our module to create a new record, and finally wire these two separate events together using the *Tasks* module.

Declaring the Event System Dependency

To include events in our module, we first need to declare the dependency on the event system in our *pom.xml* file. Between the `<dependencies>` opening and closing tags, add the following:

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

Secondly, our module will hook into Motech's Event Relay, so we need to add a reference to the relay alongside our other OSGi configuration in *blueprint.xml*:

```
<osgi:reference id="eventRelayOsgi" cardinality="0..1" interface="org.motechproject.event.listener.EventListener">
```

With these dependencies, we are now able to inject the event relay into our own code, send events, and define event listeners.

Event Subjects and Event Parameters

Motech events are identified by a String value called the subject. To encapsulate these values as constant values, let's create an event package under our main module package to contain the following class:

```
package org.motechproject.helloworld.event;

public final class HelloWorldEventSubjects {

    public static final String SEND_HELLO = "helloworld_hello_event";

    public static final String CREATE_HELLO_RECORD = "helloworld_create_hello_record";

    private HelloWorldEventSubjects() {}
}
```

Similarly, each event can be supplied with parameters, represented by a map of key-value pairs in which the keys are also constant String values. This class encapsulates those values:

```
package org.motechproject.helloworld.event;

public final class HelloWorldEventParams {

    public static final String NAME = "name";

    public static final String MESSAGE = "message";

    private HelloWorldEventParams() {}
}
```

Finally, let's create a helper class to simplify packaging our parameters together into a functional Motech event:

```
package org.motechproject.helloworld.event;

import org.motechproject.event.MotechEvent;

import java.util.HashMap;
import java.util.Map;

public final class HelloWorldEvents {
    public static MotechEvent sendHelloWorldEvent(String name, String message) {
        Map<String, Object> params = new HashMap<>();
        params.put(HelloWorldEventParams.NAME, name);
        params.put(HelloWorldEventParams.MESSAGE, message);
        return new MotechEvent(HelloWorldEventSubjects.SEND_HELLO, params);
    }

    private HelloWorldEvents() {}
}
```

Given a name and a message, our static helper method will bundle up the parameters and return a Motech event, which in turn will be passed to the event relay.

Sending an Event

To send our event, let's modify our implementation of the HelloWorldService to send events whenever the sayHello method is called:

```
package org.motechproject.helloworld.service.impl;

import org.motechproject.event.listener.EventRelay;
import org.motechproject.helloworld.event.HelloWorldEvents;
import org.motechproject.helloworld.service.HelloWorldService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service("helloWorldService")
public class HelloWorldServiceImpl implements HelloWorldService {

    @Autowired
    private EventRelay eventRelay;

    @Override
    public String sayHello() {
        eventRelay.sendEventMessage(HelloWorldEvents.sendHelloWorldEvent("HWEvent", "Hello world!"));
        return "Hello World";
    }
}
```

In the HelloWorldServiceImpl listing, we declare a private field to contain a reference to the Event Relay. Annotating the field with Spring's @Autowired annotation injects the dependency we declared earlier in the *blueprint.xml* configuration file. Then, in the body of the sayHello method, prior to the return statement, we use the event relay's sendEventMessage method to send our Motech event.

And that's it! Any Motech module that defines a listener for our SEND_HELLO subject can inspect our event and its parameters.

Listening for Events

In our Hello World module, we provide a data service layer to create `HelloWorldRecord` instances, but currently do not make use of it. Rather than keep that functionality to ourselves, let's define an event listener that allows other modules to send us requests to create new records. First, create a new subpackage to the event package called `handler`, then add the following class:

```
package org.motechproject.helloworld.event.handler;

import org.motechproject.event.MotechEvent;
import org.motechproject.event.listener.annotations.MotechListener;
import org.motechproject.helloworld.domain>HelloWorldRecord;
import org.motechproject.helloworld.event>HelloWorldEventParams;
import org.motechproject.helloworld.event>HelloWorldEventSubjects;
import org.motechproject.helloworld.service>HelloWorldRecordService;
import org.springframework.beans.factory.annotation.Autowired;

@Component
public class HelloWorldEventHandler {

    @Autowired
    private HelloWorldRecordService helloWorldRecordService;

    @MotechListener(subjects = {HelloWorldEventSubjects.CREATE_HELLO_RECORD})
    public void handleCreateHelloEvents(MotechEvent event) {
        String name = (String)event.getParameters().get(HelloWorldEventParams.NAME);
        String message = (String)event.getParameters().get(HelloWorldEventParams.MESSAGE);
        HelloWorldRecord record = new HelloWorldRecord(name, message);
        helloWorldRecordService.add(record);
    }
}
```

In our event handling class, the `@MotechListener` annotation does the work of transforming a standard Java method into an event-handling method. Using the annotation's `subjects` element, we declare our interest in receiving `CREATE_HELLO_RECORD` events. Our event-handling method accepts the `MotechEvent` as an argument, extracts the name and message parameters, creates a new `HelloWorldRecord`, and finally uses the injected data service to persist the new record.

Since we are listening for a different event than the one we emit, there is no interaction between our module's event-emitting functionality and event-listening functionality. In the next section, we look at a way to integrate these two systems with the Tasks module's graphical user interface.

Task Triggers and Task Actions

The Tasks module allows implementers to define interactions between modules through a graphical user interface. A Task consists of a trigger and an action. A trigger is a standard Motech event that has been exposed to the Task module. An action is an event for which the implementer's module provides a listener. Between the trigger and action, an implementer can add filters to control whether a task should run based on conditional logic. Also, tasks may use data loaders to query Motech Data Services for additional information.

In the listings that follow, we will declare a Task channel for our module in the form of a JSON document. We will expose our `SEND_HELLO` event as a task trigger, declare our `CREATE_HELLO_RECORD` event as a Task action, and then link the two in Motech's GUI.

First, create a new file called `task-channel.json` in our module's `main/resources` directory. Each Task channel must define a display name, a module name, and a module version as JSON properties. A Task channel may provide a list of triggers, identified by the `triggerTaskEvents` property. Each trigger must define a display name and event

subject, and may provide a list of parameters passed with the event, in which case each event has an event key and a display name. Similarly, a Task channel may provide a list of Task actions, identified by the `actionTaskEvents` property. Like triggers, each action must provide a display name and an event subject, and an optional list of parameters identified by key and display name:

```
{
  "displayName": "helloworld.task.channel.name",
  "moduleName": "${project.artifactId}",
  "moduleVersion": "${parsedVersion.osgiVersion}",
  "triggerTaskEvents" : [
    {
      "displayName" : "helloworld.task.trigger.send_hello.name",
      "subject" : "helloworld_hello_event",
      "eventParameters" : [
        {
          "eventKey" : "name",
          "displayName" : "helloworld.task.trigger.send_hello.param.name"
        },
        {
          "eventKey" : "message",
          "displayName" : "helloworld.task.trigger.send_hello.param.message"
        }
      ]
    }
  ],
  "actionTaskEvents" : [
    {
      "displayName" : "helloworld.task.action.create_hello_record.name",
      "subject" : "helloworld_create_hello_record",
      "actionParameters" : [
        {
          "key" : "name",
          "displayName" : "helloworld.task.action.create_hello_record.param.name"
        },
        {
          "key" : "message",
          "displayName" : "helloworld.task.action.create_hello_record.param.message"
        }
      ]
    }
  ]
}
```

In the listing above, the subjects for the trigger and action are the hard-coded String constants we defined in our `HelloWorldEventSubjects` class. Since our `HelloWorldRecord` provides fields for name and message, we pass that data as parameters on the trigger, and accept that data as parameters of the action.

Throughout the JSON listing, the values we provide as display names are not String literals, but rather references to Strings in our message properties files. To provide the String literals for the references above, open our `main/resources/webapp/messages/messages.properties` file and append the following:

```
#Tasks
helloworld.task.channel.name=Hello World

helloworld.task.trigger.send_hello.name=Hello World
helloworld.task.trigger.send_hello.param.name=Name
helloworld.task.trigger.send_hello.param.message=Message

helloworld.task.action.create_hello_record.name=Create Hello World Record
```

```
helloworld.task.action.create_hello_record.param.name=Name
helloworld.task.action.create_hello_record.param.message=Message
```

Now, all of the components of our task channel will display correct English values in Motech’s graphical user interface. Before we turn to the user interface, let’s rebuild the module with `mvn clean install` and install our module in the *Admin* -> *Manage Modules* user interface, as we did when touring the minimal bundle.

Creating a Task in the User Interface

Once the module has been updated, we’ll create the relationship between task trigger and action in the Motech user interface. Navigate to the Tasks interface by clicking *Modules* -> *Tasks*. Click the *New Task* button and enter “Create Hello Record” in the *Task Name* field. Our module appears in the Trigger dropdown with the default module icon, a clipboard. Click on the icon, and select Hello World from the available triggers.

New buttons will appear allowing us to add filters, data sources, and actions. Click the *Add action* button, then, from the *Channel* dropdown menu, select our Hello World module. Click the *Action* dropdown menu, and select the *Create Hello World Record* action that we defined as part of our Task channel.

Finally, notice that the fields we declared as parameters of our Task trigger appear as draggable elements in the *Available Fields* list. Also notice that our Task action allows us to enter a name and message as text in two form input fields. To complete the association between the trigger and action parameters, drag the *Name* element to the *Name* input field, and the *Message* element to the *Message* input field. Once the form fields are populated, click the *Save & Enable* button to activate our new task.

Final Walkthrough

To review, in the Event-driven Hello World section of the tutorial, we added the Motech event system to our module. We defined two event subjects: a `SEND_HELLO` event that our module emits to other modules in the system, and a `CREATE_HELLO_RECORD` event to which our module listens for and responds. Our module emits the `SEND_HELLO` event whenever the `HelloWorldService`’s `sayHello` method is called (in this case, when a user loads the `/sayHello` route as defined by the `HelloWorldController`). Using the Tasks module’s user interface, we defined a task linking the `SEND_HELLO` and `CREATE_HELLO_RECORD` events as a task trigger and task action, respectively. Our `HelloWorldEventHandler` listens for `CREATE_HELLO_RECORD` and adds new `code:HelloWorldRecord` instances to our Motech Data Services repository.

To verify that we implemented our feature correctly, load <http://localhost:8080/motech-platform-server/module/helloworld/sayHello> in your browser. The response sent to the browser remains the same. To check for the new record, click *Modules* -> *Data Services* in the Motech user interface. Select the *Data Browser* tab, then click on the `HelloWorldRecord` under the heading for our *Hello World* module. In the list of record instances that appear, we should find a record with the name “HWEvent” and the message “Hello world!”

6.2 IVR Demos

First of all, just in case you didn’t know, IVR (or [Interactive Voice Response](#)) is a system that enables computers to interact with humans using a phone. The computer interacts with the human using pre-recorded or synthesized messages and the human either speaks back or uses her phone keypad (also called [DTMF](#)) to interact with the computer.

6.2.1 Initial Setup

In addition to the Motech platform, and the IVR module, you’ll also need to build and/or install the SMS Module.

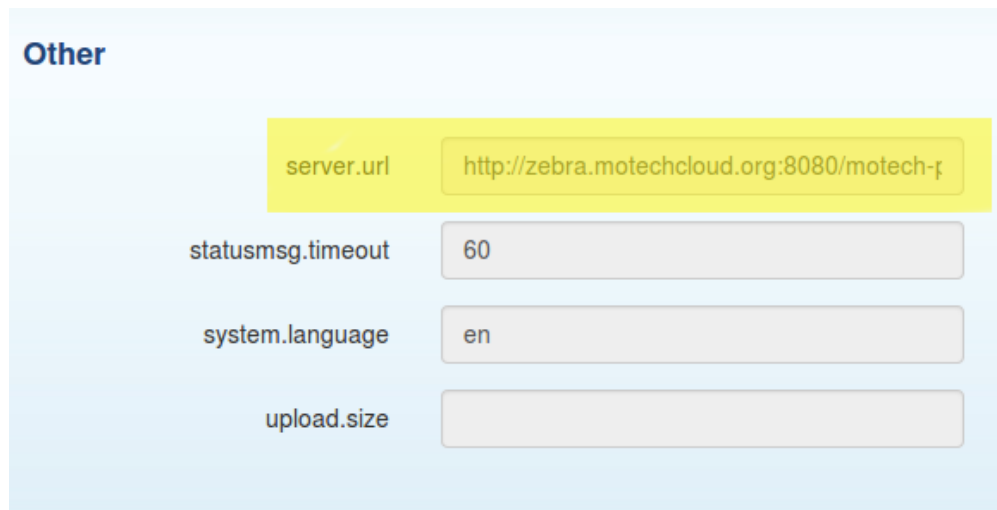
6.2.2 Server Settings

Also be sure that the `server.url` property is properly set to your server's URL and that your server is publicly reachable from the internet. If your server's config source is file based, locate the `motech-settings.properties` file and make sure the `server.url` is set. If your server's config is done through the UI, then navigate to **Admin / Settings** and set the `server.url` property there.

So, for example, if your server's public address was `zebra.motechcloud.org` and it was accessible on port 8080, then you should see:

```
server.url=http://zebra.motechcloud.org:8080/motech-platform-server
```

Confirm the setting (or set it [#]) by clicking **Admin / Settings**:



The screenshot shows a settings interface with a light blue background. At the top left, the word "Other" is displayed in blue. Below it, there are four settings. The first setting, "server.url", is highlighted with a yellow background and has a checkmark icon to its left; its value is "http://zebra.motechcloud.org:8080/motech-". The other three settings are "statusmsg.timeout" with a value of "60", "system.language" with a value of "en", and "upload.size" which is currently empty.

Property	Value
server.url	http://zebra.motechcloud.org:8080/motech-
statusmsg.timeout	60
system.language	en
upload.size	

6.2.3 SMS Module Config

For the demos that have you send or receive SMS, you need a valid SMS config. You'll need to establish an account with an SMS provider and then configure the SMS Module accordingly. In these demos we're using [Plivo](#). To confirm your SMS Settings, click **Modules / SMS / Settings**:

The screenshot shows the MOTECH Admin interface. On the left is a sidebar with a menu containing: Modules, Admin, Security, CMS Lite, Data Services, Email, SMS (selected), Scheduler, and Tasks. The main content area has a breadcrumb trail: Home / SMS / Settings. Below this are tabs for Send SMS, Log, and Settings (selected). The title is 'SMS Provider Configurations'. There is a green '+ Add Configuration' button. Below that, a configuration for 'plivo' is shown with a star icon and a dropdown arrow. The configuration fields are: Name (plivo), Template (Plivo), Maximum Retries (3), Split Message Header (Msg \$m of \$t), Split Message Footer (...), Exclude footer from last split message (checked), AUTH ID (blurred), AUTH TOKEN (blurred), and From (blurred). At the bottom of the configuration box is a red 'Delete' button. Below the configuration box are 'Cancel' and 'Save' buttons.

If, in addition to sending, you're going to be receiving SMS, you must also tell your SMS provider what to do when they receive an SMS. There typically will be some way of setting that up on their website. They probably need a URL to send an HTTP request to. This is where you give them the address of your Motech server. So if your server is accessible on the web at `http://zebra.motechcloud.org:8080` the complete URL you would provide your SMS provider would be `http://zebra.motechcloud.org:8080/motech-platform-server/module/sms/incoming/plivo` where `plivo` is the name of the SMS Config you created for that SMS provider.

6.2.4 The Demos, Finally...

There are three IVR demos:

Generic VXML IVR Provider Demo: Receiving Calls

You accept phone calls, prompt for and record a code, and then send the code in an SMS to the caller.

In more details:

1. You call the IVR provider ¹.
2. The IVR provider calls Motech ² to request the `hello` VXML template which will tell it what to do. This demo uses VXML but, depending on the provider, it could also be CCXML or some proprietary language.
3. Motech returns the requested `hello` VXML template.
4. The IVR provider executes the instructions in `hello` which prompt the caller to type in a number.
5. The caller (that's you!) types in a number.
6. The IVR provider sends the number to Motech along with a request for the `thankyou` template.
7. Motech receives the number you typed, sends a Motech Event, and returns the requested `thankyou` template.
8. The IVR provider executes the instructions in `thankyou`: say 'Thank you' and hang up.
9. Meanwhile, on the Motech side, the Tasks module was waiting for a *[we received a template request and the value of the `callStatus` parameter is ANSWERED]* ³ Motech Event. When it finally receives it, it extracts the number you typed, creates a 'You chose *x*' message and sends an SMS ⁴ to the number that originated the call, which it also receives as a standard parameter from the IVR provider.
10. You receive 'You picked *x*' SMS. Nifty, eh?

IVR Provider

For this demo we used [Voxeo](#), a generic VXML/CCXML provider.

IVR Settings

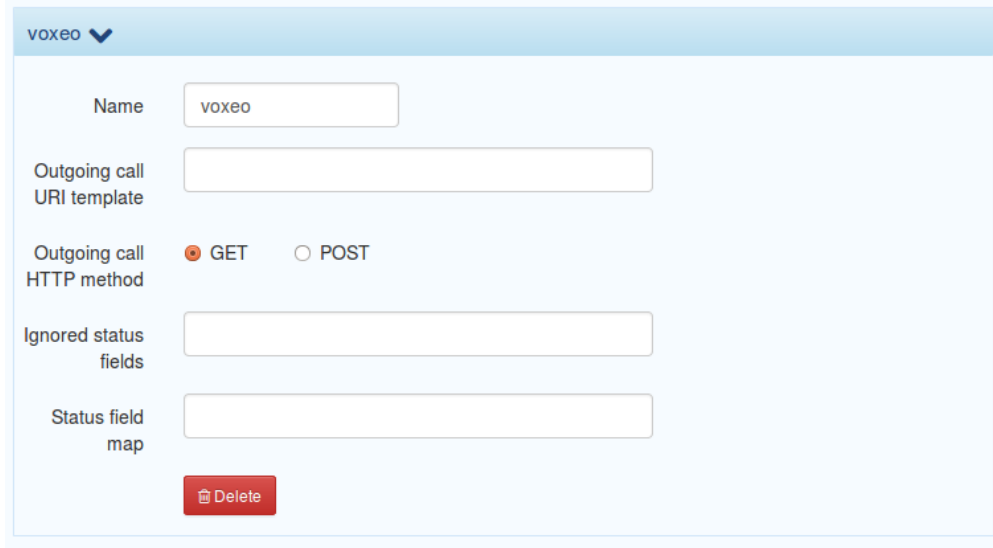
We need to create an IVR Config so the IVR module knows what to do when it receives HTTP requests from IVR providers. Click on **Modules / IVR / Settings**:

¹ IVR providers will typically give you a phone number you can use to call your application on their system. They also will require that you give them a publicly accessible URL so they can request the VXML that you want to be executed when a call is received at this number.

² More precisely makes a REST call to the IVR module at the `/template` HTTP endpoint.

³ To differentiate it from other template requests.

⁴ From the SMS provider you configured.



Note: Because we're using the Voxeo IVR provider we named our config **voxeo**. The name isn't important, but it's a good idea to use a simple one as it'll be part of the URL you'll provide your IVR provider so it knows where to get the VXML.

A little VXML

Here's the hello VXML template:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version = "2.1">
  <form id="enterCode">
    <field name="code" type="digits?minlength=1;maxlength=1">
      <prompt>
        Hello! Please pick a number between 0 and 9.
      </prompt>
    </field>
    <filled>
      <prompt>
        You picked <value expr="code" />.
      </prompt>
      <assign name="from" expr="session.callerid" />
      <assign name="providerCallId" expr="session.sessionid" />
      <assign name="callStatus" expr="'ANSWERED'" />
      <submit name="sendCode" next="http://zebra.motechproject.org:8080/motech-platform-serv
    </filled>
  </form>
</vxml>
```

Note: The selected number is passed as one of query parameters (`code`) the IVR provider sends along with its request for the `thankyou` template.

And the `thankyou` VXML template, which simply says 'Thank you' and hangs up:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version = "2.1" >
  <form>
```

```

    <block>
      <prompt>
        Thank you
      </prompt>
    </block>
  </form>
</vxml>

```

Creating the Templates

To add these two templates, click **Module / IVR / Settings** and then **+ Add Template**:

Templates

[+ Add Template](#)

hello >

thankyou ▼

Name	thankyou
Value	<pre> <?xml version="1.0" encoding="UTF-8"?> <vxml version = "2.1" > <form> <block> <prompt> Thank you </prompt> </block> </form> </vxml> </pre>

[Delete](#)

Creating the Task

We need to create a task where the trigger is an IVR template request where the call status is **ANSWERED** and the action is to send an SMS to the original caller with the code she entered in the message:

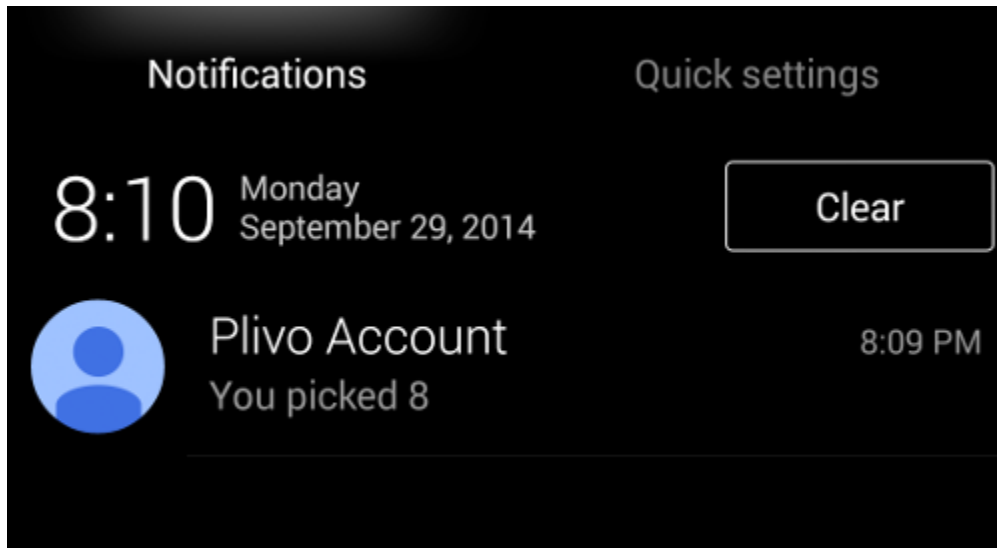
The screenshot shows a configuration window titled "Trigger » IVR : Template Request". It has a "Filters" section with a filter for "trigger.call_status" using the "Is" operator and "Equals" value, set to "ANSWERED". Below the filter is a "Help" button and a "+ Add another filter" button. There are also three orange buttons: "+ Add filter set", "+ Add data source", and "+ Add action". The "Action » SMS : Send SMS" section is expanded, showing "Available Fields" like "Motech Timestamp", "Provider Timestamp", "Config", "Template", "From", "To", "Call Direction", "MOTech Call ID", "Provider Call ID", and "Provider Extra Data". The "Channel" is set to "SMS" and the "Action" is "Send SMS". The "Recipient(s)" field contains "+1{{trigger.from}}". The "Message" field contains "You picked {{trigger.provider_extra_data.code}}". The "Configuration" field is set to "plivo". The "Delivery Time" field has a placeholder "Select date or drag trigger field" and a "Select date" button.

Note: `code` is extracted from the Motech event payload with `{{trigger.provider_extra_data.code}}`

Note: A `+1` is added to the SMS recipient because our sample SMS provider, [Plivo](#), needs it.

Et Voila!

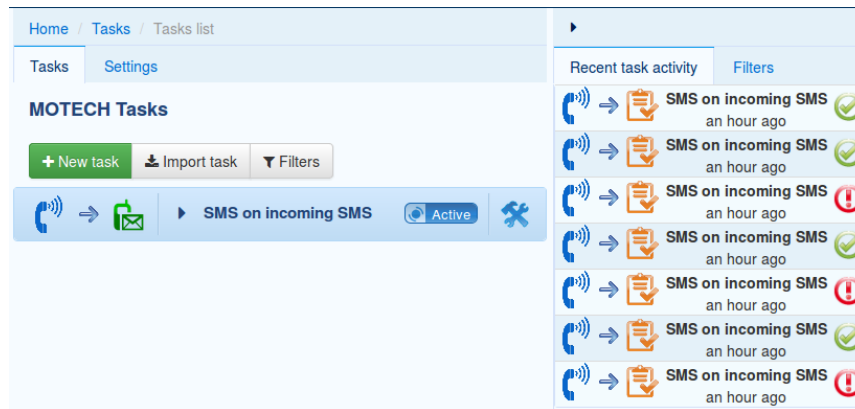
Now call your application at the phone number that your IVR provider gave you, then listen to the “Hello! Please pick a number between 0 and 9.” prompt, type in a number (say 8). The IVR system will say “You picked 8. Thank you”, then the call will disconnect and soon enough you should receive an SMS:





Did it work?

In addition to the obvious sign that you're receiving an SMS from your SMS provider, there are other ways you can check your application works.










- You can look at the Tasks module's **Recent task activity** list to see if your task was executed:



- Or you can look at your task's history:



SMS on incoming SMS
 Recent Activity for Task
 Active

Filter Activity ALL
 Activities Per Page 10
Clear history

Status	Message	Information
	Action was performed correctly	an hour ago
	Action was performed correctly	an hour ago
	Unrecognized error. Show stack trace	an hour ago
	Action was performed correctly	an hour ago
	Unrecognized error. Show stack trace	an hour ago
	Action was performed correctly	2 hours ago
	Unrecognized error. Show stack trace	2 hours ago
	Action was performed correctly	2 hours ago
	Unrecognized error. Show stack trace	2 hours ago

Clear history
Back

- You can also browse the IVR CallDetailRecord entity in the database using the MDS Data Browser:

Id	Config Name	From	To	Call Direction	Call Status	Template Name	Provider Extra Da	Motech Call Id	Provider Call Id	Motech Timestamp	Provid
14	voico				ANSWERED	thankyou	code: 3		8067430b6f3d4a26	2014-09-30 03:20:4	
13	voico				UNKNOWN	hello	session.accountId: ; session.callerId: 20; session.sessionId: ; session.parentSessionId: ; session.virtualPlatformId: ; session.calledId: 20			2014-09-30 03:20:1	
12	voico				ANSWERED	thankyou	code: 0		36cdff8db71cf878f	2014-09-30 03:09:1	
11	voico				UNKNOWN	hello	session.accountId: ; session.callerId: 20; session.sessionId: ; session.parentSessionId: ; session.virtualPlatformId: ; session.calledId: 20			2014-09-30 03:09:1	
10	voico				ANSWERED	thankyou	code: 4		47e7c2b443b8b5fa	2014-09-30 03:06:4	
9	voico				UNKNOWN	hello	session.accountId: ; session.callerId: 20; session.sessionId: ; session.parentSessionId: ; session.virtualPlatformId: ; session.calledId: 20			2014-09-30 03:06:4	

Note: Our simple VXML application did not bother to set the CallDirection nor many other fields in its status callback to Motech.

- Yet another way to see how your application would be to be to look at the SMS log or, for even more details, the Server Log.

Generic VXML IVR Provider Demo: Making Calls

Upon receiving an SMS, call the sender back and speak the content of the SMS.

The details:

- You send an SMS to the number provided to you by your SMS provider ⁵.
- The SMS module receives a `/incoming` HTTP request from the SMS provider and sends a corresponding `inbound_sms` Motech Event.

⁵ You must also tell your SMS provider what to do when they receive an SMS, remember?

3. The Tasks module listens to the `inbound_sms` Motech event and triggers ⁶ an outbound IVR call, passing the text of the SMS as a parameter named `message`.
4. Your IVR provider receives the outbound call request.
5. Your IVR provider then asks Motech for the VXML template ⁷, executes the VXML.
6. You receive a phone call, pick up, hear the IVR computer voice speak the content of your SMS.

Creating a Config

In order for the IVR provider to initiate a call, we need to create a Config, click **Modules / IVR / Settings**:

The screenshot shows the 'voxeo' configuration page in the Motech IVR Settings. The 'Name' field is set to 'voxeo'. The 'Outgoing call URI template' field contains the URL: `http://api.voxeo.net/SessionControl/4.5.41/CCXML10.start?tokenid=[tokenid]&callerid=[callerid]&to=[to]`. The 'Outgoing call HTTP method' is set to 'GET'. The 'Ignored status fields' and 'Status field map' fields are empty. A 'Delete' button is visible at the bottom.

Note: We named ours **voxeo**. Note that it's a bit different than the one we created in the *incoming-calls* demo, we need to tell the IVR module how to reach the IVR provider by settings the `outgoingCallUriTemplate` and `outgoingCallMethod` properties.

The VXML

We need a simple VXML script that will say something that was passed to the IVR provider via the outgoing call initiation HTTP request:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version = "2.1" >
  <form>
    <block>
      <prompt>
```


⁶ By issuing an HTTP request to an URL provided by your IVR provider.

⁷ At the URI you told your provider to find the VXML for outgoing calls from your number.

```
        <value expr="session.connection.ccxml.values.message" />
      </prompt>
    </block>
  </form>
</vxml>
```

Note: `session.connection.ccxml.values.message` implies Motech will have to add a parameter named `message` to the HTTP request querystring to the IVR provider.

We'll name this template `say`:



The screenshot shows a web interface for managing templates. At the top, there's a header bar with the text "say" and a dropdown arrow. Below this, there's a form with two main sections: "Name" and "Value". The "Name" section has a text input field containing the word "say". The "Value" section has a large text area containing a VXML script. The script is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml version = "2.1">
  <form>
    <block>
      <prompt>
        <value expr="session.connection.ccxml.values.message" />
      </prompt>
    </block>
  </form>
</vxml>
```

Below the text area, there is a red button with a trash icon and the text "Delete".

Gluing things together with the Tasks module

Let's create a task which, upon receipt of an SMS, initiates an outgoing call and passes a message for the VXML script to say:

MOTECH Edit Task

Task Name:

Task Description:

► 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: Action:

Config:

Parameters: to:
message:

Help: modifications Help: custom syntax

Note: we specify the number to call (in this case the sender of the SMS) and what do say (the content of the SMS) using a map notation in the action `Parameters` field.

Drum roll...

Now send an SMS with a simple 'hello world'. Wait a few seconds ⁸. You should receive a 'hello world' voice call from your IVR provider. Et voila!

Notes

As in the previous example, you can check the **Recent tasks activity** pane on the Tasks module, or check the SMS or the IVR log to see what happened.

It's important to note that this very crude & simple demo does not deal with call status, so the IVR `CallDetailRecord` log will not be very useful.

Proprietary IVR Provider Demo: KooKoo

Introduction

KooKoo is a popular IVR provider from India. They are not a standard CCXML/VXML provider, but instead offer a language that's somewhat similar to a very simplified version of

⁸ Crossing your fingers always helps

VXML, named [KooKoo Tunes](#).

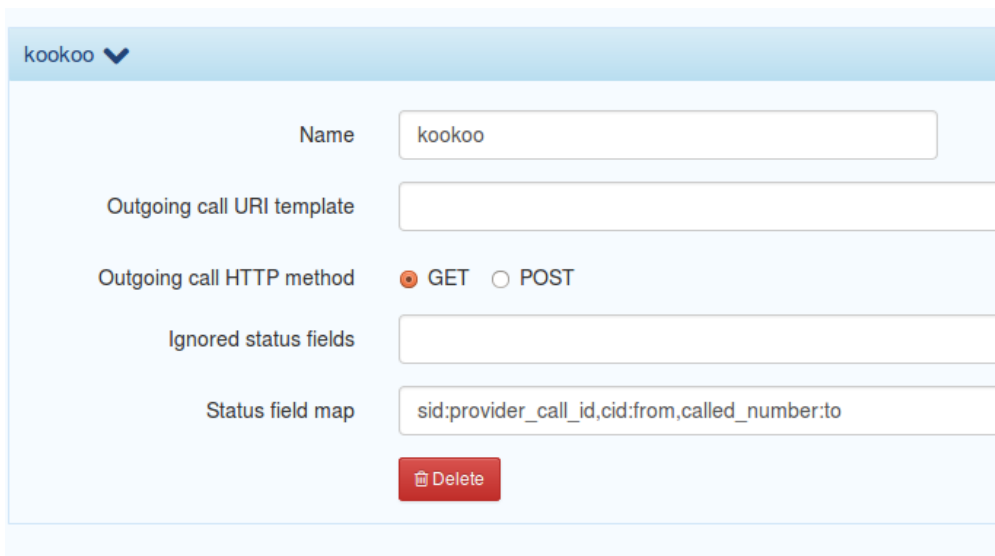
This demo is similar to the Incoming SMS Demo but uses KooKoo's simplified XML language instead of standard VXML. We'll only explain the IVR specific parts here, to create the full demo that sends you an SMS, please see the demo.

Initial Setup

You'll need to setup a KooKoo account.

IVR Config

In this demo we're only receiving calls (to the phone number provided to us by KooKoo) so we only need create a minimal config, click **Modules / IVR**:

The screenshot shows the KooKoo IVR configuration interface. At the top, there's a header with the 'kookoo' logo and a dropdown arrow. Below this, the configuration form has several fields: 'Name' with the value 'kookoo', 'Outgoing call URI template' which is empty, 'Outgoing call HTTP method' with radio buttons for 'GET' (selected) and 'POST', 'Ignored status fields' which is empty, and 'Status field map' with the value 'sid:provider_call_id,cid:from,called_number:to'. At the bottom of the form is a red 'Delete' button with a trash icon.

Note: We're mapping three of the parameters sent back to Motech by KooKoo: `sid` will be mapped to `provider_call_id`, `cid` will be mapped to `from` (the number of the phone placing the call) and `called_number` will be mapped to `to` (the number of the phone receiving the phone, in this case the number assigned to you by KooKoo)

Two KooKoo Tunes

The funny people at KooKoo call an XML file a "Tune". We're creating two by going to **Modules / IVR** and clicking on **+ Add Template**. The first one says "Hello from KooKoo, please type a number from 0 to 9" and then sends the response back in a `data` parameter and requests the next thing to do from `goodbye` template:

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <playtext>Hello from KooKoo</playtext>
  <collectdtmf l="1">
    <playtext>Please type a number from 0 to 9</playtext>
  </collectdtmf>
```

```
<gotourl>http://yourserver.com/motech-platform-server/module/ivr/template/kookoo/goodbye
</Response>
```

Name it **helloworld** and copy/paste the XML above in the value text area:

ModulesAdminSecurity

Home / IVR / Templates

TemplatesSettings

Templates

+ Add Template

helloworld

Name

helloworld

Value

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <playtext>Hello from KooKoo</playtext>
  <collectdtmf l="1">
    <playtext>Please type a number from 0 to 9</playtext>
  </collectdtmf>
  <gotourl>http://yourserver.com/motech-platform-server/module/ivr/template/kookoo
/ goodbye</gotourl>
</Response>
```

Delete

Cancel

Save

The goodbye template simply says “Thank you” and hangs up:

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <playtext>Thank you</playtext>
  <hangup></hangup>
</response>
```

The first script sends an additional query parameter named `data` containing the key pressed during the call to Motech. Since `data` is not a standard property, it will be added to the `CallDetailRecord`'s `providerExtraData` map property. Actually KooKoo sends yet another query parameter named `event` with the value `GotDTMF`, we'll use `event` to filter the callback from KooKoo such that we pick the one which contains the `data` parameter.

Let's Create a **:doc:'Task <tasks>'**_____

We need to create a task where the trigger is an IVR template request and where the **event** key in the `providerExtraData` map field is equal to `GotDTMF`. We also want the action to send an SMS to the original caller with the code she entered in the message:

The screenshot shows the MOTECH configuration interface. At the top, there's a section titled 'Trigger » IVR : Template Request'. Below it is a 'Filters' section with a dropdown menu showing 'trigger.provider_extra_data.' and two 'Select One' buttons. There are also 'Help' and '+ Add another filter' buttons. Below the filters, there are three orange buttons: '+ Add filter set', '+ Add data source', and '+ Add action'. The main section is titled 'Action » SMS : Send SMS'. It has an 'Available Fields' section with buttons for 'Motech Timestamp', 'Provider Timestamp', 'Config', 'Template', 'From', 'To', 'Call Direction', 'Call Status', 'MOTECHE Call ID', 'Provider Call ID', and 'Provider Extra Data'. Below this, there are fields for 'Channel' (set to 'SMS'), 'Action' (set to 'Send SMS'), 'Recipient(s)' (set to 'From'), 'Message' (set to 'You chose {{trigger.provider_extra_data.event}}'), 'Configuration' (set to 'plivo'), and 'Delivery Time' (set to 'Select date'). There are also two help buttons: 'Help: modifications' and 'Help: custom syntax'.

Note: The filter source is partially obscured, here it is in full: `{{trigger.provider_extra_data.event}}`

Note: **data** (the parameter containing the key pressed, remember?) is extracted from the Motech event payload with `{{trigger.provider_extra_data.data}}`

Et Voila!

Now call your application at the phone number that KooKoo gave you, then listen to the lady⁹ say 'Hello from KooKoo, please type a number from 0 to 9', type in a number (say 4). She'll say 'Thank you' and will hang up. Soon enough you should receive an SMS with the following message: 'You chose 4'.

6.3 Demo: Create a Message Campaign

Text to come

⁹ The default might not be a lady's voice on your IVR provider, it was on mine.

6.4 Demo: Create a Care Schedule

Text to come

6.5 Demo: Modeling a New System with Data Services

Text to come

6.6 Demo: OpenMRS Schedule Tracking

Text to come

Contribute

Thank you for your interest in contributing to MOTECH. There are many different ways to get involved - regardless of your area of expertise and time commitment, there is likely a useful way for you to contribute. Please peruse the sections below for some instructions on how to get started contributing in specific areas. If there is another way you'd like to help that isn't listed, just [let us know](#) what your interests are and we can help find a project for you.

7.1 Development

Want to help us develop MOTECH? The step by step guide below will help you get started. The instructions may be incomplete, or may contain some errors, so please just let us know if something is missing or incorrect. Our mailing list is populated with helpful people who will help you get going - and will get this documentation up to date with any issues you encounter.

Here is how to get started:

7.1.1 Mailing Lists and Accounts

1. Sign up for the MOTECH Developer [mailing list](#) - this is the best place to get help with any questions about MOTECH development.
2. Create an account on [Gerrit](#), our code review system.
3. Get a Jira account if you'll need to report/track issues - there is no self-serve way to request a Jira account yet, so please just email the MOTECH Developer list and we'll get you set up.

7.1.2 Dev Environment & Tools

1. Configure your *dev machine*.
2. Configure your Git client to *submit changes via Gerrit*.
3. Familiarize yourself with our [CI](#).

7.1.3 Finding Something to Work On

1. Some MOTECH devs have found that reviewing code written by others before they jump in and develop their own is a good way to get their feet wet. If that sounds like your style, feel free to jump into any code review on Gerrit and add your comments.

2. If you'd rather dive right in, we'd recommend that you find a "community" issue from our issue tracker - these are bugs and user stories that we think are good introductory items for MOTECH newbies. See a list of all current "community" issues [here](#).
3. If you're already building your own system on top of MOTECH and you'd like us to incorporate your changes for an issue you've found and fixed, please first check whether the issue already exists in our issue tracker. If you are not sure, please email the mailing list and we'll help you determine whether the issue is already known. Please track your work with a new issue so that we can evaluate it for inclusion in the Platform.

7.1.4 Developing and Submitting Your Code

1. If your fix will be nontrivial, please leverage the mailing list for feedback on your design before you get too far - we are a friendly bunch and can help ensure you are headed in the right direction.
2. When you're ready to push your changes, please squash your commits to keep the change history concise, and write a commit message that conforms to our [guidelines](#).
3. Submit your code using **git push origin** - if you configured your environment correctly, this sends your changes to Gerrit, our code review system.
4. Please incorporate code review feedback and update your patch as needed - once your change has passed code review, one of the project maintainers will merge your change to the repo.
5. Resolve the relevant issue(s) in Jira.

7.2 Documentation

There will be text here.

7.3 Translation

Bonjour!

We are using [Transifex](#) to manage MOTECH translations, which makes it easy for non-geeks to help. If you speak multiple languages and would like to help us make MOTECH multilingual, please start by checking out our translation page to determine whether your language(s) are on our list. Do we have a translation in progress for your language? Great! We'd love your help translating additional strings. Do we not have a translation started for your language? Also great! We'd love your help getting one started.

Either way, please [sign up for Transifex](#) (free), and then [contact us](#). Let us know your Transifex user ID and which language(s) you'd like to work on, and we'll help you get started.

Release Notes

8.1 Current Version

Version 0.24 Release Notes

8.2 Older Versions

[Version 0.19 Release Notes](#)

[Version 0.18 Release Notes](#)

[Version 0.17 Release Notes](#)

[Version 0.16 Release Notes](#)

[Version 0.15 Release Notes](#)

[Version 0.14 Release Notes](#)

[Version 0.13 Release Notes](#)

[Version 0.12 Release Notes](#)

[Version 0.11 Release Notes](#)

[Version 0.10 Release Notes](#)

[Version 0.9 Release Notes](#)

[Version 0.8 Release Notes](#)

[Version 0.7 Release Notes](#)

[Version 0.6 Release Notes](#)

[Version 0.5 Release Notes](#)

[Version 0.4 Release Notes](#)

8.2.1 Version 0.24 Release Notes

Release Date: September 3, 2014

Release Summary

The 0.24 release is primarily dedicated to removing MOTECH's dependency on CouchDB, as well as enhancing MOTECH Data Services (MDS) to make it usable as the data layer for most MOTECH applications. All modules have now been migrated to MDS, with a few exceptions noted below. One notable enhancement to MDS in this

release is the support for relationships between entities (1:1, 1:many, master-detail), in order to enable a number of the module migrations.

This release also features some consolidation of our code repositories (details below) and deprecation of a few modules. The Platform modules were also refactored to reduce their public surface area. Three new modules have been developed: mTraining, Batch, and Hub.

Where to Get it

Source Code: [Platform](#) | [Modules](#)

Binary Distribution

[Platform WAR](#)

Modules:

- Alerts
- Appointments
- Batch
- CMS Lite
- CommCare
- Event Logging
- Hindi Transliteration
- HTTP Agent
- Hub
- Message Campaign
- mTraining
- OpenMRS 1.9
- Pill Reminder
- Schedule Tracking
- SMS

Major Changes

Modules Migrated from CouchDB to MDS

All MOTECH modules (with the exception of the IVR modules mentioned later) are now using MOTECH Data Services for data storage and retrieval. Any modules that are used in MOTECH implementations should likewise be migrated from CouchDB to MDS. The best reference for using MDS at this time is the source code for the existing modules; soon we will provide developer-focused *documentation for MDS*.

New MDS Features

A number of new features were added to MDS in order to support migration of the existing modules. These features include:

- Support for relationships among MDS entities. This includes 1:1, 1:many, and master-detail relationships (bi-directional and many:many relationships are expected in a future release)
- UI support for browsing and restoring items from Trash
- UI support for Filters
- Support for additional types: Locale, Date, File, Map
- Various UI changes to improve the schema editor and data browser

Repository Consolidation

In order to simplify development, some of the MOTECH source code repositories have been merged. Rather than maintaining four separate repositories for MOTECH source code (motech, platform-campaigns, platform-communications, and platform-medical-records), there are now two repositories: motech and modules. Please see the [MOTECH Code Repositories](#) topic for more information.

Modules Moved to motech-contrib

As part of the repository cleanup effort, a few modules were moved out to the motech-contrib repository. The specific modules that were chosen are those that have very few (or in some cases no) consumers. These modules will no longer be maintained by Grameen Foundation, but MOTECH implementers are welcome to continue using them and either fork the code or submit pull requests to the repo as needed. The list of migrated modules is as follows:

- Event Aggregation
- Mobile Forms
- OpenMRS OMOD
- Outbox

New Modules

Several new modules were developed as part of this release:

- **mTraining** - The mTraining module provides data containers and APIs for defining training courses and tracking user enrollment and progress. A typical use case might be an IVR-based training course for a health worker.
- **Batch** - The Batch module is an implementation of Spring batch (version:3.0.0.M3). It essentially deals with scheduling triggering of jobs.
- **Hub** - The Hub module is the implementation of Google's [PubSubHubbub Hub 0.4 Specifications](#). It exposes an API so other modules can act as publisher and make contents available to it for distribution whenever there is an update in a topic.

Legacy IVR Modules Deprecated - Replacements Coming

The following modules have been deprecated:

- Call Flow
- Decision Tree
- IVR (including API, Asterisk, Kookoo, Verboice, and Voxeo)

These modules are still present in the source code repository, but they were not built as part of the release. If you need to use one or more of these modules, you can build each module that you require by executing **mvn install** from the module's base directory. Note that these modules depend on CouchDB.

These legacy modules will be replaced in 0.25 by new generic modules for handling VXML and CCXML. There will also be a re-worked version of the Verboice module in a coming release that removes the dependencies on call-flow and decision-tree.

Known Issues

- **MOTECH-818** - Able to remove a field from a lookup even if the lookup is being used in a Task

Summary: User should not be able to remove a field from an MDS Lookup if the Lookup is used as a data source in a Task. Currently this is not prevented.

Workaround: When modifying a Lookup, the user will need to verify manually that it is not being used as a data source in a Task (this can be checked via the Tasks UI).

- **MOTECH-1084** - MDS ComboBox UI Bugs

Summary: There are some problems with combobox fields when we add two or more of them to an MDS entity:

1. After opening instance view in Data Browser we can see error message "This field is required" under dropboxes of all comboboxes (except for the first one) even though they aren't.
2. After clicking "Add option" button and filling text field with any value on more than one combobox, when we click "Save" for any of them, save buttons for all others will just gray out and turn off. They'll become active again when we enter anything in text fields in any combobox.

Workaround:

1. A number of workarounds may exist depending on the nature of your application. For example, one could create a dummy default option for the ComboBox with a name like "Empty" or "No Value" when defining the ComboBox field.
2. Enter anything in the text fields of any combobox, and the "Save" buttons will become active again.

- **MOTECH-1125** - Getters starting with "is" are not recognized by the MDS annotation processor

Summary: The MDS annotation processor should recognize boolean getters, starting with "is", eg. for field "completed", the getter method "isCompleted()" should be recognized. Currently, it seems to only recognize getters starting with "get".

Workaround: This issue may be temporarily avoided by prefixing getters on MDS entity classes with "get".

- **MOTECH-1147** - Default value for Date type fields doesn't work

Summary: Create an entity and add a Date type field. Set a default value to any date and save changes. Notice that the default value field is clear and when you add an instance of that entity, there's no default value inserted.

Workaround: When using an MDS Entity with a field of type Date, the values for all Date fields will need to be set explicitly.

- **MOTECH-1153** - Creating an MDS entity with an enum field fails if the enum has many values

Summary: Attempting to create an entity that has an enum field (allow user supplied option disabled), that has got many values causes failures. This does not happen when user supplied option is enabled (as it creates a list, instead of enum).

Workaround: Some possible workarounds for this issue (depending on the nature of the application) include: - Enabling user supplied values on enum fields, if appropriate for the application - If possible, splitting the enum into multiple enums

- [MOTECH-1156](#) - Error when adding MDS Entity with space in name

Summary: If a user adds an Entity with spaces in the name then there is an error. After that it is impossible to add other Entities. Entity name should be validated against spaces in the name or they should be deleted.

Workaround: Avoid spaces in Entity names.

Tickets

You can browse the list of tickets resolved for this release on our [issue tracker](#).

Roadmap

This page describes the high-level roadmap for the next few MOTECH Platform releases. The specific features that comprise the releases listed below may be rescheduled as additional information comes to light. For more granular and up-to-date information about release plans, click on the issue tracker links for each of the releases below.

9.1 Version 0.25 - Fall 2014

[Issue Tracker](#)

9.1.1 MOTECH Data Services Performance

The goal of this effort will be to ensure that MDS performance is as good as (or better than) MOTECH running on CouchDB. We will begin by conducting performance testing and benchmarking of MDS against CouchDB to find performance bottlenecks. These will be prioritized, and the most important ones will be fixed for this release (others may be postponed to future releases).

9.1.2 MOTECH Scale Testing

A test environment will be created to simulate MOTECH running at scale. Once this environment is created, we will test performance of MOTECH under various configurations including clustering. Through this process, we expect to identify, document, and/or fix specific issues discovered, including making clustered mode operational and performant, as well as providing configuration recommendations for ActiveMQ at scale.

9.1.3 IVR Support

The legacy MOTECH IVR modules were deprecated in the 0.24 release. Starting with version 0.25, there will be two IVR modules - one for VXML/CCXML, and one for Verboice.

9.1.4 MDS REST API Generation

MOTECH will automatically generate REST APIs for CRUD operations on entities defined in MOTECH Data Services.

9.2 Version 0.26 - Late 2014

[Issue Tracker](#)

9.2.1 DHIS2 Module

A new module will be created to support pushing individual level anonymous data (DHIS2 Event Capture) to DHIS2. DHIS2 data push will be exposed as a new Action through MOTECH Tasks. Support for additional DHIS2 use cases will likely come in future releases.

9.2.2 ETL

We will provide a demo and documentation illustrating how to export MOTECH data using one or more popular ETL tools to a star schema for reporting using Pentaho, Jasper or other BI toolkit.

9.3 Version 1.0 - 2015

[Issue Tracker](#)

9.3.1 Stable Semantic Versioning

We will apply a semantic version scheme to MOTECH Platform and all modules thus making it easier to determine backward compatibility. We will stabilize our core system and API providing implementers with a level of confidence that their system will be compatible with future releases of MOTECH.

9.3.2 End User Install

End users should be able to install MOTECH without compiling. Install should be scriptable and unattended. We will provide an example install script that operations engineers may reuse or modify as desired for their purposes.

9.3.3 Platform API Documentation

Every public API will be documented with standard javadoc that is published with each release.

9.3.4 API Sanitization

The process of cleaning up MOTECH's public API - which was started in v0.24 - will be completed, resulting in a stable public API for MOTECH.

MOTECH Mailing Lists

The mailing lists below are the best way to keep in touch with us. Please join the discussion!

10.1 MOTECH Developers

Mailing list for regular contributors to the MOTECH Platform source repository - used for design discussions and other issues impacting the developer community.

motech-dev@googlegroups.com	Join Dev List
--	-------------------------------

10.2 MOTECH Implementers

Mailing list for implementers and other users of MOTECH - used mostly for communication related to releases. Traffic is moderated and very low volume.

motech-implementers@googlegroups.com	Join Implementers List
--	--

11.1 org.motechproject.admin.domain

11.1.1 AdminSettings

public class **AdminSettings**

AdminSettings class is used to store settings for Admin module and specifies that these settings are read-only (by checking config source from bootstrap)

Constructors

AdminSettings

public **AdminSettings** (*List<Settings> settingsList*, boolean *readOnly*)

Methods

getSettingsList

public *List<Settings>* **getSettingsList** ()

isReadOnly

public boolean **isReadOnly** ()

setReadOnly

public void **setReadOnly** (boolean *readOnly*)

setSettingsList

public void **setSettingsList** (*List<Settings> settingsList*)

11.1.2 NotificationRule

public class **NotificationRule**

A notification rule persisted in the database. Represents a rule for sending out a single notification. Contains information about this notification's recipient and the `ActionType` representing a method used for notifying the recipient.

Constructors

NotificationRule

public **NotificationRule** ()

NotificationRule

public **NotificationRule** (*String* recipient, *ActionType* actionType, *Level* level, *String* moduleName)

Methods

getActionType

public *ActionType* **getActionType** ()

getId

public *Long* **getId** ()

getLevel

public *Level* **getLevel** ()

getModuleName

public *String* **getModuleName** ()

getRecipient

public *String* **getRecipient** ()

matches

public boolean **matches** (*StatusMessage* message)

setActionType

public void **setActionType** (*ActionType* actionType)

setId

public void **setId** (*Long id*)

setLevel

public void **setLevel** (*Level level*)

setModuleName

public void **setModuleName** (*String moduleName*)

setRecipient

public void **setRecipient** (*String recipient*)

11.1.3 QueueMBean

public class **QueueMBean**

Represents a JMS queue. Holds information about the queue statistics.

Constructors**QueueMBean**

public **QueueMBean** (*String destination*)

Methods**getConsumerCount**

public long **getConsumerCount** ()

getDequeueCount

public long **getDequeueCount** ()

getDestination

public *String* **getDestination** ()

getEnqueueCount

public long **getEnqueueCount** ()

getExpiredCount

public long **getExpiredCount** ()

getInflightCount

public long **getInflightCount** ()

getQueueSize

public long **getQueueSize** ()

setConsumerCount

public void **setConsumerCount** (long *consumerCount*)

setDequeueCount

public void **setDequeueCount** (long *dequeueCount*)

setDestination

public void **setDestination** ([String](#) *destination*)

setEnqueueCount

public void **setEnqueueCount** (long *enqueueCount*)

setExpiredCount

public void **setExpiredCount** (long *expiredCount*)

setInflightCount

public void **setInflightCount** (long *inflightCount*)

setQueueSize

public void **setQueueSize** (long *queueSize*)

11.1.4 QueueMessage

public class **QueueMessage**

Represents a message from the JMS queue.

Constructors

QueueMessage

```
public QueueMessage (String messageId, Boolean redelivered, Date timestamp)
```

Methods

getMessageId

```
public String getMessageId ()
```

getRedelivered

```
public Boolean getRedelivered ()
```

getTimestamp

```
public String getTimestamp ()
```

11.1.5 StatusMessage

```
public class StatusMessage
```

Represents a message displayed in the ‘messages’ section of the Admin UI. Persisted by MDS module. Apart from the message and its `Level`, contains also information about the module that sent the message. The timeout field represents the `DateTime` of the message expiration.

Constructors

StatusMessage

```
public StatusMessage ()
```

StatusMessage

```
public StatusMessage (String text, String moduleName, Level level)
```

StatusMessage

```
public StatusMessage (String text, String moduleName, Level level, DateTime timeout)
```

Methods

getDate

```
public DateTime getDate ()
```

getLevel

public [Level](#) **getLevel** ()

getModuleName

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

getText

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

getTimeout

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

setDate

public void **setDate** ([DateTime](#) *date*)

setLevel

public void **setLevel** ([Level](#) *level*)

setModuleName

public void **setModuleName** ([String](#) *moduleName*)

setText

public void **setText** ([String](#) *text*)

setTimeout

public void **setTimeout** ([DateTime](#) *timeout*)

11.2 org.motechproject.admin.messages

11.2.1 ActionType

public enum **ActionType**

Represents an action which should be taken for a notification, in particular which message channel should be used to communicate the notification.

Enum Constants

EMAIL

public static final [ActionType](#) **EMAIL**

SMS

public static final [ActionType](#) **SMS**

11.2.2 Level

public enum **Level**

Represents the level of a `org.motechproject.admin.domain.StatusMessage`, which is reflected on the UI. Posting a `org.motechproject.admin.domain.StatusMessage` with a **CRITICAL** level will trigger notifications.

See also: `org.motechproject.admin.domain.StatusMessage`

Enum Constants

CRITICAL

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

DEBUG

public static final [Level](#) **DEBUG**

ERROR

public static final [Level](#) **ERROR**

INFO

public static final [Level](#) **INFO**

WARN

public static final [Level](#) **WARN**

11.3 org.motechproject.admin.service

11.3.1 NotificationRulesDataService

public interface **NotificationRulesDataService** extends [MotechDataService](#)<[NotificationRule](#)>
MDS data service for [NotificationRules](#).

11.3.2 StatusMessageService

public interface **StatusMessageService**

Methods

critical

void **critical** ([String](#) *text*, [String](#) *moduleName*)

critical

void **critical** ([String](#) *text*, [String](#) *moduleName*, [DateTime](#) *timeout*)

debug

void **debug** ([String](#) *text*, [String](#) *moduleName*)

debug

void **debug** ([String](#) *text*, [String](#) *moduleName*, [DateTime](#) *timeout*)

error

void **error** ([String](#) *text*, [String](#) *moduleName*)

error

void **error** ([String](#) *text*, [String](#) *moduleName*, [DateTime](#) *timeout*)

getActiveMessages

[List](#)<[StatusMessage](#)> **getActiveMessages** ()

getAllMessages

[List](#)<[StatusMessage](#)> **getAllMessages** ()

getNotificationRules

`List<NotificationRule> getNotificationRules ()`

info

`void info (String text, String moduleName)`

info

`void info (String text, String moduleName, DateTime timeout)`

postMessage

`void postMessage (StatusMessage message)`

postMessage

`void postMessage (String text, String moduleName, Level level)`

postMessage

`void postMessage (String text, String moduleName, Level level, DateTime timeout)`

removeMessage

`void removeMessage (StatusMessage message)`

removeNotificationRule

`void removeNotificationRule (String id)`

removeNotificationRule

`void removeNotificationRule (Long id)`

saveNotificationRules

`void saveNotificationRules (List<NotificationRule> notificationRules)`

saveRule

`void saveRule (NotificationRule notificationRule)`

warn

void **warn** (*String text*, *String moduleName*)

warn

void **warn** (*String text*, *String moduleName*, *DateTime timeout*)

11.3.3 StatusMessagesDataService

public interface **StatusMessagesDataService** extends *MotechDataService<StatusMessage>*
MDS data service for *StatusMessages*.

Methods

findByTimeout

List<StatusMessage> **findByTimeout** (*Range<DateTime> timeout*)

11.4 org.motechproject.bundle.extender

11.4.1 MotechExtenderConfigFactory

public class **MotechExtenderConfigFactory** implements *FactoryBean<Properties>*

Creates the extender configuration for Motech, currently only blueprint dependency wait time.
In order to change the blueprint extender dependency wait time used during platform runtime,
org.motechproject.blueprint.dependencies.waittime should be set with the wait time in
milliseconds. The default blueprint timeout is 5 minutes.

Fields

DEP_WAIT_TIME_ENV

public static final *String* **DEP_WAIT_TIME_ENV**

DEP_WAIT_TIME_KEY

public static final *String* **DEP_WAIT_TIME_KEY**

Methods

getObject

public *Properties* **getObject** ()

getObjectType

```
public Class<?> getObjectType ()
```

isSingleton

```
public boolean isSingleton ()
```

11.4.2 MotechOsgiApplicationContextCreator

```
public class MotechOsgiApplicationContextCreator implements OsgiApplicationContextCreator
```

Methods**createApplicationContext**

```
public DelegatedExecutionOsgiBundleApplicationContext createApplicationContext (BundleContext  
bundleCon-  
text)
```

setConfigurationScanner

```
public void setConfigurationScanner (ConfigurationScanner configurationScanner)
```

11.4.3 MotechOsgiConfigurableApplicationContext

```
public class MotechOsgiConfigurableApplicationContext extends OsgiBundleXmlApplicationContext implements Conf
```

Constructors**MotechOsgiConfigurableApplicationContext**

```
public MotechOsgiConfigurableApplicationContext (String[] configurationLocations)
```

Methods**getNamespace**

```
public String getNamespace ()
```

getServletConfig

```
public ServletConfig getServletConfig ()
```

getServletContext

public [ServletContext](#) **getServletContext** ()

setConfigLocation

public void **setConfigLocation** ([String](#) *configLocation*)

setNamespace

public void **setNamespace** ([String](#) *namespace*)

setServletConfig

public void **setServletConfig** ([ServletConfig](#) *servletConfig*)

setServletContext

public void **setServletContext** ([ServletContext](#) *servletContext*)

waitForContext

public void **waitForContext** (int *waitTimeInMillis*)

11.5 org.motechproject.commons.api

11.5.1 AbstractDataProvider

public abstract class **AbstractDataProvider** extends [MotechObject](#) implements [DataProvider](#)

Methods

getBody

protected [String](#) **getBody** ()

getClassForType

protected [Class](#)<?> **getClassForType** ([String](#) *type*)

getPackageRoot

public abstract [String](#) **getPackageRoot** ()

getSupportClasses

public abstract `List<Class<?>>` **getSupportClasses** ()

isAssignable

protected boolean **isAssignable** (`Class<?>` *check*, `List<Class<?>>` *classes*)

setBody

protected void **setBody** (`String` *body*)

setBody

protected void **setBody** (`Resource` *resource*)

supports

public boolean **supports** (`String` *type*)

toJSON

public `String` **toJSON** ()

11.5.2 ApplicationContextServiceReferenceUtils

public final class **ApplicationContextServiceReferenceUtils**

Fields

SERVICE_NAME

public static final `String` **SERVICE_NAME**

Methods

isValid

public static boolean **isValid** (`ServiceReference` *serviceReference*)

isNotValid

public static boolean **isNotValid** (`ServiceReference` *serviceReference*)

11.5.3 CastUtils

public final class **CastUtils**

Methods

cast

public static <T> List<T> **cast** (Class<T> *clazz*, Enumeration *enumeration*)

11.5.4 DataProvider

public interface **DataProvider**

Methods

getName

String **getName** ()

lookup

Object **lookup** (String *type*, String *lookupName*, Map<String, String> *lookupFields*)

supports

boolean **supports** (String *type*)

toJSON

String **toJSON** ()

11.5.5 IdentityProvider

public interface **IdentityProvider**

Methods

getIdentity

String **getIdentity** ()

11.5.6 MotechEnumUtils

public final class **MotechEnumUtils**
Misc enum-related helper functions

Methods

toEnumSet

public static <T extends Enum> **Set**<T> **toEnumSet** (**Class**<T> *enumClass*, **Set**<**String**> *strings*)

Returns a set of enums given a set of strings and an enum class

Parameters

- **enumClass** – the enum class
- **strings** – a set of strings

Returns the enum set constructed from the given string set

toEnumSet

public static <T extends Enum> **Set**<T> **toEnumSet** (**Class**<T> *enumClass*, **String** *csv*)

Returns a set of enums given a comma separated string and an enum class

Parameters

- **enumClass** – the enum class
- **csv** – a comma separated string representing a set of enum values

Returns the enum set constructed from the given string

toString

public static **String** **toString** (**Set**<? extends **Enum**> *items*)

Returns a csv string given a set of enums

Parameters

- **items** – a set of enums

Returns the csv string constructed from the given enum set

toStringSet

public static **Set**<**String**> **toStringSet** (**Set**<? extends **Enum**> *items*)

Returns a set of strings given a set of enums

Parameters

- **items** – a set of enums

Returns the string set constructed from the given enum set

11.5.7 MotechException

public class **MotechException** extends **RuntimeException**

Constructors

MotechException

```
public MotechException (String s, Throwable throwable)
```

MotechException

```
public MotechException (String message)
```

11.5.8 MotechMapUtils

```
public final class MotechMapUtils
```

The `MotechMapUtils` class contains methods that allow modifications and operations on maps

Methods

asProperties

```
public static Properties asProperties (Map<Object, Object> map)
```

Converts `java.util.Map` into `java.util.Properties`

Parameters

- **map** – Map to convert

Returns Properties, created from given map

mergeMaps

```
public static Map<Object, Object> mergeMaps (Map<Object, Object> overridingMap, Map<Object, Object>  
baseMap)
```

Null-safe merge of two maps. If both parameters are null it returns empty map. If one of the maps is null, it returns the other one. If a key is present in two maps, the value in the merged map will be taken from the overriding map.

Parameters

- **overridingMap** – The map overriding values in the base map
- **baseMap** – The base map

Returns merged map

11.5.9 MotechObject

```
public class MotechObject
```

LayerSupertype for entire motech project. Used optionally through the motech project.

Methods

assertArgumentNotEmpty

protected void **assertArgumentNotEmpty** (*String* *objectName*, *String* *argument*)

assertArgumentNotNull

protected void **assertArgumentNotNull** (*String* *objectName*, *Object* *object*)

logError

protected void **logError** (*String* *templateMessage*, *Object...* *params*)

logError

protected void **logError** (*String* *message*)

logError

protected void **logError** (*String* *message*, *Exception* *e*)

logInfo

protected void **logInfo** (*String* *templateMessage*, *Object...* *params*)

logger

protected *Logger* **logger** ()

11.5.10 NanoStopWatch

public class **NanoStopWatch**

Constructors

NanoStopWatch

public **NanoStopWatch** ()

Methods

duration

public long **duration** ()

start

```
public NanoStopWatch start ()
```

11.5.11 Range

```
public class Range<T>
```

Constructors

Range

```
public Range (T min, T max)
```

Methods

equals

```
public boolean equals (Object o)
```

getMax

```
public T getMax ()
```

getMin

```
public T getMin ()
```

hashCode

```
public int hashCode ()
```

11.5.12 SystemIdentityProvider

```
public class SystemIdentityProvider implements IdentityProvider
```

Methods

getIdentity

```
public String getIdentity ()
```

11.5.13 Tenant

```
public class Tenant
```

Constructors

Tenant

Tenant (*TenantIdentity identity*)

Methods

canHaveQueue

public boolean **canHaveQueue** (*String queueName*)

current

public static *Tenant* **current** ()

getId

public *String* **getId** ()

getSuffixedId

public *String* **getSuffixedId** ()

11.5.14 TenantIdentity

public class **TenantIdentity**

Constructors

TenantIdentity

public **TenantIdentity** ()

TenantIdentity

public **TenantIdentity** (*IdentityProvider identityProvider*)

Methods

getId

public *String* **getId** ()

11.6 org.motechproject.commons.api.json

11.6.1 MotechJsonReader

public class **MotechJsonReader**

Constructors

MotechJsonReader

public **MotechJsonReader** ()

MotechJsonReader

public **MotechJsonReader** (FieldNamingStrategy *fieldNamingStrategy*)

Methods

readFromFile

public Object **readFromFile** (String *classpathFile*, Type *ofType*)

readFromStream

public Object **readFromStream** (InputStream *stream*, Type *ofType*)

readFromStreamOnlyExposeAnnotations

public Object **readFromStreamOnlyExposeAnnotations** (InputStream *stream*, Type *ofType*)

readFromString

public Object **readFromString** (String *text*, Type *ofType*)

readFromString

public Object **readFromString** (String *text*, Type *ofType*, Map<Type, Object> *typeAdapters*)

readFromStringOnlyExposeAnnotations

public Object **readFromStringOnlyExposeAnnotations** (String *text*, Type *ofType*)

11.7 org.motechproject.commons.api.model

11.7.1 ExtensibleDataObject

public abstract class **ExtensibleDataObject**<T>

Methods

addData

public T **addData** (*String key*, *Object value*)

addData

public T **addData** (*Map<String, Object> data*)

getData

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

11.7.2 MotechProperties

public class **MotechProperties** extends *HashMap<String, String>*

11.8 org.motechproject.commons.date

11.8.1 ParseException

public class **ParseException** extends *RuntimeException*

Constructors

ParseException

public **ParseException** (*String s*)

11.9 org.motechproject.commons.date.model

11.9.1 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**

11.9.2 Time

public class **Time** implements [Comparable<Time>](#), [Serializable](#)

Constructors

Time

public **Time** ()

Time

public **Time** (int *hour*, int *minute*)

Time

public **Time** (*Integer hour, Integer minute*)

Time

public **Time** (*LocalTime localTime*)

Time

public **Time** (*String timeStr*)

Methods**compareTo**

public int **compareTo** (*Time otherTime*)

equals

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

ge

public boolean **ge** (*Time toCompare*)

getDateTime

public *DateTime* **getDateTime** (*DateTime dateTime*)

getHour

public *Integer* **getHour** ()

getMinute

public *Integer* **getMinute** ()

hashCode

public int **hashCode** ()

isBefore

public boolean **isBefore** (*Time other*)

le

public boolean **le** (*Time toCompare*)

parseTime

public static *Time* **parseTime** (*String time*, *String separator*)

setHour

public void **setHour** (*Integer hour*)

setMinute

public void **setMinute** (*Integer minute*)

timeStr

public *String* **timeStr** ()

toString

public *String* **toString** ()

valueOf

public static *Time* **valueOf** (*String str*)

11.10 org.motechproject.commons.date.util

11.10.1 DateTimeSourceUtil

public final class **DateTimeSourceUtil**

Methods

now

public static *DateTime* **now** ()

setSourceInstance

public static void **setSourceInstance** (*DateTimeSource sourceInstance*)

timeZone

```
public static DateTimeZone timeZone ()
```

today

```
public static LocalDate today ()
```

11.10.2 DateUtil

```
public final class DateUtil
```

Methods**daysPast**

```
public static int daysPast (LocalDate localDate, DayOfWeek dayOfWeek)
```

daysStarting

```
public static List<DayOfWeek> daysStarting (DayOfWeek day, int numberOfDays)
```

daysToCalendarWeekEnd

```
public static int daysToCalendarWeekEnd (LocalDate date, int calendarWeekStartDay)
```

endOfDay

```
public static DateTime endOfDay (Date dateTime)
```

getDifferenceOfDatesInYears

```
public static int getDifferenceOfDatesInYears (Date startDate)
```

greaterThanOrEqualTo

```
public static List<DateTime> greaterThanOrEqualTo (DateTime date, List<DateTime> dates)
```

inRange

```
public static boolean inRange (DateTime reference, DateTime start, DateTime end)
```

isOnOrAfter

```
public static boolean isOnOrAfter (DateTime firstDate, DateTime secondDate)
```

isOnOrBefore

public static boolean **isOnOrBefore** (*DateTime* *firstDate*, *DateTime* *secondDate*)

lessThan

public static List<*DateTime*> **lessThan** (*DateTime* *date*, List<*DateTime*> *dates*)

newDate

public static *LocalDate* **newDate** (int *year*, int *month*, int *day*)

newDate

public static *LocalDate* **newDate** (*Date* *date*)

newDate

public static *LocalDate* **newDate** (*DateTime* *dateTime*)

newDateTime

public static *DateTime* **newDateTime** (*LocalDate* *localDate*, int *hour*, int *minute*, int *second*)

newDateTime

public static *DateTime* **newDateTime** (*Date* *date*)

newDateTime

public static *DateTime* **newDateTime** (*LocalDate* *date*)

newDateTime

public static *DateTime* **newDateTime** (*LocalDate* *localDate*, *Time* *time*)

newDateTime

public static *DateTime* **newDateTime** (int *year*, int *month*, int *day*, *Time* *time*)

newDateTime

public static *DateTime* **newDateTime** (int *year*, int *month*, int *day*)

newDateTime

```
public static DateTime newDateTime (int year, int month, int day, int hour, int minute, int second)
```

nextApplicableWeekDay

```
public static DateTime nextApplicableWeekDay (DateTime fromDate, List<DayOfWeek> applicable-  
Days)
```

nextApplicableWeekDayIncludingFromDate

```
public static DateTime nextApplicableWeekDayIncludingFromDate (DateTime fromDate, List<DayOfWeek> applicableDays)
```

now

```
public static DateTime now ()
```

nowUTC

```
public static DateTime nowUTC ()
```

setTimeZone

```
public static DateTime setTimeZone (DateTime dateTime)
```

setTimeZoneUTC

```
public static DateTime setTimeZoneUTC (DateTime dateTime)
```

time

```
public static Time time (DateTime dateTime)
```

today

```
public static LocalDate today ()
```

tomorrow

```
public static LocalDate tomorrow ()
```

11.10.3 JodaFormatter

public class **JodaFormatter**

Constructors

JodaFormatter

public **JodaFormatter** ()

Methods

formatDateTime

public **String** **formatDateTime** (*DateTime dateTime*)

formatPeriod

public **String** **formatPeriod** (*Period period*)

Format joda period as text, eg: "1 year"

Parameters

- **period** – time interval

parse

public **Period** **parse** (*String intervalString*, *Locale locale*)

Parse time interval in different units, eg: "1 year"

Parameters

- **intervalString** – time interval format number: integer unit : year, month, week, day, hour, minute, second (can use plural forms also) currently compound units like 1 year and 2 months are not supported

parseDateTime

public **DateTime** **parseDateTime** (*String isoDateTime*)

parsePeriod

public **Period** **parsePeriod** (*String intervalString*)

11.10.4 StringUtil

public final class **StringUtil**

Methods

isEmpty

public static boolean **isEmpty** (*String str*)

11.11 org.motechproject.commons.date.util.datetime

11.11.1 DateTimeSource

public interface **DateTimeSource**

Methods

now

DateTime **now** ()

timeZone

DateTimeZone **timeZone** ()

today

LocalDate **today** ()

11.11.2 DefaultDateTimeSource

public class **DefaultDateTimeSource** implements *DateTimeSource*

Constructors

DefaultDateTimeSource

public **DefaultDateTimeSource** ()

Methods

now

public *DateTime* **now** ()

timeZone

public *DateTimeZone* **timeZone** ()

today

public `LocalDate` **today** ()

11.12 org.motechproject.commons.date.valueobjects

11.12.1 WallTime

public class **WallTime** implements `Serializable`

Constructors

WallTime

public **WallTime** (`String` *userReadableForm*)

Methods

asPeriod

public `Period` **asPeriod** ()

equals

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

getHours

public int **getHours** ()

getMinutes

public int **getMinutes** ()

hashCode

public int **hashCode** ()

inDays

public int **inDays** ()

inMillis

```
public long inMillis()
```

inMinutes

```
public int inMinutes()
```

isLessThanADay

```
public boolean isLessThanADay()
```

11.12.2 WallTimeUnit

```
public enum WallTimeUnit
```

Enum Constants

Day

```
public static final WallTimeUnit Day
```

Hour

```
public static final WallTimeUnit Hour
```

Minute

```
public static final WallTimeUnit Minute
```

Week

```
public static final WallTimeUnit Week
```

11.13 org.motechproject.commons.sql.service

11.13.1 SqlDBManager

```
public interface SqlDBManager
```

Classes inheriting this interface are responsible for getting sql properties from the bootstrap configuration.

Methods

getSqlProperties

`Properties` **getSqlProperties** (`Properties` *propertiesToUpdate*)

updateSqlProperties

`void` **updateSqlProperties** ()

11.14 org.motechproject.config.core

11.14.1 MotechConfigurationException

public class **MotechConfigurationException** extends `RuntimeException`

The object of this class is thrown when there is a problem with reading the configuration from the predefined sources.

Constructors

MotechConfigurationException

public **MotechConfigurationException** (`String` *message*, `Exception` *exception*)

Parameters

- **message** – A descriptive message explaining the nature of the problem resulted in exception
- **exception** – Actual exception (if any) that this exception resulted from

MotechConfigurationException

public **MotechConfigurationException** (`String` *message*)

11.15 org.motechproject.config.core.constants

11.15.1 ConfigurationConstants

public final class **ConfigurationConstants**

Provides all the configuration constants.

Fields

AMQ_BROKER_URL

public static final `String` **AMQ_BROKER_URL**

AMQ_CONCURRENT_CONSUMERS

public static final [String](#) **AMQ_CONCURRENT_CONSUMERS**

AMQ_MAX_CONCURRENT_CONSUMERS

public static final [String](#) **AMQ_MAX_CONCURRENT_CONSUMERS**

AMQ_MAX_REDELIVERIES

public static final [String](#) **AMQ_MAX_REDELIVERIES**

AMQ_QUEUE_EVENTS

public static final [String](#) **AMQ_QUEUE_EVENTS**

AMQ_QUEUE_SCHEDULER

public static final [String](#) **AMQ_QUEUE_SCHEDULER**

AMQ_REDELIVERY_DELAY_IN_MILLIS

public static final [String](#) **AMQ_REDELIVERY_DELAY_IN_MILLIS**

BUNDLE_ID

public static final [String](#) **BUNDLE_ID**

BUNDLE_SECTION

public static final [String](#) **BUNDLE_SECTION**

BUNDLE_SETTINGS_CHANGED_EVENT_SUBJECT

public static final [String](#) **BUNDLE_SETTINGS_CHANGED_EVENT_SUBJECT**

BUNDLE_SYMBOLIC_NAME

public static final [String](#) **BUNDLE_SYMBOLIC_NAME**

CONFIG_MODULE_DIR_PREFIX

public static final [String](#) **CONFIG_MODULE_DIR_PREFIX**

EVENT_RELAY_CLASS_NAME

public static final `String` **EVENT_RELAY_CLASS_NAME**

FILE_CHANGED_EVENT_SUBJECT

public static final `String` **FILE_CHANGED_EVENT_SUBJECT**

FILE_CREATED_EVENT_SUBJECT

public static final `String` **FILE_CREATED_EVENT_SUBJECT**

FILE_DELETED_EVENT_SUBJECT

public static final `String` **FILE_DELETED_EVENT_SUBJECT**

FILE_PATH

public static final `String` **FILE_PATH**

JSON_EXTENSION

public static final `String` **JSON_EXTENSION**

LANGUAGE

public static final `String` **LANGUAGE**

LOGINMODE

public static final `String` **LOGINMODE**

MOTECH_EVENT_CLASS_NAME

public static final `String` **MOTECH_EVENT_CLASS_NAME**

PLATFORM_SETTINGS_CHANGED_EVENT_SUBJECT

public static final `String` **PLATFORM_SETTINGS_CHANGED_EVENT_SUBJECT**

PROPERTIES_EXTENSION

public static final `String` **PROPERTIES_EXTENSION**

PROVIDER_NAME

```
public static final String PROVIDER_NAME
```

PROVIDER_URL

```
public static final String PROVIDER_URL
```

RAW_DIR

```
public static final String RAW_DIR
```

SERVER_URL

```
public static final String SERVER_URL
```

SETTINGS

```
public static final String SETTINGS
```

SETTINGS_FILE_NAME

```
public static final String SETTINGS_FILE_NAME
```

STATUS_MSG_TIMEOUT

```
public static final String STATUS_MSG_TIMEOUT
```

UPLOAD_SIZE

```
public static final String UPLOAD_SIZE
```

11.16 org.motechproject.config.core.domain

11.16.1 AbstractDBConfig

```
public abstract class AbstractDBConfig
```

This abstract class encapsulates the database configuration, composed of as db url, username and password.

Constructors

AbstractDBConfig

public **AbstractDBConfig** (*String url*, *String driver*, *String username*, *String password*)

Parameters

- **url** –
- **driver** –
- **username** –
- **password** –

Methods

equals

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

getDriver

public *String* **getDriver** ()

getPassword

public *String* **getPassword** ()

getUrl

public *String* **getUrl** ()

getUsername

public *String* **getUsername** ()

hashCode

public int **hashCode** ()

toString

public *String* **toString** ()

11.16.2 BootstrapConfig

public class **BootstrapConfig**

Represents the bootstrap configuration object. It is composed of:

- 1.DBConfig - represents the database related bootstrap object.
- 2.Tenant ID - represents the identifier of the tenant.
- 3.Configuration source - represents the source of configuration (FILE / UI).

Fields

CONFIG_SOURCE

public static final [String](#) **CONFIG_SOURCE**

DEFAULT_TENANT_ID

public static final [String](#) **DEFAULT_TENANT_ID**

OSGI_FRAMEWORK_STORAGE

public static final [String](#) **OSGI_FRAMEWORK_STORAGE**

SQL_DRIVER

public static final [String](#) **SQL_DRIVER**

SQL_PASSWORD

public static final [String](#) **SQL_PASSWORD**

SQL_URL

public static final [String](#) **SQL_URL**

SQL_USER

public static final [String](#) **SQL_USER**

TENANT_ID

public static final [String](#) **TENANT_ID**

Constructors

BootstrapConfig

```
public BootstrapConfig (SQLDBConfig sqlConfig, String tenantId, ConfigSource configSource, String osgiFrameworkStorage)
```

Parameters

- **sqlConfig** –
- **tenantId** –
- **configSource** –
- **osgiFrameworkStorage** –

Throws

- **org.motechproject.config.core.MotechConfigurationException** – if dbConfig is null.

Methods

equals

```
public boolean equals (Object o)
```

getConfigSource

```
public ConfigSource getConfigSource ()
```

getOsgiFrameworkStorage

```
public String getOsgiFrameworkStorage ()
```

getSqlConfig

```
public SQLDBConfig getSqlConfig ()
```

getTenantId

```
public String getTenantId ()
```

hashCode

```
public int hashCode ()
```

toString

```
public String toString ()
```


11.16.3 ConfigLocation

public class **ConfigLocation**

Defines a MOTECH configuration location. If the given location starts with a leading file separator character, the location is treated as a file system directory. Otherwise, it is treated as a classpath location.

Constructors

ConfigLocation

public **ConfigLocation** (*String configLocation*)

Methods

equals

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

getExistingConfigFiles

public *List<File>* **getExistingConfigFiles** ()

getFile

public *File* **getFile** (*String fileName*, *FileAccessType accessType*)

This method Returns the *java.io.File* object for the given file name relative to the config location. It also checks for the requested file accessibility. If the requested access type check is *ConfigLocation.FileAccessType.READABLE*, the file's existence and readability will be checked. Similarly, if the requested access type check is *ConfigLocation.FileAccessType.WRITABLE*, then the write accessibility to the file will be checked. If the file does not exists, write accessibility of its ancestors will be checked.

Parameters

- **fileName** – Name of the file to be added to the config location.
- **accessType** – One of *ConfigLocation.FileAccessType.READABLE* or *ConfigLocation.FileAccessType.WRITABLE*.

Throws

- **MotechConfigurationException** – if the file is not readable or writable depending on the given access type.

Returns File relative to the config location.

getLocation

public *String* **getLocation** ()

`getUrlResource`

`UrlResource` `getUrlResource()`

`hasPlatformConfigurationFile`

public boolean `hasPlatformConfigurationFile()`

`hashCode`

public int `hashCode()`

`toResource`

public `Resource` `toResource()`
Resource corresponding to the config location.
Returns resource

`toString`

public `String` `toString()`

11.16.4 ConfigLocation.FileAccessType

public static enum **FileAccessType**
Defines the access check required.

Enum Constants

READABLE

public static final `ConfigLocation.FileAccessType` **READABLE**

WRITABLE

public static final `ConfigLocation.FileAccessType` **WRITABLE**

11.16.5 ConfigSource

public final class **ConfigSource**
Represents the source from which MOTECH configuration should be read.

Fields

FILE

public static final [ConfigSource](#) **FILE**

UI

public static final [ConfigSource](#) **UI**

Methods

getName

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

isFile

public boolean **isFile** ()

isValid

public static boolean **isValid** ([String](#) *name*)

toString

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

valueOf

public static [ConfigSource](#) **valueOf** ([String](#) *name*)

11.16.6 DBConfig

public class **DBConfig** extends [AbstractDBConfig](#)

DBConfig encapsulates the database configuration, composed of as db url, username and password.

Constructors

DBConfig

public **DBConfig** ([String](#) *url*, [String](#) *username*, [String](#) *password*)

Parameters

- **url** –
- **username** –

- **password** –

Throws

- **org.motechproject.config.core.MotechConfigurationException** – if given url is invalid.

11.16.7 SQLDBConfig

public class **SQLDBConfig** extends [AbstractDBConfig](#)

This class encapsulates the SQL database configuration, composed of as db url, username and password.

Constructors

SQLDBConfig

public **SQLDBConfig** (*String url*, *String driver*, *String username*, *String password*)

Parameters

- **url** –
- **driver** –
- **username** –
- **password** –

Throws

- **org.motechproject.config.core.MotechConfigurationException** – if given url is invalid.

11.17 org.motechproject.config.core.filestore

11.17.1 ConfigLocationFileStore

public class **ConfigLocationFileStore**

Used to read default platform config location(s) from `config-location.properties` and also to save in the file in the default location.

`config-location.properties` file will be loaded according to the behaviour of `org.apache.commons.configuration.PropertiesConfiguration` as specified [here](#).

Fields

CONFIG_LOCATION_PROPERTY_KEY

public static final *String* **CONFIG_LOCATION_PROPERTY_KEY**

Constructors

ConfigLocationFileStore

public **ConfigLocationFileStore** (*PropertiesConfiguration propertiesConfiguration*)

Methods

add

public void **add** (*String location*)

getAll

public *Iterable*<*ConfigLocation*> **getAll** ()

11.17.2 PropertiesReader

public final class **PropertiesReader**

A utility class for loading properties from a file.

Methods

getProperties

public static *Properties* **getProperties** (*File file*)

11.18 org.motechproject.config.core.filters

11.18.1 ConfigFileFilter

public class **ConfigFileFilter** extends *FileFileFilter*
FileFilter implementation to filter configuration files.

Fields

PLATFORM_CORE_CONFIG_FILTER

public static final *FileFileFilter* **PLATFORM_CORE_CONFIG_FILTER**

Methods

accept

public boolean **accept** (*File file*)

isFileSupported

public static boolean **isFileSupported** (*File file*)

isPlatformCoreConfigFile

public static boolean **isPlatformCoreConfigFile** (*File file*)

11.19 org.motechproject.config.core.service

11.19.1 CoreConfigurationService

public interface **CoreConfigurationService**

Loads and saves the core configuration required to start the Motech instance.

Fields

CORE_SETTINGS_CACHE_NAME

String **CORE_SETTINGS_CACHE_NAME**

Methods

addConfigLocation

void **addConfigLocation** (*String location*)

Adds the new config location to the list of existing config locations where configurations are loaded from in the file system.

Parameters

- **location** – config location to add.

Throws

- **FileSystemException** –

evictMotechCoreSettingsCache

void **evictMotechCoreSettingsCache** ()

getConfigLocation

ConfigLocation **getConfigLocation** ()

Returns the config location where all the config files are present.

Returns configLocation.

loadBootstrapConfig

BootstrapConfig **loadBootstrapConfig** ()

Loads the bootstrap configuration.

Returns bootstrap configuration.

saveBootstrapConfig

void **saveBootstrapConfig** ([BootstrapConfig](#) *bootstrapConfig*)

Saves the bootstrap configuration

Parameters

- **bootstrapConfig** – Bootstrap config

11.20 org.motechproject.config.domain

11.20.1 ModulePropertiesRecord

public class **ModulePropertiesRecord**

The `ModulePropertiesRecord` class represents a record of a certain module properties. This class is exposed as an `org.motechproject.mds.annotations.Entity` through Motech Data Services.

See also: `org.motechproject.mds.annotations`

Fields

PROPERTIES_FILE_EXTENSION

public static final [String](#) **PROPERTIES_FILE_EXTENSION**

Constructors

ModulePropertiesRecord

public **ModulePropertiesRecord** ()

ModulePropertiesRecord

public **ModulePropertiesRecord** ([Map](#)<[String](#), [Object](#)> *properties*, [String](#) *module*, [String](#) *version*, [String](#) *bundle*, [String](#) *filename*, boolean *raw*)

ModulePropertiesRecord

public **ModulePropertiesRecord** ([Properties](#) *props*, [String](#) *module*, [String](#) *version*, [String](#) *bundle*, [String](#) *filename*, boolean *raw*)

Methods

buildFrom

public static [ModulePropertiesRecord](#) **buildFrom** ([File](#) *file*)

equals

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

getBundle

public *String* **getBundle** ()

getFilename

public *String* **getFilename** ()

getModule

public *String* **getModule** ()

getProperties

public *Map*<*String*, *Object*> **getProperties** ()

getVersion

public *String* **getVersion** ()

hashCode

public int **hashCode** ()

isRaw

public boolean **isRaw** ()

sameAs

public boolean **sameAs** (*Object dataObject*)

setBundle

public void **setBundle** (*String bundle*)

setFilename

public void **setFilename** (*String filename*)

setModule

```
public void setModule (String module)
```

setProperties

```
public void setProperties (Map<String, Object> properties)
```

setRaw

```
public void setRaw (boolean raw)
```

setVersion

```
public void setVersion (String version)
```

toString

```
public String toString ()
```

11.21 org.motechproject.config.monitor

11.21.1 ConfigFileMonitor

```
public class ConfigFileMonitor implements FileListener
```

The `ConfigFileMonitor` is used to monitor changes in config files and send appropriate events.

Methods**fileChanged**

```
public void fileChanged (FileChangeEvent fileChangeEvent)
```

fileCreated

```
public void fileCreated (FileChangeEvent fileChangeEvent)
```

fileDeleted

```
public void fileDeleted (FileChangeEvent fileChangeEvent)
```

init

```
public void init ()
```

setFileMonitor

public void **setFileMonitor** (DefaultFileMonitor *fileMonitor*)

stop

public void **stop** ()

updateFileMonitor

public void **updateFileMonitor** ()

11.22 org.motechproject.config.service

11.22.1 ConfigurationService

public interface **ConfigurationService**

Central configuration service that monitors and manages configurations.

Fields**SETTINGS_CACHE_NAME**

String **SETTINGS_CACHE_NAME**

Methods**addOrUpdate**

void **addOrUpdate** (File *file*)

Saves both property and raw configurations in FILE mode only. Files are classified as either raw config or properties based on the extension of the file.

Parameters

- **file** – File to read configuration from.

addOrUpdateModuleRecord

void **addOrUpdateModuleRecord** (ModulePropertiesRecord *record*)

A convenient method for adding or updating the properties, which determines on its own whether the record should be added or updated

Parameters

- **record** – a record to store

addOrUpdateModuleRecords

void **addOrUpdateModuleRecords** (*List<ModulePropertiesRecord> records*)

Bulk add or update method for the Module Properties records. Iterates through the passed records and either adds them, if they are not present, or updates otherwise.

Parameters

- **records** – a list of properties records

addOrUpdateProperties

void **addOrUpdateProperties** (*String module, String version, String bundle, String filename, Properties newProperties, Properties defaultProperties*)

Depending on the config source, it will either store properties in the DB or file. Only properties that are different from the default ones are stored. If the properties database record or file doesn't exist yet for the given module, it will be created.

Parameters

- **module** – The module we wish to update properties for
- **version** – Version of updated bundle
- **bundle** – Symbolic name of updated bundle
- **filename** – Resource filename
- **newProperties** – New properties to store
- **defaultProperties** – Default properties of the module

Throws

- **IOException** – if module properties cannot be retrieved from file

createZipWithConfigFiles

FileInputStream **createZipWithConfigFiles** (*String propertyFile, String fileName*)

Uses current configuration and default one to find changed properties and then connects them with annotations. Moreover creates file with non default configurations and packs it into the zip file.

Parameters

- **propertyFile** – name of exported file

Throws

- **IOException** –

Returns *FileInputStream* that contains zip file

delete

void **delete** (*String module*)

Deletes the db record corresponding to the module.

deleteByBundle

void **deleteByBundle** (*String module*)
Deletes the db record corresponding to the module with given bundle symbolic name.

evictMotechSettingsCache

void **evictMotechSettingsCache** ()

getAllModuleProperties

Map<String, Properties> **getAllModuleProperties** (*String module*, Map<String, Properties> *defaultProperties*)
Retrieves all the module properties and returns them as Map, where key is the filename.

Parameters

- **module** – The module we wish to retrieve properties for
- **defaultProperties** – Default properties of the module

Throws

- **IOException** – if any of the module properties file cannot be read

Returns Properties mapped by filename

getConfigSource

ConfigSource **getConfigSource** ()
This method allows to check whether MOTECH is currently running in the FILE or UI mode
Returns Current Config Source

getModuleProperties

Properties **getModuleProperties** (*String module*, *String filename*, Properties *defaultProperties*)
Retrieves merged properties, given default set. Depending on the ConfigSource, it will either merge default properties with the properties from DB or get properties from file.

Parameters

- **module** – The module we wish to retrieve properties for
- **filename** – Resource filename
- **defaultProperties** – Default properties of the module

Throws

- **IOException** – if module properties cannot be read from file

Returns Merged properties of the certain module

getPlatformSettings

`MotechSettings` **getPlatformSettings** ()

getRawConfig

`InputStream` **getRawConfig** (`String` *module*, `String` *filename*, `Resource` *resource*)

Allows to retrieve raw JSON data either from the database or file, depending on the specified ConfigSource mode.

Parameters

- **module** – Module we wish to retrieve raw data for
- **filename** – Resource filename
- **resource** – Resource file containing default rawConfig, in case no other has been found

Throws

- **IOException** –

Returns Raw JSON data as `InputStream`

listRawConfigNames

`List<String>` **listRawConfigNames** (`String` *module*)

Depending on the selected ConfigSource mode, this method looks for all registered raw data properties within the specified module.

Parameters

- **module** – Module we wish to perform look for

Returns List of filenames that register raw config for specified module

loadBootstrapConfig

`BootstrapConfig` **loadBootstrapConfig** ()

Loads bootstrap config that is used to start up the Motech server.

The bootstrap configuration is loaded in the following order:

1. Load the configuration from `bootstrap.properties` from the config directory specified by the environment variable `MOTech_CONFIG_DIR`. `bootstrap.properties` contains the following properties:

```
sql.url (Mandatory)
sql.username (If required)
sql.password (If required)
tenant.id (Optional. Defaults to 'DEFAULT')
config.source (Optional. Defaults to 'UI')
```

An example `bootstrap.properties` is given below:

```
sql.url=jdbc:mysql://localhost:3306/  
sql.username=motech  
sql.password=motech  
tenant.id=MotherChildCare  
config.source=FILE
```

2.If MOTECH_CONFIG_DIR environment variable is **not** set, load the specific configuration values from the following environment variables:

```
MOTECH_SQL_URL (Mandatory)  
MOTECH_SQL_USERNAME (If required)  
MOTECH_SQL_PASSWORD (If required)  
MOTECH_TENANT_ID (Optional. Defaults to 'DEFAULT')  
MOTECH_CONFIG_SOURCE (Optional. Defaults to 'UI')
```

3.If MOTECH_DB_URL environment is not set, load the configuration from `bootstrap.properties` from the default MOTECH config directory specified in the file `config-locations.properties`.

Throws

- [org.motechproject.config.core.MotechConfigurationException](#) – if bootstrap configuration cannot be loaded.

Returns Bootstrap configuration

loadConfig

`SettingsRecord loadConfig()`

Loads current MOTECH configuration

Returns current MOTECH settings

loadDefaultConfig

`SettingsRecord loadDefaultConfig()`

Loads the default config for MOTECH from the resource file.

Returns default settings

processExistingConfigs

`void processExistingConfigs(List<File> files)`

Adds, updates, or deletes configurations in FILE mode only. Files are classified as either raw config or properties based on the extension of the file.s

Parameters

- **files** – Files to read configuration from.

rawConfigExists

`boolean rawConfigExists(String module, String filename)`

Allows to check if raw data has been registered for specified module

Parameters

- **module** – Module symbolic name
- **filename** – Resource filename

Returns True if raw data exists for given parameters, false otherwise

registersProperties

boolean **registersProperties** (*String module*, *String filename*)

Checks if given module registers certain property file

Parameters

- **module** – Module we wish to perform check for
- **filename** – Resource filename

Returns True if properties exist, false otherwise

removeModuleRecords

void **removeModuleRecords** (*List<ModulePropertiesRecord> records*)

Removes given module properties records

Parameters

- **records** – a list of properties records to remove

removeProperties

void **removeProperties** (*String module*, *String filename*)

Removes properties for given module from database or file.

Parameters

- **module** – The module we wish to remove properties for
- **filename** – Resource filename

requiresConfigurationFiles

boolean **requiresConfigurationFiles** ()

Checks whether set MOTECH configuration requires the configuraton files to be present

Returns true if files are required, false otherwise

retrieveRegisteredBundleNames

List<String> **retrieveRegisteredBundleNames** ()

Depending on the selected ConfigSource mode, this method looks for registered bundle properties and returns a list of files it has found

Returns List of files with registered properties

save

void **save** ([BootstrapConfig](#) *bootstrapConfig*)

Saves the given `BootstrapConfig` in the `bootstrap.properties` file located in default MOTECH config location. The default motech config location is specified in the file `config-locations.properties`.

Parameters

- **bootstrapConfig** – Bootstrap configuration.

Throws

- [org.motechproject.config.core.MotechConfigurationException](#) – if bootstrap configuration cannot be saved.

savePlatformSettings

void **savePlatformSettings** ([Properties](#) *settings*)

savePlatformSettings

void **savePlatformSettings** ([MotechSettings](#) *settings*)

saveRawConfig

void **saveRawConfig** ([String](#) *module*, [String](#) *version*, [String](#) *bundle*, [String](#) *filename*, [InputStream](#) *rawData*)

Allows persisting of raw json properties either in the database or file, depending on the selected `ConfigSource` mode.

Parameters

- **module** – Module we wish to save properties for
- **filename** – Resource filename
- **rawData** – Raw JSON data to persist

Throws

- [IOException](#) –

setPlatformSetting

void **setPlatformSetting** ([String](#) *key*, [String](#) *value*)

updateConfigLocation

void **updateConfigLocation** ([String](#) *newConfigLocation*)

Adds a new config location and restarts the monitor.

Parameters

- **newConfigLocation** – New config location

updatePropertiesAfterReinstallation

```
void updatePropertiesAfterReinstallation (String module, String version, String bundle, String
                                         filename, Properties defaultProperties, Properties
                                         newProperties)
```

Works similar to `addOrUpdateProperties` but instead of just adding / updating properties checks database for any deprecated properties and removes to ensure that only current ones are available

Parameters

- **module** – The module we wish to update properties for
- **version** – Version of updated bundle
- **bundle** – Symbolic name of updated bundle
- **filename** – Resource filename
- **newProperties** – New properties to store
- **defaultProperties** – Default properties of the module

Throws

- **IOException** – if module properties cannot be retrieved from file

11.22.2 ModulePropertiesService

public interface **ModulePropertiesService** extends `MotechDataService<ModulePropertiesRecord>`

This service provides data access for `org.motechproject.config.domain.ModulePropertiesRecord`.

The implementation is generated by Motech Data Services and published as an OSGi service.

Methods

findByBundle

```
List<ModulePropertiesRecord> findByBundle (String bundle)
```

findByModule

```
List<ModulePropertiesRecord> findByModule (String module)
```

findByModuleAndFileName

```
List<ModulePropertiesRecord> findByModuleAndFileName (String module, String filename)
```

11.23 org.motechproject.email.builder

11.23.1 EmailRecordSearchCriteria

```
public class EmailRecordSearchCriteria
```

The `EmailRecordSearchCriteria` class represents search criteria that may be used for searching `org.motechproject.email.domain.EmailRecord` entities in Motech

Data Services. A consumer of this class may create search criteria to query on multiple fields by calling several of the with* methods. To perform the search, use `org.motechproject.email.service.EmailAuditService.findEmailRecords (EmailRecordSearchCr`

Methods

getDeliveryStatuses

public `Set<DeliveryStatus>` **getDeliveryStatuses** ()

Gets the delivery statuses criterion.

Returns the delivery statuses criterion for this search criteria

getDeliveryTimeRange

public `Range<DateTime>` **getDeliveryTimeRange** ()

Gets the delivery time range criterion.

Returns the deliveryTimeRange criterion for this search criteria

getFromAddress

public `String` **getFromAddress** ()

Gets the from address criterion.

Returns the fromAddress criterion for this search criteria

getMessage

public `String` **getMessage** ()

Gets the message body criterion.

Returns the message criterion for this search criteria

getQueryParams

public `QueryParams` **getQueryParams** ()

Gets the query paramaters that are used for controlling order and size of the query results for this search criteria.

Returns the query params for this search criteria

getSubject

public `String` **getSubject** ()

Gets the subject criterion.

Returns the subject criterion for this search criteria

getToAddress

public **String** **getToAddress** ()
Gets the to address criterion.

Returns the toAddress criterion for this search criteria

withDeliveryStatuses

public **EmailRecordSearchCriteria** **withDeliveryStatuses** (**Set**<**DeliveryStatus**> *deliveryStatuses*)
Sets the delivery statuses criterion to the set specified

Parameters

- **deliveryStatuses** – the delivery statuses on which to search

Returns this **EmailRecordSearchCriteria** with its **deliveryStatuses** criterion set to the provided statuses

withDeliveryStatuses

public **EmailRecordSearchCriteria** **withDeliveryStatuses** (**DeliveryStatus**... *deliveryStatuses*)
Sets the delivery statuses criterion to the set specified

Parameters

- **deliveryStatuses** – the delivery statuses on which to search

Returns this **EmailRecordSearchCriteria** with its **deliveryStatuses** criterion set to the provided statuses

withFromAddress

public **EmailRecordSearchCriteria** **withFromAddress** (**String** *fromAddress*)
Sets the fromAddress criterion to the address specified

Parameters

- **fromAddress** – the sender email address on which to search

Returns this **EmailRecordSearchCriteria** with its **fromAddress** criterion set to the provided address

withMessage

public **EmailRecordSearchCriteria** **withMessage** (**String** *message*)
Sets the message criterion to the message specified

Parameters

- **message** – the email message body on which to search

Returns this **EmailRecordSearchCriteria** with its **message** criterion set to the provided message

withMessageTime

public `EmailRecordSearchCriteria` **withMessageTime** (`DateTime` *deliveryTimeRange*)

Sets the send time criterion to the time specified. Use this method to search on a specific date/time; if a range is needed, use `withMessageTimeRange` instead.

Parameters

- **deliveryTimeRange** – the specific time on which to search

Returns this `EmailRecordSearchCriteria` with its `deliveryTimeRange` criterion set to the specified date/time

withMessageTimeRange

public `EmailRecordSearchCriteria` **withMessageTimeRange** (`Range<DateTime>` *deliveryTimeRange*)

Sets the sent time criterion to the range specified. Use this method to search on a time range; if searching on a specific date/time is needed, use `withMessageTime` instead.

Parameters

- **deliveryTimeRange** – the date/time range on which to search

Returns this `EmailRecordSearchCriteria` with its `deliveryTimeRange` criterion set to the specified date/time range

withQueryParams

public `EmailRecordSearchCriteria` **withQueryParams** (`QueryParams` *queryParams*)

Sets the `queryParams` of the search criteria to the parameters specified. Use this method when it is necessary to specify order and size of query results. This is used mainly for paging/ordering queries from the UI.

Parameters

- **queryParams** – the query parameters to include with the search criteria

Returns this `EmailRecordSearchCriteria` with its `queryParams` set to the provided parameters

withSubject

public `EmailRecordSearchCriteria` **withSubject** (`String` *subject*)

Sets the subject criterion to the subject specified

Parameters

- **subject** – the subject on which to search

Returns this `EmailRecordSearchCriteria` with its subject criterion set to the provided subject

withToAddress

public `EmailRecordSearchCriteria` **withToAddress** (`String` *toAddress*)

Sets the `toAddress` criterion to the address specified

Parameters

- **toAddress** – the recipient email address on which to search

Returns this `EmailRecordSearchCriteria` with its `toAddress` criterion set to the provided address

11.24 org.motechproject.email.contract

11.24.1 Mail

public class **Mail**

The `Mail` class represents an email message.

Constructors**Mail**

public **Mail** (*String fromAddress*, *String toAddress*, *String subject*, *String message*)

Creates a new instance of `Mail`, with all fields set to the values specified in the parameters.

Parameters

- **fromAddress** – the email address of the sender
- **toAddress** – the email address of the recipient
- **subject** – the subject of the email
- **message** – the body of the email

Methods**equals**

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

Indicates whether some other object is “equal to” this one. Returns true if this `Mail` and the object to compare have reference equality or their field values are all equal.

Parameters

- **obj** – The reference object with which to compare

Returns true if this object is the same as the `obj` argument; false otherwise.

getFromAddress

public *String* **getFromAddress** ()

Gets the email address of the sender.

Returns the sender of the message

`getMessage`

public `String` **getMessage** ()

Gets the message body.

Returns the body of the message

`getSubject`

public `String` **getSubject** ()

Gets the message subject.

Returns the subject of the message

`getText`

public `String` **getText** ()

`getToAddress`

public `String` **getToAddress** ()

Gets the email address of the recipient.

Returns the recipient of the message

`hashCode`

public int **hashCode** ()

Returns a hash code value for this `Mail` object.

Returns a hash code value for this `Mail` object

`toString`

public `String` **toString** ()

Returns a string representation of this `Mail` object.

Returns a string representation of this `Mail` object

11.25 org.motechproject.email.domain

11.25.1 DeliveryStatus

public enum **DeliveryStatus**

The `DeliveryStatus` Enum contains the possible delivery states for an email message.

Enum Constants

ERROR

public static final `DeliveryStatus` **ERROR**
There was an error sending the message.

RECEIVED

public static final `DeliveryStatus` **RECEIVED**
The message was received.

SENT

public static final `DeliveryStatus` **SENT**
The message was sent.

11.25.2 EmailRecord

public class **EmailRecord**
The `EmailRecord` class represents a record of a sent Email. This class is exposed as an `org.motechproject.mds.annotations.Entity` through Motech Data Services.
See also: `org.motechproject.mds.annotations`

Constructors

EmailRecord

public **EmailRecord**()
Creates a new instance of `EmailRecord`, with all fields set to null.

EmailRecord

public **EmailRecord**(*String* fromAddress, *String* toAddress, *String* subject, *String* message, *DateTime* deliveryTime, *DeliveryStatus* deliveryStatus)
Creates a new instance of `EmailRecord`, with all fields set to the values specified in the parameters.

Parameters

- **fromAddress** – the email address of the sender
- **toAddress** – the email address of the recipient
- **subject** – the subject of the email
- **message** – the body of the email
- **deliveryTime** – the date and time that the email was sent
- **deliveryStatus** – the delivery status of the email

Methods

equals

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

Indicates whether some other object is “equal to” this one. Returns true if this EmailRecord and the object to compare have reference equality or their field values are all equal.

Parameters

- **obj** – The reference object with which to compare.

Returns true if this object is the same as the obj argument; false otherwise.

getDeliveryStatus

public *DeliveryStatus* **getDeliveryStatus** ()

Gets the delivery status.

Returns the delivery status of the message

getDeliveryTime

public *DateTime* **getDeliveryTime** ()

Gets the delivery time.

Returns the time that the email was sent

getFromAddress

public *String* **getFromAddress** ()

Gets the email address of the sender.

Returns the sender of the message

getId

public *Long* **getId** ()

getMessage

public *String* **getMessage** ()

Gets the message body.

Returns the body of the message

getSubject

public *String* **getSubject** ()

Gets the message subject.

Returns the subject of the message

getToAddress

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

Gets the email address of the recipient.

Returns the recipient of the message

hashCode

public int **hashCode** ()

Returns a hash code value for this `EmailRecord` object.

Returns a hash code value for this `EmailRecord` object

setFromAddress

public void **setFromAddress** ([String](#) *fromAddress*)

Sets the email address of the sender.

Parameters

- **fromAddress** – the sender of the message

setId

public void **setId** ([Long](#) *id*)

setMessage

public void **setMessage** ([String](#) *message*)

Sets the message body.

Parameters

- **message** – the body of the message

setSubject

public void **setSubject** ([String](#) *subject*)

Sets the message subject.

Parameters

- **subject** – the subject of the message

setToAddress

public void **setToAddress** ([String](#) *toAddress*)

Sets the email address of the recipient.

Parameters

- **toAddress** – the recipient of the message

toString

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

Returns a string representation of this `EmailRecord` object.

Returns a string representation of this `EmailRecord` object

11.25.3 EmailRecordComparator

public class **EmailRecordComparator** implements `Comparator<EmailRecord>`

The `EmailRecordComparator` class is an implementation of the `Comparator` interface, that allows callers to compare `org.motechproject.email.domain.EmailRecord` objects by a single field.

Constructors

EmailRecordComparator

public **EmailRecordComparator** (*Boolean ascending*, *String compareField*)

Creates a new `EmailRecordComparator` that supports comparison based on the specified field.

Parameters

- **ascending** – boolean indicating whether comparisons should be ascending or descending
- **compareField** – the field for which comparisons should be performed

Methods

compare

public int **compare** (*EmailRecord o1*, *EmailRecord o2*)

Compares its two arguments for order. If `ascending` is `true`, returns a negative integer, zero, or a positive integer as the first argument is less than, equal to, or greater than the second. If `ascending` is `false`, returns a positive integer, zero, or negative integer as the first argument is less than, equal to, or greater than the second.

Parameters

- **o1** – the first `EmailRecord` to be compared
- **o2** – the second `EmailRecord` to be compared

Returns a positive integer, zero, or negative integer indicating the result of comparing the objects

11.25.4 EmailRecords

public class **EmailRecords**<T>

The `EmailRecords` class wraps the `EmailRecord` list and stores the current item count.

Constructors

EmailRecords

public **EmailRecords** ()

Creates a new instance of EmailRecords, which contains no records.

EmailRecords

public **EmailRecords** (*Integer totalRecords*, *Integer page*, *Integer totalPages*, *List<T> allRecords*)

Creates a new instance of EmailRecords, with all fields set to the values specified in the parameters. The *page* and *totalPages* parameters are for the purposes of paginating the list of records in the UI.

Parameters

- **totalRecords** – the total number of records
- **page** – the current page
- **totalPages** – the total number of pages
- **allRecords** – the list of records

Methods

getPage

public *Integer* **getPage** ()

Gets the current page.

Returns the current page

getRecords

public *Integer* **getRecords** ()

Gets the total number of records.

Returns the total number of records

getRows

public *List<T>* **getRows** ()

Gets the list of records.

Returns the list of records

getTotal

public *Integer* **getTotal** ()

Gets the total number of pages.

Returns the total number of pages

setRows

public void **setRows** ([List<T>](#) rows)
Sets the list of records.

Parameters

- **rows** – the list of records

setTotal

public void **setTotal** ([Integer](#) total)
Sets the total number of records.

Parameters

- **total** – the total number of records

toString

public [String](#) **toString** ()
Returns a string representation of this `EmailRecords` object.

Returns a string representation of this `EmailRecords` object

11.26 org.motechproject.email.service

11.26.1 EmailAuditService

public interface **EmailAuditService**

The `EmailAuditService` interface provides methods for logging email activity, as well as searching and deleting the email logs.

Methods

countEmailRecords

long **countEmailRecords** ([EmailRecordSearchCriteria](#) criteria)
Returns the count of `EmailRecord` entries matching the specified search criteria.

Returns the count of email records matching the provided criteria

delete

void **delete** ([EmailRecord](#) emailRecord)
Deletes the specified `EmailRecord` entry from the email log.

findAllEmailRecords

List<EmailRecord> **findAllEmailRecords** ()

Finds and returns all EmailRecord entries in the email log.

Returns all email records in the email log

findById

EmailRecord **findById** (long *id*)

Finds an EmailRecord in the log by ID.

Parameters

- **id** – the identifier of the record to find

Returns the email record that matches the provided identifier, or null if no matching record exists

findEmailRecords

List<EmailRecord> **findEmailRecords** (EmailRecordSearchCriteria *criteria*)

Finds and returns all EmailRecord entries matching the specified search criteria.

Returns all email records matching the provided criteria

log

void **log** (EmailRecord *mailRecord*)

Adds the provided EmailRecord to the email log.

Parameters

- **mailRecord** – the record to add to the email log

11.26.2 EmailRecordService

public interface **EmailRecordService** extends MotechDataService<EmailRecord>

This service provides data access for `org.motechproject.email.domain.EmailRecord`. The implementation is generated by Motech Data Services and published as an OSGi service.

Methods

countFind

long **countFind** (String *fromAddress*, String *toAddress*, String *subject*, String *message*, Range<DateTime> *deliveryTimeRange*, Set<DeliveryStatus> *deliveryStatuses*)

Returns the count of all EmailRecord entries matching the specified search parameters.

Parameters

- **fromAddress** – the sender address on which to search
- **toAddress** – the recipient address on which to search
- **subject** – the subject on which to search

- **message** – the message body on which to search
- **deliveryTimeRange** – the delivery time range on which to search
- **deliveryStatuses** – the delivery statuses on which to search

Returns the count of `EmailRecord` entries that match the specified criteria

find

```
List<EmailRecord> find(String fromAddress, String toAddress, String subject, String message,  
    Range<DateTime> deliveryTimeRange, Set<DeliveryStatus> deliveryStatuses,  
    QueryParams queryParams)
```

Finds and returns all `EmailRecord` entries matching the specified search parameters. This method is exposed as a `org.motechproject.mds.annotations.Lookup` through Motech Data Services.

Parameters

- **fromAddress** – the sender address on which to search
- **toAddress** – the recipient address on which to search
- **subject** – the subject on which to search
- **message** – the message body on which to search
- **deliveryTimeRange** – the delivery time range on which to search
- **deliveryStatuses** – the delivery statuses on which to search
- **queryParams** – the query parameters to include with the search criteria

Returns the list of `EmailRecord` entries that match the specified criteria

See also: `org.motechproject.mds.annotations`

findByRecipientAddress

```
List<EmailRecord> findByRecipientAddress(String recipientAddress)
```

Finds and returns all `EmailRecord` entries for the specified recipient address. This method is exposed as a `org.motechproject.mds.annotations.Lookup` through Motech Data Services.

Parameters

- **recipientAddress** – the recipient address on which to search

Returns the list of `EmailRecord` entries that match the specified address

See also: `org.motechproject.mds.annotations`

11.26.3 EmailSenderService

public interface **EmailSenderService**

The `EmailSenderService` interface provides a method for sending email.

Methods

send

void **send** (*Mail message*)

Attempts to send the supplied email message. Adds an `org.motechproject.email.domain.EmailRecord` entry to the log with the details of the activity.

Parameters

- **message** – the message to send

Throws

- **MailException** – if an error occurs when attempting to send the message

11.27 org.motechproject.event

11.27.1 MotechEvent

public class **MotechEvent** implements `Serializable`

Motech Scheduled Event data carrier class, Instance of this class with event specific data will be send by Motech Scheduler when a scheduled event is fired

This class is immutable

Fields

EVENT_TYPE_KEY_NAME

public static final `String` **EVENT_TYPE_KEY_NAME**

PARAM_DISCARDED_MOTECH_EVENT

public static final `String` **PARAM_DISCARDED_MOTECH_EVENT**

PARAM_INVALID_MOTECH_EVENT

public static final `String` **PARAM_INVALID_MOTECH_EVENT**

PARAM_REDELIVERY_COUNT

public static final `String` **PARAM_REDELIVERY_COUNT**

Constructors

MotechEvent

public **MotechEvent** ()

MotechEvent

public **MotechEvent** (*String subject*)

Constructor with subject only (parameters can be added interactively)

Parameters

- **subject** –
 - event destination

Throws

- **IllegalArgumentException** –

MotechEvent

public **MotechEvent** (*String subject, Map<String, Object> parameters*)

Constructor

Parameters

- **subject** –
 - event type: Pill Reminder, Appointment Reminder ...
- **parameters** –
 - a Map of additional parameters

Throws

- **IllegalArgumentException** –

Methods

copy

public **MotechEvent** **copy** (*String subject, Map<String, Object> parameters*)

equals

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

getEndTime

public **Date** **getEndTime** ()

getId

public **UUID** **getId** ()

getMessageRedeliveryCount

```
public int getMessageRedeliveryCount ()
```

getParameters

```
public Map<String, Object> getParameters ()  
    Sets empty HashMap if parameters=null
```

getSubject

```
public String getSubject ()
```

hashCode

```
public int hashCode ()
```

incrementMessageRedeliveryCount

```
public void incrementMessageRedeliveryCount ()
```

isLastEvent

```
public boolean isLastEvent ()
```

setEndTime

```
public MotechEvent setEndTime (Date endDate)
```

setId

```
public void setId (UUID id)
```

setLastEvent

```
public MotechEvent setLastEvent (boolean lastEvent)
```

toString

```
public String toString ()
```

11.28 org.motechproject.event.listener

11.28.1 EventListener

public interface **EventListener**

Methods

getIdentifier

String **getIdentifier** ()

Retrieve a unique identifier/key for the given listener class. This identifier is used when messages are destined for this specific listener type

Returns Unique listener identifier/key

handle

void **handle** (*MotechEvent event*)

Handle an particular event that has been received

Parameters

- **event** –

11.28.2 EventListenerRegistryService

public interface **EventListenerRegistryService**

This interface is necessary for OSGi service publication. The implementing class acts as a registry for all scheduled event listeners. One can register themselves to listen for a specific set of event types.

Methods

clearListenersForBean

void **clearListenersForBean** (*String beanName*)

This method is responsible for removing listeners for a particular bean. This is necessary when bundles are stopped in some fashion so that the listener does not persist.

Parameters

- **beanName** – The bean name from the Spring context of the candidate class for listener clearing

getListenerCount

int **getListenerCount** (*String subject*)

Get the count of listeners for a particular subject.

Parameters

- **subject** –

getListeners

Set<EventListener> **getListeners** (*String subject*)

Retrieve a list of event listeners for a given event type. If there are no listeners, an empty list is returned.

Parameters

- **subject** – The event type that you are seeking listeners for

Returns A list of scheduled event listeners that are interested in that event

hasListener

boolean **hasListener** (*String subject*)

See if a particular subject has any listeners.

Parameters

- **subject** –

registerListener

void **registerListener** (EventListener *listener*, List<String> *subjects*)

Register an event listener to be notified when events of a given type are received via the Server JMS Event Queue.

Parameters

- **listener** – the listener instance
- **subjects** – the event types that a listener is interested in

registerListener

void **registerListener** (EventListener *listener*, String *subject*)

11.28.3 EventRelay

public interface **EventRelay**

Methods

sendEventMessage

void **sendEventMessage** (MotechEvent *motechEvent*)

11.29 org.motechproject.event.listener.annotations

11.29.1 MotechListener

public @interface **MotechListener**

Author yyonkov

11.29.2 MotechListenerAbstractProxy

public abstract class **MotechListenerAbstractProxy** implements [EventListener](#)
Event Listener Proxy base abstract class

Author yyonkov

Constructors

MotechListenerAbstractProxy

public **MotechListenerAbstractProxy** ([String](#) *name*, [Object](#) *bean*, [Method](#) *method*)

Parameters

- **name** –
- **bean** –
- **method** –

Methods

callHandler

public abstract void **callHandler** ([MotechEvent](#) *event*)
Needs to be implemented by concrete Proxies

Parameters

- **event** –

getBean

public [Object](#) **getBean** ()

getIdentifier

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

getMethod

public [Method](#) **getMethod** ()

handle

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

11.29.3 MotechListenerEventProxy

public class **MotechListenerEventProxy** extends [MotechListenerAbstractProxy](#)

Responsible for dispatching to void `doSomething(MotechEvent event)` {} type of handlers

Author yyonkov

Constructors

MotechListenerEventProxy

public **MotechListenerEventProxy** ([String name](#), [Object bean](#), [Method method](#))

Parameters

- **name** –
- **bean** –
- **method** –

Methods

callHandler

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

11.29.4 MotechListenerNamedParametersProxy

public class **MotechListenerNamedParametersProxy** extends [MotechListenerAbstractProxy](#)

Author yyonkov

Constructors

MotechListenerNamedParametersProxy

public **MotechListenerNamedParametersProxy** ([String name](#), [Object bean](#), [Method method](#))

Parameters

- **name** –
- **bean** –
- **method** –

Methods

callHandler

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

11.29.5 MotechListenerType

public enum **MotechListenerType**

ENUM defining the annotation types

Author yyonkov

Enum Constants

MOTech_EVENT

public static final [MotechListenerType](#) **MOTech_EVENT**

NAMED_PARAMETERS

public static final [MotechListenerType](#) **NAMED_PARAMETERS**

11.29.6 MotechParam

public @interface **MotechParam**

Author yyonkov

11.30 org.motechproject.event.queue

11.30.1 MotechCachingConnectionFactory

public class **MotechCachingConnectionFactory** extends [CachingConnectionFactory](#)

The `MotechCachingConnectionFactory` is used to created connection to ActiveMQ.

Methods

doCreateConnection

protected [Connection](#) **doCreateConnection** ()

setBrokerUrl

public void **setBrokerUrl** ([String](#) *brokerURL*)

setPassword

public void **setPassword** ([String](#) *password*)

setUsername

```
public void setUsername (String username)
```

11.30.2 MotechEventConfig

```
public class MotechEventConfig
```

Methods**getMessageMaxRedeliveryCount**

```
public int getMessageMaxRedeliveryCount ()
```

getMessageRedeliveryDelay

```
public long getMessageRedeliveryDelay ()
```

11.30.3 MotechEventHeaderMapper

```
public class MotechEventHeaderMapper extends DefaultJmsHeaderMapper
```

For the delay to work, set attribute schedulerSupport="true" in the broker element of the activemq.xml Ref:
<http://activemq.apache.org/delay-and-schedule-message-delivery.html>

Methods**fromHeaders**

```
public void fromHeaders (MessageHeaders messageHeaders, Message message)
```

11.30.4 MotechEventTransformer

```
public class MotechEventTransformer
```

Methods**transform**

```
public MotechEvent transform (MotechEvent motechEvent)
```

11.30.5 OutboundEventGateway

```
public interface OutboundEventGateway
```

Methods

sendEventMessage

void **sendEventMessage** (*MotechEvent* *motechEvent*)

Sends the given MotechEvent message as a payload to the message channel defined in the Spring Integration configuration file.

Parameters

- **motechEvent** –

11.31 org.motechproject.mds.annotations

11.31.1 Cascade

public @interface **Cascade**

The Cascade annotation is used by developers to set correct cascade properties for the given field that is a relationship.

See also: `org.motechproject.mds.annotations.internal.FieldProcessor`

11.31.2 Entity

public @interface **Entity**

The Entity annotation is used by developers to point classes, that should be mapped as Motech Dataservices Entities. The discovery logic for this annotation is done in `org.motechproject.mds.annotations.internal.EntityProcessor`

See also: `org.motechproject.mds.annotations.internal.EntityProcessor`

11.31.3 Field

public @interface **Field**

The Field annotation is used by developers to point fields, that should be mapped as entity fields. The discovery logic for this annotation is done in `org.motechproject.mds.annotations.internal.FieldProcessor`.

Only fields, 'getter' or 'setter' methods can have this annotation for other methods this annotation is omitted.

See also: `org.motechproject.mds.annotations.internal.FieldProcessor`

11.31.4 Ignore

public @interface **Ignore**

Thanks to this annotation, developers can point public fields and getter/setter method of a private field that should not be added to the schema definition of an entity. By default, all public fields of an object are added. The discovery logic for this annotation is done in the FieldProcessor

See also: `org.motechproject.mds.annotations.internal.FieldProcessor`

11.31.5 InSet

public @interface **InSet**

The annotated element must have value that will be in defined set.

Supported types are:

- Integer
- Double
- int, double

11.31.6 Lookup

public @interface **Lookup**

The `Lookup` annotation is used by developers to point methods, in classes that implements `org.motechproject.mds.service.MotechDataService`, that should be mapped as Motech Dataservices Lookups. The discovery logic for this annotation is done in `org.motechproject.mds.annotations.internal.LookupProcessor`

See also: `org.motechproject.mds.annotations.internal.LookupProcessor`

11.31.7 LookupField

public @interface **LookupField**

The `LookupField` annotation allows developers to point fields in `Lookup` method, that should be mapped as `Lookup` fields for Developer Defined `Lookup`. The discovery logic for this annotation is done in `org.motechproject.mds.annotations.internal.LookupProcessor`

See also: `org.motechproject.mds.annotations.internal.LookupProcessor`

11.31.8 NotInSet

public @interface **NotInSet**

The annotated element must not have value that will be in defined set.

Supported types are:

- Integer
- Double
- int, double

11.31.9 RestExposed

public @interface **RestExposed**

The `RestExposed` annotation is used by developers to mark lookups that should be exposed by REST. It is being processed by `org.motechproject.mds.annotations.internal.LookupProcessor`.

See also: `org.motechproject.mds.annotations.internal.LookupProcessor`

11.31.10 RestIgnore

public @interface **RestIgnore**

The `RestIgnore` annotation is used by developer to mark a field of entity as not exposed over REST. By default all fields (including auto-generated ones) are exposed. To ignore one of the auto-generated fields, it have to be declared in child entity and marked with this annotation.

This annotation is processed by `org.motechproject.mds.annotations.internal.RestIgnoreProcessor`

See also: `org.motechproject.mds.annotations.internal.RestIgnoreProcessor`

11.31.11 RestOperations

public @interface **RestOperations**

The `RestOperations` annotation is used by developers to specify which CRUD operations should be enabled for entity. `RestOperations` value is an array of one or more values specified in `org.motechproject.mds.domain.RestOperation` enum, that is: `CREATE`, `READ`, `UPDATE`, `DELETE`. There is also one special value - `ALL`. When provided, all CRUD operations are enabled for entity, regardless of presence of other values.

This annotation is processed by `org.motechproject.mds.annotations.internal.RestOperationsProcessor` and can be applied only to class which is also annotated with `org.motechproject.mds.annotations.Entity`. It has no effect otherwise.

See also: `org.motechproject.mds.domain.RestOperation`, `org.motechproject.mds.annotations.int`

11.31.12 UIDisplayable

public @interface **UIDisplayable**

The `UIDisplayable` annotation is used by developers to mark a field as being in the default display for a listing of objects. The discovery logic for this annotation is done in `org.motechproject.mds.annotations.internal.UIDisplayableProcessor`.

Only fields, 'getter' or 'setter' methods can have this annotation for other methods this annotation is omitted.

See also: `org.motechproject.mds.annotations.internal.UIDisplayableProcessor`

11.31.13 UIFilterable

public @interface **UIFilterable**

The `UIFilterable` annotation is used by developers to mark a field that allow users to filter a list of objects by the values in the field. The discovery logic for this annotation is done in `org.motechproject.mds.annotations.internal.UIFilterableProcessor`.

Only fields, 'getter' or 'setter' methods can have this annotation for other methods this annotation is omitted. Also this annotation is permitted on fields of type: `Date`, `DateTime`, `LocalDate`, `Boolean` or `List`.

See also: `org.motechproject.mds.annotations.internal.UIFilterableProcessor`

11.32 org.motechproject.mds.builder

11.32.1 EntityBuilder

public interface **EntityBuilder**

An entity builder is responsible for building the entity class from an Entity schema.

Methods

build

ClassData **build** (Entity *entity*)

Builds a class definition for a given entity. The class is not registered with any classloader.

Parameters

- **entity** – the entity schema

Returns bytes of the newly constructed class

buildDDE

ClassData **buildDDE** (Entity *entity*, Bundle *bundle*)

Builds Developer Defined Entity. The main difference between regular build method and this one, is that this method fetches the class definition from the given bundle, and injects members to the constructed class from there, if possible, rather than building everything from scratch.

Parameters

- **entity** – the entity schema
- **bundle** – the bundle to fetch class definition from

Returns bytes of the constructed class

buildHistory

ClassData **buildHistory** (Entity *entity*)

Builds History class definition for the given entity. The history class definition contains the same members as the entity class, plus some fields history-exclusive, like schema version.

Parameters

- **entity** – the entity schema

Returns bytes of the constructed class

buildTrash

ClassData **buildTrash** (Entity *entity*)

Builds Trash class definition for the given entity. The trash class definition contains the same members as the entity class, plus some fields trash-exclusive.

Parameters

- **entity** – the entity schema

Returns bytes of the constructed class

prepareHistoryClass

void **prepareHistoryClass** (*Entity entity*)

Builds empty history class definition for the given entity and adds it to the class pool.

Parameters

- **entity** – the entity schema

prepareTrashClass

void **prepareTrashClass** (*Entity entity*)

Builds empty trash class definition for the given entity and adds it to the class pool.

Parameters

- **entity** – the entity schema

11.32.2 EntityInfrastructureBuilder

public interface **EntityInfrastructureBuilder**

The `EntityInfrastructureBuilder` is responsible for building infrastructure for a given entity: repository, interface and service classes.

Methods

buildHistoryInfrastructure

List<ClassData> **buildHistoryInfrastructure** (*String className*)

Builds the repository, interface and implementation of this interface classes for the given history class. The names for classes are generated by `org.motechproject.mds.util.ClassName.getRepositoryName(String)`, `org.motechproject.mds.util.ClassName.getInterfaceName(String)`, `org.motechproject.mds.util.ClassName.getServiceName(String)`, respectively.

Parameters

- **className** – a name of history class.

Returns a list of classes that represents infrastructure for the history class.

buildInfrastructure

List<ClassData> **buildInfrastructure** (*Entity entity*)

Builds the repository, interface and implementation of this interface classes for the given entity. The names for classes are generated by `org.motechproject.mds.util.ClassName.getRepositoryName(String)`, `org.motechproject.mds.util.ClassName.getInterfaceName(String)`, `org.motechproject.mds.util.ClassName.getServiceName(String)`, respectively.

Parameters

- **entity** – an instance of `org.motechproject.mds.domain.Entity`

Returns a list of classes that represents infrastructure for the given entity.

11.32.3 EntityMetadataBuilder

public interface **EntityMetadataBuilder**

The `EntityMetadataBuilderImpl` class is responsible for building jdo metadata for an entity class.

Methods**addBaseMetadata**

void **addBaseMetadata** (`JDOMetadata jdoMetadata`, `ClassData classData`, `EntityType entityType`)

Adds base information about package and class name to a `javax.jdo.metadata.JDOMetadata` instance.

Parameters

- **jdoMetadata** – an empty instance of `javax.jdo.metadata.JDOMetadata`.
- **classData** – an instance of `org.motechproject.mds.domain.ClassData`
- **entityType** – type of the entity(regular, history or trash)

addEntityMetadata

void **addEntityMetadata** (`JDOMetadata jdoMetadata`, `Entity entity`)

Adds information about package and class name to a `javax.jdo.metadata.JDOMetadata` instance.

Parameters

- **jdoMetadata** – a empty instance of `javax.jdo.metadata.JDOMetadata`.
- **entity** – a instance of `org.motechproject.mds.domain.Entity`

addHelperClassMetadata

void **addHelperClassMetadata** (`JDOMetadata jdoMetadata`, `ClassData classData`, `Entity entity`, `EntityType entityType`)

Creates metadata with basic information about package and class name to the `javax.jdo.metadata.JDOMetadata` instance. Additionally, fetches fields from passed entities and adds metadata for fields, if it's necessary. If entity is null, it will work just like `addBaseMetadata(JDOMetadata, ClassData)` and won't add any metadata for fields.

Parameters

- **jdoMetadata** – an empty instance of `javax.jdo.metadata.JDOMetadata`.
- **classData** – an instance of `org.motechproject.mds.domain.ClassData`
- **entity** – an entity to fetch fields from
- **entityType** – type of the entity(history or trash)

fixEnhancerIssuesInMetadata

void **fixEnhancerIssuesInMetadata** (*JDOMetadata jdoMetadata*)

This updates the metadata after enhancement. Nucleus makes some “corrections” which do not work with the runtime enhancer.

11.32.4 EnumBuilder

public interface **EnumBuilder**

An enum builder is responsible for building the enum class with the same values as those are defined in the field.

Methods

build

ClassData **build** (*ComboBoxHolder holder*)

11.32.5 MDSConstructor

public interface **MDSConstructor**

This interface provide method to create a class for the given entity. The implementation of this interface should also construct other classes like repository, service interface and implementation for this service interface.

Methods

constructEntities

boolean **constructEntities** (boolean *buildDDE*)

Creates a class definition and insert it into the MDS class loader based on data from database. The implementation of this method should also create a repository, interface (when it’s necessary) and implementation of this interface.

After executing this method, there should be possible to create an instance of the given class definition and save it to database by *javax.jdo.PersistenceManager* provided by DataNucleus.

An interface related with class definition should be created only for entities from outside bundles and if the bundle does not define its interface.

Parameters

- **buildDDE** – *true* if class definitions for entities from outside bundles should also be created; otherwise *false*.

Returns *true* if there were entities for which class definitions should be created; otherwise *false*.

updateFields

void **updateFields** (*Long entityId*, *Map<String, String> fieldNameChanges*)

Updates the field names of an entity. This method changes the database schema by changing column names to the new value. This is done for the entity instances, history instances and trash instances.

Parameters

- **entityId** – The ID of an entity to update
- **fieldNameChanges** – A map, indexed by current field names and values being updated field names.

11.32.6 MDSDataProviderBuilder

public interface **MDSDataProviderBuilder**

The `MDSDataProviderBuilder` class is responsible for building the MDS Data Provider JSON, required to register a data provider in the Tasks module.

Methods

`generateDataProvider`

`String generateDataProvider()`

Generates JSON String, containing information about all entities with lookups. It contains all the information required by the task module to register a data provider.

Returns JSON-formatted String, containing information about MDS Data Provider

11.33 org.motechproject.mds.config

11.33.1 DeleteMode

public enum **DeleteMode**

The `DeleteMode` presents what should happen with objects when they are deleted. They can be deleted permanently `DELETE` or moved to the trash `TRASH`. This enum is related with the property `org.motechproject.mds.util.Constants.Config.MDS_DELETE_MODE`.

The `UNKNOWN` value should not be used in code as appropriate value. It was added to ensure that the `fromString(String)` method will not return `{@value null}` value.

Enum Constants

DELETE

public static final `DeleteMode` **DELETE**

TRASH

public static final `DeleteMode` **TRASH**

UNKNOWN

public static final `DeleteMode` **UNKNOWN**

11.33.2 MdsConfig

public class **MdsConfig**

Class responsible for handling MDS configuration. Since MDS don't use Server Config everything connected to the MDS configuration needs to be handled by the module itself.

Constructors

MdsConfig

public **MdsConfig** ()

Methods

asProperties

public [Properties](#) **asProperties** ()

getDataNucleusProperties

public [Properties](#) **getDataNucleusProperties** ()

getProperties

public [Properties](#) **getProperties** ([String](#) filename)

getResourceFileName

public [String](#) **getResourceFileName** ([Resource](#) resource)

setConfig

public void **setConfig** ([List](#)<[Resource](#)> resources)

11.33.3 ModuleSettings

public class **ModuleSettings** extends [Properties](#)

The `ModuleSettings` contains the base module settings which are inside the `org.motechproject.mds.util.Constants.Config.MODULE_FILE`. The getters and setters inside this class always checks the given property and if it is incorrect then the default value of the given property will be returned.

Fields

DEFAULT_DELETE_MODE

public static final [DeleteMode](#) **DEFAULT_DELETE_MODE**

DEFAULT_EMPTY_TRASH

```
public static final Boolean DEFAULT_EMPTY_TRASH
```

DEFAULT_TIME_UNIT

```
public static final TimeUnit DEFAULT_TIME_UNIT
```

DEFAULT_TIME_VALUE

```
public static final Integer DEFAULT_TIME_VALUE
```

Methods**equals**

```
public boolean equals (Object obj)
```

getDeleteMode

```
public DeleteMode getDeleteMode ()
```

getTimeUnit

```
public TimeUnit getTimeUnit ()
```

getTimeValue

```
public Integer getTimeValue ()
```

hashCode

```
public int hashCode ()
```

isEmptyTrash

```
public Boolean isEmptyTrash ()
```

setDeleteMode

```
public void setDeleteMode (String deleteMode)
```

setDeleteMode

```
public void setDeleteMode (DeleteMode deleteMode)
```

setEmptyTrash

```
public void setEmptyTrash (String emptyTrash)
```

setEmptyTrash

```
public void setEmptyTrash (Boolean emptyTrash)
```

setTimeUnit

```
public void setTimeUnit (String timeUnit)
```

setTimeUnit

```
public void setTimeUnit (TimeUnit timeUnit)
```

setTimeValue

```
public void setTimeValue (String timeValue)
```

setTimeValue

```
public void setTimeValue (Integer timeValue)
```

toString

```
public String toString ()
```

11.33.4 ModuleSettings.Deserializer

```
public static final class Deserializer extends JsonSerializer<ModuleSettings>
```

Methods**deserialize**

```
public ModuleSettings deserialize (JsonParser jp, DeserializationContext ctxt)
```

11.33.5 ModuleSettings.Serializer

```
public static final class Serializer extends JsonSerializer<ModuleSettings>
```

Methods

serialize

public void **serialize** ([ModuleSettings](#) *value*, [JsonGenerator](#) *jgen*, [SerializerProvider](#) *provider*)

11.33.6 SettingsService

public interface **SettingsService**

The `SettingsService` is a service class for `org.motechproject.server.config.SettingsFacade`.

Its main purpose is to create better access to module settings for developers.

Methods

getDeleteMode

[DeleteMode](#) **getDeleteMode** ()

getModuleSettings

[ModuleSettings](#) **getModuleSettings** ()

getProperties

[Properties](#) **getProperties** ()

getTimeUnit

[TimeUnit](#) **getTimeUnit** ()

getTimeValue

[Integer](#) **getTimeValue** ()

isEmptyTrash

[Boolean](#) **isEmptyTrash** ()

saveModuleSettings

void **saveModuleSettings** ([ModuleSettings](#) *settings*)

11.33.7 TimeUnit

public enum **TimeUnit**

The `TimeUnit` specifies what time unit should be used to specify time when the module trash should be cleaned. This enum is related with the property `org.motechproject.mds.util.Constants.Config.MDS_TIME_UNIT`.

Each value from this enum can be converted to long value that presents time interval in milliseconds. For example the `HOURS` value is equal to `{@value 3.6E6}`.

The `UNKNOWN` value should not be used in code as appropriate value. It was added to ensure that the `fromString(String)` method will not return `{@value null}` value.

Enum Constants

DAYS

public static final `TimeUnit` **DAYS**

HOURS

public static final `TimeUnit` **HOURS**

MONTHS

public static final `TimeUnit` **MONTHS**

UNKNOWN

public static final `TimeUnit` **UNKNOWN**

WEEKS

public static final `TimeUnit` **WEEKS**

YEARS

public static final `TimeUnit` **YEARS**

11.34 org.motechproject.mds.domain

11.34.1 BrowsingSettings

public class **BrowsingSettings**

The `BrowsingSettings` contains information about fields that will be visible on UI and could be used as filter on UI.

This class is read only (the data are not saved to database) and its main purpose is to provide methods that help developer to get displayed and filterable fields.

Constructors

BrowsingSettings

```
public BrowsingSettings (Entity entity)
```

Methods

getDisplayedFields

```
public List<Field> getDisplayedFields ()
```

getEntity

```
public Entity getEntity ()
```

getFilterableFields

```
public List<Field> getFilterableFields ()
```

setEntity

```
public void setEntity (Entity entity)
```

toDto

```
public BrowsingSettingsDto toDto ()
```

11.34.2 ClassData

```
public class ClassData  
    Represents a class name and its byte code.
```

Constructors

ClassData

```
public ClassData (String className, byte[] bytecode)
```

ClassData

```
public ClassData (String className, byte[] bytecode, boolean interfaceClass)
```

ClassData

```
public ClassData (Entity entity, byte[] bytecode)
```

ClassData

```
public ClassData (Entity entity, byte[] bytecode, boolean interfaceClass)
```

ClassData

```
public ClassData (String className, String module, String namespace, byte[] bytecode)
```

ClassData

```
public ClassData (String className, String module, String namespace, byte[] bytecode, EntityType type)
```

ClassData

```
public ClassData (String className, String module, String namespace, byte[] bytecode, boolean interface-  
Class)
```

ClassData

```
public ClassData (String className, String module, String namespace, byte[] bytecode, boolean interface-  
Class, EntityType type)
```

Methods

getBytecode

```
public byte[] getBytecode ()
```

getClassName

```
public String getClassName ()
```

getModule

```
public String getModule ()
```

getNamespace

```
public String getNamespace ()
```

getType

```
public EntityType getType ()
```

isDDE

```
public boolean isDDE ()
```

isInterfaceClass

```
public boolean isInterfaceClass ()
```

toString

```
public String toString ()
```

11.34.3 ComboboxHolder

```
public class ComboboxHolder extends FieldHolder
```

The main purpose of this class is to find out what kind of type should be used when the field will be added to the class definition.

Constructors**ComboboxHolder**

```
public ComboboxHolder (Field field)
```

ComboboxHolder

```
public ComboboxHolder (Entity entity, Field field)
```

ComboboxHolder

```
public ComboboxHolder (EntityDto entity, FieldDto field)
```

ComboboxHolder

```
public ComboboxHolder (Class<?> entityClass, FieldDto field)
```

ComboboxHolder

```
public ComboboxHolder (List<? extends Pair<String, String>> metadata, List<? extends Pair<String, ?>>  
    settings, String defaultEnumName)
```

Methods

`getEnumName`

```
public String getEnumName ()
```

`getValues`

```
public String[] getValues ()
```

`isAllowMultipleSelections`

```
public boolean isAllowMultipleSelections ()
```

`isAllowUserSupplied`

```
public boolean isAllowUserSupplied ()
```

`isEnum`

```
public boolean isEnum ()
```

`isEnumList`

```
public boolean isEnumList ()
```

`isString`

```
public boolean isString ()
```

`isStringList`

```
public boolean isStringList ()
```

11.34.4 ConfigSettings

```
public class ConfigSettings
```

The `ConfigSettings` class represents Data Services settings, that can be adjusted by users via UI.

Constructors

`ConfigSettings`

```
public ConfigSettings ()
```


ConfigSettings

public **ConfigSettings** ([DeleteMode](#) *deleteMode*, boolean *emptyTrash*, int *afterTimeValue*, [TimeUnit](#) *afterTimeUnit*)

Methods

getAfterTimeUnit

public [TimeUnit](#) **getAfterTimeUnit** ()

getAfterTimeValue

public int **getAfterTimeValue** ()

getDeleteMode

public [DeleteMode](#) **getDeleteMode** ()

getEmptyTrash

public boolean **getEmptyTrash** ()

getId

public [Long](#) **getId** ()

setAfterTimeUnit

public void **setAfterTimeUnit** ([TimeUnit](#) *afterTimeUnit*)

setAfterTimeValue

public void **setAfterTimeValue** (int *afterTimeValue*)

setDeleteMode

public void **setDeleteMode** ([DeleteMode](#) *deleteMode*)

setEmptyTrash

public void **setEmptyTrash** (boolean *emptyTrash*)

setId

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

11.34.5 Entity

```
public class Entity
```

The `Entity` class contains information about an entity. Also it contains information about advanced settings related with the entity.

Constructors

Entity

```
public Entity ()
```

Entity

```
public Entity (String className)
```

Entity

```
public Entity (String className, String module, String namespace, SecurityMode securityMode)
```

Entity

```
public Entity (String className, String name, String module, String namespace, SecurityMode security-  
Mode, Set<String> securityMembers)
```

Methods

addField

```
public void addField (Field field)
```

addLookup

```
public void addLookup (Lookup lookup)
```

advancedSettingsDto

```
public AdvancedSettingsDto advancedSettingsDto ()
```

getBrowsingSettings

```
public BrowsingSettings getBrowsingSettings ()
```

getClassName

```
public String getClassName ()
```

getDrafts

```
public List<EntityDraft> getDrafts ()
```

getEntityVersion

```
public Long getEntityVersion ()
```

getField

```
public Field getField (Long id)
```

getField

```
public Field getField (String name)
```

getFieldDtos

```
public List<FieldDto> getFieldDtos ()
```

getFields

```
public List<Field> getFields ()
```

getId

```
public Long getId ()
```

getLookupById

```
public Lookup getLookupById (Long lookupId)
```

getLookupByName

```
public Lookup getLookupByName (String lookupName)
```

getLookupDtos

```
public List<LookupDto> getLookupDtos ()
```

getLookups

```
public List<Lookup> getLookups ()
```

getModule

```
public String getModule ()
```

getName

```
public String getName ()
```

getNamespace

```
public String getNamespace ()
```

getRestOptions

```
public RestOptions getRestOptions ()
```

getSecurityMembers

```
public Set<String> getSecurityMembers ()
```

getSecurityMode

```
public SecurityMode getSecurityMode ()
```

getStringComboboxFields

```
public List<Field> getStringComboboxFields ()
```

getSuperClass

```
public String getSuperClass ()
```

getTracking

```
public Tracking getTracking ()
```

isAbstractClass

public boolean **isAbstractClass** ()

isActualEntity

public boolean **isActualEntity** ()

isBaseEntity

public boolean **isBaseEntity** ()

isDDE

public boolean **isDDE** ()

isDraft

public boolean **isDraft** ()

isSubClassOfMdsEntity

public boolean **isSubClassOfMdsEntity** ()

removeField

public void **removeField** (Long *fieldId*)

removeLookup

public void **removeLookup** (Long *lookupId*)

setAbstractClass

public void **setAbstractClass** (boolean *abstractClass*)

setClassName

public void **setClassName** (String *className*)

setDrafts

public void **setDrafts** (List<EntityDraft> *drafts*)

setEntityVersion

public void **setEntityVersion** ([Long](#) *entityVersion*)

setFields

public void **setFields** ([List](#)<[Field](#)> *fields*)

setId

public void **setId** ([Long](#) *id*)

setLookups

public void **setLookups** ([List](#)<[Lookup](#)> *lookups*)

setModule

public void **setModule** ([String](#) *module*)

setName

public final void **setName** ([String](#) *name*)

setNamespace

public void **setNamespace** ([String](#) *namespace*)

setRestOptions

public void **setRestOptions** ([RestOptions](#) *restOptions*)

setSecurity

public void **setSecurity** ([SecurityMode](#) *securityMode*, [List](#)<[String](#)> *securityMembersList*)

setSecurityMembers

public void **setSecurityMembers** ([Set](#)<[String](#)> *securityMembers*)

setSecurityMode

public void **setSecurityMode** ([SecurityMode](#) *securityMode*)

setSuperClass

```
public void setSuperClass (String superClass)
```

setTracking

```
public void setTracking (Tracking tracking)
```

supportsAnyRestOperations

```
public boolean supportsAnyRestOperations ()
```

toDto

```
public EntityDto toDto ()
```

updateAdvancedSetting

```
public void updateAdvancedSetting (AdvancedSettingsDto advancedSettings)
```

updateFromDraft

```
public void updateFromDraft (EntityDraft draft)
```

updateRestOptions

```
public void updateRestOptions (RestOptionsDto restOptionsDto)
```

11.34.6 EntityAudit

```
public class EntityAudit extends Entity
```

This class represents a single historical commit of an Entity.

Methods**getModificationDate**

```
public DateTime getModificationDate ()
```

getOwnerUsername

```
public String getOwnerUsername ()
```

getVersion

```
public Long getVersion ()
```

isActualEntity

```
public boolean isActualEntity ()
```

setModificationDate

```
public void setModificationDate (DateTime modificationDate)
```

setOwnerUsername

```
public void setOwnerUsername (String ownerUsername)
```

setVersion

```
public void setVersion (Long version)
```

11.34.7 EntityDraft

```
public class EntityDraft extends Entity
```

This class represents a users draft of an Entity. A draft is a users work in progress from the UI. This shares the table with its superclass, `Entity`.

Constructors

EntityDraft

```
public EntityDraft ()
```

Methods

getDraftOwnerUsername

```
public String getDraftOwnerUsername ()
```

getFieldNameChanges

```
public Map<String, String> getFieldNameChanges ()
```

getLastModificationDate

```
public DateTime getLastModificationDate ()
```


getParentEntity

```
public Entity getParentEntity ()
```

getParentVersion

```
public Long getParentVersion ()
```

isActualEntity

```
public boolean isActualEntity ()
```

isChangesMade

```
public boolean isChangesMade ()
```

isDraft

```
public boolean isDraft ()
```

isOutdated

```
public boolean isOutdated ()
```

setChangesMade

```
public void setChangesMade (boolean changesMade)
```

setDraftOwnerUsername

```
public void setDraftOwnerUsername (String draftOwnerUsername)
```

setFieldNameChanges

```
public void setFieldNameChanges (Map<String, String> fieldNameChanges)
```

setLastModificationDate

```
public void setLastModificationDate (DateTime lastModificationDate)
```

setParentEntity

```
public void setParentEntity (Entity parentEntity)
```

setParentVersion

public void **setParentVersion** (*Long parentVersion*)

toDto

public *EntityDto* **toDto** ()

11.34.8 EntityInfo

public class **EntityInfo**

The *EntityInfo* class contains base information about the given entity like class name or infrastructure classes name.

See also: *org.motechproject.mds.service.JarGeneratorService*

Methods

getClassName

public *String* **getClassName** ()

getEntityName

public *String* **getEntityName** ()

getInfrastructure

public *String*[] **getInfrastructure** ()

getInterfaceName

public *String* **getInterfaceName** ()

getModule

public *String* **getModule** ()

getName

public *String* **getName** ()

getNamespace

public *String* **getNamespace** ()

getRepository

```
public String getRepository ()
```

getRestId

```
public String getRestId ()
```

getServiceName

```
public String getServiceName ()
```

setClassName

```
public void setClassName (String className)
```

setEntityName

```
public void setEntityName (String entityName)
```

setInterfaceName

```
public void setInterfaceName (String interfaceName)
```

setModule

```
public void setModule (String module)
```

setNamespace

```
public void setNamespace (String namespace)
```

setRepository

```
public void setRepository (String repository)
```

setServiceName

```
public void setServiceName (String serviceName)
```

11.34.9 EntityType

```
public enum EntityType
```

Enum Constants

HISTORY

public static final [EntityType](#) **HISTORY**

STANDARD

public static final [EntityType](#) **STANDARD**

TRASH

public static final [EntityType](#) **TRASH**

11.34.10 Field

public class **Field**

The `Field` class contains information about a single field.

Constructors

Field

public **Field**()

Field

public **Field**([Entity](#) entity, [String](#) displayName, [String](#) name, [Set](#)<[Lookup](#)> lookups)

Field

public **Field**([Entity](#) entity, [String](#) name, [String](#) displayName, [Type](#) type, boolean required, boolean readOnly)

Field

public **Field**([Entity](#) entity, [String](#) displayName, [String](#) name, boolean required, boolean readOnly, [String](#) defaultValue, [String](#) tooltip, [Set](#)<[Lookup](#)> lookups)

Methods

addMetadata

public void **addMetadata**([FieldMetadata](#) metadata)

addSetting

```
public void addSetting (FieldSetting setting)
```

addValidation

```
public void addValidation (FieldValidation validation)
```

copy

```
public Field copy ()
```

getDefaultValue

```
public String getDefaultValue ()
```

getDisplayName

```
public String getDisplayName ()
```

getEntity

```
public Entity getEntity ()
```

getId

```
public Long getId ()
```

getLookups

```
public Set<Lookup> getLookups ()
```

getMetadata

```
public List<FieldMetadata> getMetadata ()
```

getMetadata

```
public FieldMetadata getMetadata (String key)
```

getMetadataById

```
public FieldMetadata getMetadataById (Long id)
```

getName

public `String` **getName** ()

getSettingByName

public `FieldSetting` **getSettingByName** (`String` *name*)

getSettings

public `List<FieldSetting>` **getSettings** ()

getTooltip

public `String` **getTooltip** ()

getType

public `Type` **getType** ()

getUIDisplayPosition

public `Long` **getUIDisplayPosition** ()

getValidationByName

public `FieldValidation` **getValidationByName** (`String` *name*)

getValidations

public `List<FieldValidation>` **getValidations** ()

isExposedViaRest

public boolean **isExposedViaRest** ()

isReadOnly

public boolean **isReadOnly** ()

isRequired

public boolean **isRequired** ()

isTracked

```
public boolean isTracked()
```

isUIDisplayable

```
public boolean isUIDisplayable()
```

isUIFilterable

```
public boolean isUIFilterable()
```

setDefaultValue

```
public void setDefaultValue(String defaultValue)
```

setDisplayName

```
public void setDisplayName(String displayName)
```

setEntity

```
public void setEntity(Entity entity)
```

setExposedViaRest

```
public void setExposedViaRest(boolean exposedViaRest)
```

setId

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

setLookups

```
public void setLookups(Set<Lookup> lookups)
```

setMetadata

```
public void setMetadata(List<FieldMetadata> metadata)
```

setName

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

setReadOnly

public void **setReadOnly** (boolean *readOnly*)

setRequired

public void **setRequired** (boolean *required*)

setSettings

public void **setSettings** (List<FieldSetting> *settings*)

setTooltip

public void **setTooltip** (String *tooltip*)

setTracked

public void **setTracked** (boolean *tracked*)

setType

public void **setType** (Type *type*)

setUIDisplayPosition

public void **setUIDisplayPosition** (Long *uiDisplayPosition*)

setUIDisplayable

public void **setUIDisplayable** (boolean *uiDisplayable*)

setUIFilterable

public void **setUIFilterable** (boolean *uiFilterable*)

setValidations

public void **setValidations** (List<FieldValidation> *validations*)

toDto

public FieldDto **toDto** ()

update

```
public Field update (FieldDto field)
```

11.34.11 FieldHolder

```
public class FieldHolder
```

The main purpose of this class is to give a easy way to access values inside metadata and settings related with the given field.

Constructors

FieldHolder

```
public FieldHolder (Field field)
```

FieldHolder

```
public FieldHolder (FieldDto field)
```

FieldHolder

```
protected FieldHolder (List<? extends Pair<String, String>> metadata, List<? extends Pair<String, ?>> settings)
```

Methods

getMetadata

```
public String getMetadata (String name)
```

getMetadata

```
public String getMetadata (String name, String defaultValue)
```

getSetting

```
public String getSetting (String name)
```

getSetting

```
public String getSetting (String name, String defaultValue)
```

`getSettingAsArray`

```
public String[] getSettingAsArray (String name)
```

`getSettingAsBoolean`

```
public boolean getSettingAsBoolean (String name)
```

11.34.12 FieldMetadata

```
public class FieldMetadata implements Pair<String, String>
```

The `FieldMetadata` class contains information about a single metadata added into a field.

Constructors

FieldMetadata

```
public FieldMetadata ()
```

FieldMetadata

```
public FieldMetadata (Field field, String key)
```

FieldMetadata

```
public FieldMetadata (MetadataDto metadata)
```

FieldMetadata

```
public FieldMetadata (Field field, String key, String value)
```

Methods

copy

```
public FieldMetadata copy ()
```

getField

```
public Field getField ()
```

getId

```
public Long getId ()
```

getKey

```
public String getKey ()
```

getValue

```
public String getValue ()
```

setField

```
public void setField (Field field)
```

setId

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

setKey

```
public void setKey (String key)
```

setValue

```
public void setValue (String value)
```

toDto

```
public MetadataDto toDto ()
```

update

```
public final void update (MetadataDto metadata)
```

11.34.13 FieldSetting

```
public class FieldSetting implements Pair<String, String>
```

Constructors**FieldSetting**

```
public FieldSetting ()
```

FieldSetting

```
public FieldSetting (Field field, TypeSetting details)
```

FieldSetting

public **FieldSetting** (*Field field*, *TypeSetting details*, *String value*)

Methods

copy

public *FieldSetting* **copy** ()

getDetails

public *TypeSetting* **getDetails** ()

getField

public *Field* **getField** ()

getId

public *Long* **getId** ()

getKey

public *String* **getKey** ()

getValue

public *String* **getValue** ()

setDetails

public void **setDetails** (*TypeSetting details*)

setField

public void **setField** (*Field field*)

setId

public void **setId** (*Long id*)

setValue

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

toDto

```
public SettingDto toDto ()
```

11.34.14 FieldValidation

```
public class FieldValidation
```

The `FieldValidation` class contains the value that is related with the correct type validation and information about that whether the given validation is enabled or not.

Constructors

FieldValidation

```
public FieldValidation ()
```

FieldValidation

```
public FieldValidation (Field field, TypeValidation details)
```

FieldValidation

```
public FieldValidation (Field field, TypeValidation details, String value, boolean enabled)
```

Methods

copy

```
public FieldValidation copy ()
```

getDetails

```
public TypeValidation getDetails ()
```

getField

```
public Field getField ()
```

getId

```
public Long getId ()
```

getValue

```
public String getValue ()
```

isEnabled

public boolean **isEnabled** ()

setDetails

public void **setDetails** ([TypeValidation](#) *details*)

setEnabled

public void **setEnabled** (boolean *enabled*)

setField

public void **setField** ([Field](#) *field*)

setId

public void **setId** ([Long](#) *id*)

setValue

public void **setValue** ([String](#) *value*)

toDto

public [ValidationCriterionDto](#) **toDto** ()

11.34.15 Lookup

public class **Lookup**

The `Lookup` class contains information about single lookup

Constructors

Lookup

public **Lookup** ()

Lookup

public **Lookup** ([String](#) *lookupName*, boolean *singleObjectReturn*, boolean *exposedViaRest*, [List](#)<[Field](#)> *fields*)

Lookup

```
public Lookup (String lookupName, boolean singleObjectReturn, boolean exposedViaRest, List<Field> fields,
               boolean readOnly, String methodName)
```

Lookup

```
public Lookup (String lookupName, boolean singleObjectReturn, boolean exposedViaRest, List<Field> fields,
               boolean readOnly, String methodName, List<String> rangeLookupFields, List<String> set-
               LookupFields, Map<String, String> customOperators)
```

Lookup

```
public Lookup (String lookupName, boolean singleObjectReturn, boolean exposedViaRest, List<Field> fields,
               boolean readOnly, String methodName, List<String> rangeLookupFields, List<String> set-
               LookupFields, Map<String, String> customOperators, Map<String, Boolean> useGeneric-
               Params)
```

Lookup

```
public Lookup (String lookupName, boolean singleObjectReturn, boolean exposedViaRest, List<Field> fields,
               Entity entity)
```

Lookup

```
public Lookup (LookupDto lookupDto, List<Field> lookupFields)
```

Methods

copy

```
public Lookup copy (List<Field> fields)
```

getCustomOperators

```
public Map<String, String> getCustomOperators ()
```

getEntity

```
public Entity getEntity ()
```

getFields

```
public List<Field> getFields ()
```

getId

public Long **getId** ()

getLookupFieldById

public final Field **getLookupFieldById** (Long *id*)

getLookupFieldByName

public final Field **getLookupFieldByName** (String *name*)

getLookupName

public String **getLookupName** ()

getMethodName

public String **getMethodName** ()

getRangeLookupFields

public final List<String> **getRangeLookupFields** ()

getSetLookupFields

public final List<String> **getSetLookupFields** ()

getUseGenericParams

public Map<String, Boolean> **getUseGenericParams** ()

isExposedViaRest

public boolean **isExposedViaRest** ()

isRangeParam

public boolean **isRangeParam** (Field *field*)

isReadOnly

public boolean **isReadOnly** ()

isSetParam

public boolean **isSetParam** ([Field](#) *field*)

isSingleObjectReturn

public boolean **isSingleObjectReturn** ()

setCustomOperators

public void **setCustomOperators** ([Map](#)<[String](#), [String](#)> *customOperators*)

setEntity

public void **setEntity** ([Entity](#) *entity*)

setExposedViaRest

public void **setExposedViaRest** (boolean *exposedViaRest*)

setFields

public void **setFields** ([List](#)<[Field](#)> *fields*)

setId

public void **setId** ([Long](#) *id*)

setLookupName

public final void **setLookupName** ([String](#) *lookupName*)

setMethodName

public void **setMethodName** ([String](#) *methodName*)

setRangeLookupFields

public void **setRangeLookupFields** ([List](#)<[String](#)> *rangeLookupFields*)

setReadOnly

public void **setReadOnly** (boolean *readOnly*)

setSetLookupFields

```
public void setSetLookupFields (List<String> setLookupFields)
```

setSingleObjectReturn

```
public void setSingleObjectReturn (boolean singleObjectReturn)
```

setUseGenericParams

```
public void setUseGenericParams (Map<String, Boolean> useGenericParams)
```

toDto

```
public LookupDto toDto ()
```

update

```
public final void update (LookupDto lookupDto, List<Field> lookupFields)
```

11.34.16 ManyToManyRelationship

```
public class ManyToManyRelationship extends Relationship
```

A specialization of the `Relationship` class. Represents a many-to-many relationship.

Methods**getFieldType**

```
public String getFieldType (Field field, EntityType type)
```

getGenericSignature

```
public String getGenericSignature (Field field, EntityType type)
```

11.34.17 ManyToOneRelationship

```
public class ManyToOneRelationship extends Relationship
```

A specialization of the `org.motechproject.mds.domain.Relationship` class. Represents a many-to-one relationship.

Methods**getFieldType**

```
public String getFieldType (Field field, EntityType type)
```

getGenericSignature

```
public String getGenericSignature (Field field, EntityType type)
```

11.34.18 MdsEntity

```
public abstract class MdsEntity
```

The `MdsEntity` is an optional class for all domain classes acting as Motech data services Entities. This class stores and allows access to all default fields like id, creator or modification date. All classes annotated `org.motechproject.mds.annotations.Entity` can extend this base class.

Methods**getCreationDate**

```
public DateTime getCreationDate ()
```

getCreator

```
public String getCreator ()
```

getId

```
public Long getId ()
```

getModificationDate

```
public DateTime getModificationDate ()
```

getModifiedBy

```
public String getModifiedBy ()
```

getOwner

```
public String getOwner ()
```

setCreationDate

```
public void setCreationDate (DateTime creationDate)
```

setCreator

```
public void setCreator (String creator)
```

setId

public void **setId** (*Long id*)

setModificationDate

public void **setModificationDate** (*DateTime modificationDate*)

setModifiedBy

public void **setModifiedBy** (*String modifiedBy*)

setOwner

public void **setOwner** (*String owner*)

11.34.19 OneToManyRelationship

public class **OneToManyRelationship** extends *Relationship*

A specialization of the *Relationship* class. Represents a one-to-many relationship.

Methods

getFieldType

public *String* **getFieldType** (*Field field*, *EntityType type*)

getGenericSignature

public *String* **getGenericSignature** (*Field field*, *EntityType type*)

11.34.20 OneToOneRelationship

public class **OneToOneRelationship** extends *Relationship*

A specialization of the *Relationship* class. Represents a one-to-one relationship.

Methods

getFieldType

public *String* **getFieldType** (*Field field*, *EntityType type*)

getGenericSignature

public *String* **getGenericSignature** (*Field field*, *EntityType type*)

11.34.21 Relationship

public class **Relationship**

A class representing a relationship type. This class is inherited by different types of relationships. This class only represents the field type and provides some utility methods. It is not used in entities themselves.

Methods

getFieldType

public **String** **getFieldType** (**Field** *field*, **EntityType** *type*)

getGenericSignature

public **String** **getGenericSignature** (**Field** *field*, **EntityType** *type*)

getRelatedClassName

protected **String** **getRelatedClassName** (**Field** *field*, **EntityType** *type*)

11.34.22 RelationshipHolder

public class **RelationshipHolder** extends **FieldHolder**

The main purpose of this class is to find out how cascade should be used for the given field with relationship type.

Constructors

RelationshipHolder

public **RelationshipHolder** (**Field** *field*)

RelationshipHolder

public **RelationshipHolder** (**ClassData** *data*, **Field** *field*)

Methods

getRelatedClass

public **String** **getRelatedClass** ()

getRelatedField

public **String** **getRelatedField** ()

isCascadeDelete

public boolean **isCascadeDelete** ()

isCascadePersist

public boolean **isCascadePersist** ()

isCascadeUpdate

public boolean **isCascadeUpdate** ()

isManyToMany

public boolean **isManyToMany** ()

isOneToMany

public boolean **isOneToMany** ()

isOneToOne

public boolean **isOneToOne** ()

isOwningSide

public boolean **isOwningSide** ()

11.34.23 RestOperation

public enum **RestOperation**

The `RestOperation` enum represents CRUD operations that can be enabled for entities.

See also: `org.motechproject.mds.annotations.RestOperations`

Enum Constants

ALL

public static final `RestOperation` **ALL**

CREATE

public static final `RestOperation` **CREATE**

DELETE

```
public static final RestOperation DELETE
```

READ

```
public static final RestOperation READ
```

UPDATE

```
public static final RestOperation UPDATE
```

11.34.24 RestOptions

```
public class RestOptions
```

The `RestOptions` class representing rest options of given entity. This class is related with table in database with the same name.

Constructors

RestOptions

```
public RestOptions ()
```

RestOptions

```
public RestOptions (Entity entity)
```

Methods

copy

```
public RestOptions copy ()
```

getEntity

```
public Entity getEntity ()
```

getFields

```
public List<Field> getFields ()
```

getId

```
public Long getId ()
```

getLookups

public [List](#)<[Lookup](#)> **getLookups** ()

isAllowCreate

public boolean **isAllowCreate** ()

isAllowDelete

public boolean **isAllowDelete** ()

isAllowRead

public boolean **isAllowRead** ()

isAllowUpdate

public boolean **isAllowUpdate** ()

setAllowCreate

public void **setAllowCreate** (boolean *allowCreate*)

setAllowDelete

public void **setAllowDelete** (boolean *allowDelete*)

setAllowRead

public void **setAllowRead** (boolean *allowRead*)

setAllowUpdate

public void **setAllowUpdate** (boolean *allowUpdate*)

setEntity

public void **setEntity** ([Entity](#) *entity*)

setId

public void **setId** ([Long](#) *id*)

supportsAnyOperation

```
public boolean supportsAnyOperation ()
```

toDto

```
public RestOptionsDto toDto ()
```

update

```
public final void update (RestOptionsDto restOptionsDto)
```

11.34.25 Tracking

```
public class Tracking
```

The `Tracking` contains information about which fields and what kind of actions should be logged. This class is related with table in database with the same name.

Constructors**Tracking**

```
public Tracking ()
```

Tracking

```
public Tracking (Entity entity)
```

Methods**copy**

```
public Tracking copy ()
```

getEntity

```
public Entity getEntity ()
```

getFields

```
public List<Field> getFields ()
```

getId

```
public Long getId ()
```

isAllowCreate

public boolean **isAllowCreate** ()

isAllowDelete

public boolean **isAllowDelete** ()

isAllowRead

public boolean **isAllowRead** ()

isAllowUpdate

public boolean **isAllowUpdate** ()

setAllowCreate

public void **setAllowCreate** (boolean *allowCreate*)

setAllowDelete

public void **setAllowDelete** (boolean *allowDelete*)

setAllowRead

public void **setAllowRead** (boolean *allowRead*)

setAllowUpdate

public void **setAllowUpdate** (boolean *allowUpdate*)

setEntity

public void **setEntity** ([Entity](#) *entity*)

setId

public void **setId** ([Long](#) *id*)

toDto

public [TrackingDto](#) **toDto** ()

11.34.26 Type

public class **Type**

The `Type` class contains information about a single type in mds system. The mds type can have a settings and validations that can be assigned to field with the same type.

Constructors

Type

public **Type** ()

Type

public **Type** (*Class*<?> *typeClass*)

Type

public **Type** (*String displayName*, *String description*, *Class*<?> *typeClass*)

Methods

getDefaultName

public *String* **getDefaultName** ()

getDescription

public *String* **getDescription** ()

getDisplayName

public *String* **getDisplayName** ()

getId

public *Long* **getId** ()

getSettings

public *List*<*TypeSetting*> **getSettings** ()

getTypeClass

public *Class*<?> **getTypeClass** ()

getTypeClassName

public **String** **getTypeClassName** ()

getValidations

public **List**<**TypeValidation**> **getValidations** ()

hasSettings

public boolean **hasSettings** ()

hasValidation

public boolean **hasValidation** ()

isBlob

public boolean **isBlob** ()

isCombobox

public boolean **isCombobox** ()

isRelationship

public boolean **isRelationship** ()

parse

public **Object** **parse** (**String** *str*)

setDefaultName

public void **setDefaultName** (**String** *defaultName*)

setDescription

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

setDisplayName

public void **setDisplayName** (**String** *displayName*)

setId

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

setSettings

```
public void setSettings (List<TypeSetting> settings)
```

setTypeClass

```
public void setTypeClass (Class<?> typeClass)
```

setValidations

```
public void setValidations (List<TypeValidation> validations)
```

toDto

```
public TypeDto toDto ()
```

11.34.27 TypeSetting

```
public class TypeSetting
```

The `TypeSetting` contains settings for the given mds type. This class is related with table in database with the same name.

Constructors**TypeSetting**

```
public TypeSetting ()
```

TypeSetting

```
public TypeSetting (String name)
```

Methods**getDefaultValue**

```
public String getDefaultValue ()
```

getId

```
public Long getId ()
```

getName

```
public String getName ()
```

getTypeSettingOptions

```
public List<TypeSettingOption> getTypeSettingOptions ()
```

getValueType

```
public Type getValueType ()
```

setDefaultValue

```
public void setDefaultValue (String defaultValue)
```

setId

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

setName

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

setTypeSettingOptions

```
public void setTypeSettingOptions (List<TypeSettingOption> typeSettingOptions)
```

setValueType

```
public void setValueType (Type valueType)
```

11.34.28 TypeSettingOption

```
public class TypeSettingOption
```

The `TypeSettingOption` contains a single setting option for the given type setting. This class is related with table in database with the same name.

Constructors

TypeSettingOption

```
public TypeSettingOption (String name)
```

Methods

getId

```
public Long getId ()
```

getName

```
public String getName ()
```

setId

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

setName

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

11.34.29 TypeValidation

```
public class TypeValidation
```

The `TypeValidation` contains a single validation option for the given type. This class is related with table in database with the same name.

Constructors

TypeValidation

```
public TypeValidation ()
```

TypeValidation

```
public TypeValidation (String displayName, Type valueType)
```

Methods

getAnnotations

```
public List<Class<? extends Annotation>> getAnnotations ()
```

getDisplayName

```
public String getDisplayName ()
```

getId

public Long **getId** ()

getValueType

public Type **getValueType** ()

setAnnotations

public void **setAnnotations** (List<Class<? extends Annotation>> *annotations*)

setDisplayName

public void **setDisplayName** (String *displayName*)

setId

public void **setId** (Long *id*)

setValueType

public void **setValueType** (Type *valueType*)

toString

public String **toString** ()

11.35 org.motechproject.mds.dto

11.35.1 AdvancedSettingsDto

public class **AdvancedSettingsDto**

The AdvancedSettingsDto contains information about advanced settings of an entity.

Methods

addNewIndex

public void **addNewIndex** (String *lookupName*)

equals

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getBrowsing

```
public BrowsingSettingsDto getBrowsing ()
```

getEntityId

```
public Long getEntityId ()
```

getId

```
public Long getId ()
```

getIndexes

```
public List<LookupDto> getIndexes ()
```

getRestOptions

```
public RestOptionsDto getRestOptions ()
```

getTracking

```
public TrackingDto getTracking ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

removeIndex

```
public void removeIndex (Integer idx)
```

setBrowsing

```
public void setBrowsing (BrowsingSettingsDto browsing)
```

setEntityId

```
public void setEntityId (Long entityId)
```

setId

public void **setId** ([Long id](#))

setIndexes

public void **setIndexes** ([List<LookupDto> indexes](#))

setRestOptions

public void **setRestOptions** ([RestOptionsDto restOptions](#))

setTracking

public void **setTracking** ([TrackingDto tracking](#))

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.2 BrowsingSettingsDto

public class **BrowsingSettingsDto**

The **BrowsingSettingsDto** contains informations about filed browsing settings

Methods

addDisplayedField

public void **addDisplayedField** ([Number id](#))

addFilterableField

public void **addFilterableField** ([Number id](#))

containsDisplayedField

public boolean **containsDisplayedField** ([Long number](#))

containsFilterableField

public boolean **containsFilterableField** ([Number id](#))

equals

```
public boolean equals (Object obj)
    { @inheritDoc }
```

getDisplayedFields

```
public List<Number> getDisplayedFields ()
```

getFilterableFields

```
public List<Number> getFilterableFields ()
```

hashCode

```
public int hashCode ()
    { @inheritDoc }
```

indexOfDisplayedField

```
public long indexOfDisplayedField (Long id)
```

removeFilterableField

```
public void removeFilterableField (Number id)
```

setDisplayedFields

```
public void setDisplayedFields (List<Number> displayedFields)
```

setFilterableFields

```
public void setFilterableFields (List<Number> filterableFields)
```

toString

```
public String toString ()
    { @inheritDoc }
```

11.35.3 DraftData

```
public class DraftData
```

The `DraftData` contains information that are used later for creating temporary changes in field.

Fields

ADD_NEW_INDEX

public static final [String](#) **ADD_NEW_INDEX**

ADVANCED

public static final [String](#) **ADVANCED**

DISPLAY_NAME

public static final [String](#) **DISPLAY_NAME**

FIELD

public static final [String](#) **FIELD**

FIELD_ID

public static final [String](#) **FIELD_ID**

NAME

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

PATH

public static final [String](#) **PATH**

REMOVE_INDEX

public static final [String](#) **REMOVE_INDEX**

SECURITY

public static final [String](#) **SECURITY**

TYPE_CLASS

public static final [String](#) **TYPE_CLASS**

VALUE

public static final [String](#) **VALUE**

Methods

getPath

public `String` **getPath** ()

getValue

public `Object` **getValue** (`String` *key*)

getValues

public `Map<String, Object>` **getValues** ()

isCreate

public boolean **isCreate** ()

isEdit

public boolean **isEdit** ()

isForAdvanced

public boolean **isForAdvanced** ()

isForField

public boolean **isForField** ()

isForSecurity

public boolean **isForSecurity** ()

isRemove

public boolean **isRemove** ()

setCreate

public void **setCreate** (boolean *create*)

setEdit

public void **setEdit** (boolean *edit*)

setRemove

public void **setRemove** (boolean *remove*)

setValues

public void **setValues** (Map<String, Object> *values*)

11.35.4 DraftResult

public class **DraftResult** implements [Serializable](#)

After users do draft changes an instance of this class is returned. It contains information about the draft state.

Constructors

DraftResult

public **DraftResult** (boolean *changesMade*, boolean *outdated*)

Methods

isChangesMade

public boolean **isChangesMade** ()

isOutdated

public boolean **isOutdated** ()

setChangesMade

public void **setChangesMade** (boolean *changesMade*)

setOutdated

public void **setOutdated** (boolean *outdated*)

11.35.5 DtoHelper

public final class **DtoHelper**

Utility class for managing dto collections.

Methods

asFieldMapById

```
public static Map<Long, FieldDto> asFieldMapById (Collection<FieldDto> fields)
```

Stores fields in a map using id as the key for faster lookup

Parameters

- **fields** – the field collection

Returns a map with field ids being the keys and fields being the values

asFieldMapByName

```
public static Map<String, FieldDto> asFieldMapByName (Collection<FieldDto> fields)
```

Stores fields in a map using name as the key for faster lookup

Parameters

- **fields** – the field collection

Returns a map with field names being the keys and fields being the values

11.35.6 EntityDto

```
public class EntityDto
```

The `EntityDto` class contains only basic information about an entity like id, name, module and namespace.

Constructors

EntityDto

```
public EntityDto ()
```

EntityDto

```
public EntityDto (String className)
```

EntityDto

```
public EntityDto (Long id, String className)
```

EntityDto

```
public EntityDto (String className, SecurityMode securityMode, Set<String> securityMembers)
```

EntityDto

```
public EntityDto (Long id, String className, SecurityMode securityMode, Set<String> securityMembers)
```

EntityDto

```
public EntityDto (Long id, String className, String module, SecurityMode securityMode, Set<String> securityMembers)
```

EntityDto

```
public EntityDto (Long id, String className, String module, String namespace, SecurityMode securityMode, Set<String> securityMembers)
```

EntityDto

```
public EntityDto (String className, String name, String module, String namespace, SecurityMode securityMode, Set<String> securityMembers)
```

EntityDto

```
public EntityDto (Long id, String className, String name, String module, String namespace, SecurityMode securityMode, Set<String> securityMembers)
```

EntityDto

```
public EntityDto (Long id, String className, String name, String module, String namespace, SecurityMode securityMode, Set<String> securityMembers, String superClass)
```

EntityDto

```
public EntityDto (Long id, String className, String name, String module, String namespace, SecurityMode securityMode, Set<String> securityMembers, String superClass, boolean abstractClass)
```

Methods

equals

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getClassName

```
public String getClassName ()
```

getId

```
public Long getId ()
```


getModule

```
public String getModule ()
```

getName

```
public String getName ()
```

getNamespace

```
public String getNamespace ()
```

getSecurityMembers

```
public Set<String> getSecurityMembers ()
```

getSecurityMode

```
public SecurityMode getSecurityMode ()
```

getSuperClass

```
public String getSuperClass ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

isAbstractClass

```
public boolean isAbstractClass ()
```

isDDE

```
public boolean isDDE ()
```

isModified

```
public boolean isModified ()
```

isOutdated

```
public boolean isOutdated ()
```

isReadOnly

public boolean **isReadOnly** ()

setAbstractClass

public void **setAbstractClass** (boolean *abstractClass*)

setClassName

public void **setClassName** (String *className*)

setId

public void **setId** (Long *id*)

setModified

public void **setModified** (boolean *modified*)

setModule

public void **setModule** (String *module*)

setName

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

setNamespace

public void **setNamespace** (String *namespace*)

setOutdated

public void **setOutdated** (boolean *outdated*)

setReadOnly

public void **setReadOnly** (boolean *readOnly*)

setSecurityMembers

public void **setSecurityMembers** (Set<String> *securityMembers*)

setSecurityMode

```
public void setSecurityMode (SecurityMode securityMode)
```

setSuperClass

```
public void setSuperClass (String superClass)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.7 FieldBasicDto

```
public class FieldBasicDto
```

The FieldBasicDto contains basic information about a field.

Constructors**FieldBasicDto**

```
public FieldBasicDto ()
```

FieldBasicDto

```
public FieldBasicDto (String displayName, String name)
```

FieldBasicDto

```
public FieldBasicDto (String displayName, String name, boolean required, Object defaultValue, String tooltip)
```

Methods**equals**

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getDefaultValue

```
public Object getDefaultValue ()
```

getDisplayName

```
public String getDisplayName ()
```

getName

```
public String getName ()
```

getTooltip

```
public String getTooltip ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

isRequired

```
public boolean isRequired ()
```

setDefaultValue

```
public void setDefaultValue (Object defaultValue)
```

setDisplayName

```
public void setDisplayName (String displayName)
```

setName

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

setRequired

```
public void setRequired (boolean required)
```

setTooltip

```
public void setTooltip (String tooltip)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.8 FieldDto

public class **FieldDto**

The `FieldDto` class contains information about an existing field in an entity.

Constructors

FieldDto

public **FieldDto** ()

FieldDto

public **FieldDto** (Long *id*, Long *entityId*, TypeDto *type*, FieldBasicDto *basic*, boolean *readOnly*, List<MetadataDto> *metadata*, FieldValidationDto *validation*, List<SettingDto> *settings*, List<LookupDto> *lookups*)

FieldDto

public **FieldDto** (Long *id*, Long *entityId*, TypeDto *type*, FieldBasicDto *basic*, boolean *readOnly*, FieldValidationDto *validation*)

Methods

addEmptyMetadata

public void **addEmptyMetadata** ()

addMetadata

public void **addMetadata** (MetadataDto *metadata*)

equals

public boolean **equals** (Object *obj*)
 {@inheritDoc}

getBasic

public FieldBasicDto **getBasic** ()

getEntityId

public Long **getEntityId** ()

getId

public Long **getId** ()

getLookups

public List<LookupDto> **getLookups** ()

getMetadata

public List<MetadataDto> **getMetadata** ()

getMetadata

public MetadataDto **getMetadata** (String *key*)

getSetting

public SettingDto **getSetting** (String *name*)

getSettings

public List<SettingDto> **getSettings** ()

getType

public TypeDto **getType** ()

getValidation

public FieldValidationDto **getValidation** ()

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

isReadOnly

public boolean **isReadOnly** ()

removeMetadata

public void **removeMetadata** (Integer *idx*)

setBasic

```
public void setBasic (FieldBasicDto basic)
```

setEntityId

```
public void setEntityId (Long entityId)
```

setId

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

setLookups

```
public void setLookups (List<LookupDto> lookups)
```

setMetadata

```
public void setMetadata (List<MetadataDto> metadata)
```

setReadOnly

```
public void setReadOnly (boolean readOnly)
```

setSettings

```
public void setSettings (List<SettingDto> settings)
```

setType

```
public void setType (TypeDto type)
```

setValidation

```
public void setValidation (FieldValidationDto validation)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.9 FieldInstanceDto

```
public class FieldInstanceDto
```

The `FieldInstanceDto` class contains information about an existing field in an instance.

Constructors

FieldInstanceDto

```
public FieldInstanceDto ()
```

FieldInstanceDto

```
public FieldInstanceDto (Long id, Long instanceId, FieldBasicDto basic)
```

Methods

equals

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getBasic

```
public FieldBasicDto getBasic ()
```

getId

```
public Long getId ()
```

getInstanceId

```
public Long getInstanceId ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

setBasic

```
public void setBasic (FieldBasicDto basic)
```

setId

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

setInstanceId

```
public void setInstanceId (Long instanceId)
```


toString

```
public String toString()  
    { @inheritDoc }
```

11.35.10 FieldValidationDto

```
public class FieldValidationDto
```

The `FieldValidationDto` class contains information about validation criteria for field.

Fields

DOUBLE

```
public static final FieldValidationDto DOUBLE
```

Constant `DOUBLE` contains validation criteria for double type.

INTEGER

```
public static final FieldValidationDto INTEGER
```

Constant `INTEGER` contains validation criteria for integer type.

STRING

```
public static final FieldValidationDto STRING
```

Constant `STRING` contains validation criteria for string type.

Constructors

FieldValidationDto

```
public FieldValidationDto()
```

FieldValidationDto

```
public FieldValidationDto(ValidationCriterionDto... criteria)
```

Methods

addCriterion

```
public void addCriterion(ValidationCriterionDto criterion)
```

equals

```
public boolean equals (Object obj)
    { @inheritDoc }
```

getCriteria

```
public List<ValidationCriterionDto> getCriteria ()
```

getCriterion

```
public ValidationCriterionDto getCriterion (String displayName)
```

hashCode

```
public int hashCode ()
    { @inheritDoc }
```

setCriteria

```
public void setCriteria (List<ValidationCriterionDto> criteria)
```

toString

```
public String toString ()
    { @inheritDoc }
```

11.35.11 LookupDto

```
public class LookupDto
```

The `LookupDto` class contains information about single lookup defined by user

Constructors**LookupDto**

```
public LookupDto ()
```

LookupDto

```
public LookupDto (String lookupName, boolean singleObjectReturn, boolean exposedViaRest)
```

LookupDto

```
public LookupDto (String lookupName, boolean singleObjectReturn, boolean exposedViaRest,
    List<LookupFieldDto> lookupFields, boolean readOnly)
```

LookupDto

```
public LookupDto (String lookupName, boolean singleObjectReturn, boolean exposedViaRest,  
                 List<LookupFieldDto> lookupFields, boolean readOnly, String methodName)
```

LookupDto

```
public LookupDto (Long id, String lookupName, boolean singleObjectReturn, boolean exposedViaRest,  
                 List<LookupFieldDto> lookupFields, boolean readOnly, String methodName)
```

Methods

addField

```
public void addField (Long field)
```

addField

```
public void addField (Integer field)
```

equals

```
public boolean equals (Object o)  
    { @inheritDoc }
```

getId

```
public Long getId ()
```

getLookupFields

```
public final List<LookupFieldDto> getLookupFields ()
```

getLookupName

```
public String getLookupName ()
```

getMethodName

```
public String getMethodName ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

insertField

public void **insertField** (*Integer idx*, *Integer fieldId*)

insertField

public void **insertField** (*Integer idx*, *Long fieldId*)

isExposedViaRest

public boolean **isExposedViaRest** ()

isReadOnly

public boolean **isReadOnly** ()

isReferenced

public boolean **isReferenced** ()

isSingleObjectReturn

public boolean **isSingleObjectReturn** ()

removeField

public void **removeField** (*Long fieldId*)

removeField

public void **removeField** (*Integer fieldId*)

setExposedViaRest

public void **setExposedViaRest** (boolean *isExposedViaRest*)

setId

public void **setId** (*Long id*)

setLookupFields

public void **setLookupFields** (*List<LookupFieldDto> lookupFields*)

setLookupName

```
public void setLookupName (String lookupName)
```

setMethodName

```
public void setMethodName (String methodName)
```

setReadOnly

```
public void setReadOnly (boolean readOnly)
```

setReferenced

```
public void setReferenced (boolean referenced)
```

setSingleObjectReturn

```
public void setSingleObjectReturn (boolean singleObjectReturn)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.12 LookupFieldDto

```
public class LookupFieldDto
```

Represents a field added to a lookup. The lookup using a given field can be done using multiple lookup types.

Constructors**LookupFieldDto**

```
public LookupFieldDto ()
```

LookupFieldDto

```
public LookupFieldDto (Long id, String name, Type type)
```

LookupFieldDto

```
public LookupFieldDto (Long id, String name, Type type, String customOperator)
```

LookupFieldDto

```
public LookupFieldDto (Long id, String name, Type type, String customOperator, boolean useGeneric-  
Param)
```

Methods

equals

```
public boolean equals (Object o)
```

getCustomOperator

```
public String getCustomOperator ()
```

getId

```
public Long getId ()
```

getName

```
public String getName ()
```

getType

```
public Type getType ()
```

hashCode

```
public int hashCode ()
```

isUseGenericParam

```
public boolean isUseGenericParam ()
```

setCustomOperator

```
public void setCustomOperator (String customOperator)
```

setId

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

setName

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

setType

```
public void setType (Type type)
```

setUseGenericParam

```
public void setUseGenericParam (boolean useGenericParam)
```

11.35.13 LookupFieldDto.Type

```
public static enum Type
```

The lookup type represents whether the lookup will be done by comparing to a single field, matching values to a range, or matching to a set of values.

Enum Constants**RANGE**

```
public static final LookupFieldDto.Type RANGE
```

SET

```
public static final LookupFieldDto.Type SET
```

VALUE

```
public static final LookupFieldDto.Type VALUE
```

11.35.14 MetadataDto

```
public class MetadataDto implements Pair<String, String>
```

The `MetadataDto` contains key and value of a single field metadata.

Constructors**MetadataDto**

```
public MetadataDto ()
```

MetadataDto

```
public MetadataDto (String key, String value)
```

MetadataDto

```
public MetadataDto (Long id, String key, String value)
```

Methods

equals

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getId

```
public Long getId ()
```

getKey

```
public String getKey ()
```

getValue

```
public String getValue ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

setId

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

setKey

```
public void setKey (String key)
```

setValue

```
public void setValue (String value)
```


toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.15 RestOptionsDto

```
public class RestOptionsDto  
    Class representing rest options of given entity.
```

Constructors**RestOptionsDto**

```
public RestOptionsDto ()
```

RestOptionsDto

```
public RestOptionsDto (boolean create, boolean read, boolean update, boolean delete)
```

Methods**addField**

```
public void addField (Number id)
```

addLookup

```
public void addLookup (Number id)
```

containsFieldId

```
public boolean containsFieldId (Number id)
```

containsLookupId

```
public boolean containsLookupId (Number id)
```

equals

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getFieldIds

```
public List<Number> getFieldIds ()
```

getId

```
public Long getId ()
```

getLookupIds

```
public List<Number> getLookupIds ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

isCreate

```
public boolean isCreate ()
```

isDelete

```
public boolean isDelete ()
```

isRead

```
public boolean isRead ()
```

isUpdate

```
public boolean isUpdate ()
```

removeField

```
public void removeField (Number id)
```

removeLookup

```
public void removeLookup (Number id)
```

setCreate

```
public void setCreate (boolean create)
```

setDelete

public void **setDelete** (boolean *delete*)

setFieldIds

public void **setFieldIds** (List<Number> *fieldIds*)

setId

public void **setId** (Long *id*)

setLookupIds

public void **setLookupIds** (List<Number> *lookupIds*)

setRead

public void **setRead** (boolean *read*)

setUpdate

public void **setUpdate** (boolean *update*)

supportAnyOperation

public boolean **supportAnyOperation** ()

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.16 SettingDto

public class **SettingDto** implements Pair<String, Object>

The **SettingDto** contains information about a single setting inside a field.

Constructors**SettingDto**

public **SettingDto** ()

SettingDto

```
public SettingDto (String name, Object value)
```

SettingDto

```
public SettingDto (String name, Object value, TypeDto type, SettingOptions... options)
```

Methods

copy

```
public SettingDto copy ()
```

equals

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getKey

```
public String getKey ()
```

getName

```
public String getName ()
```

getOptions

```
public List<SettingOptions> getOptions ()
```

getType

```
public TypeDto getType ()
```

getValue

```
public Object getValue ()
```

getValueAsString

```
public String getValueAsString ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

setName

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

setOptions

```
public void setOptions (List<SettingOptions> options)
```

setType

```
public void setType (TypeDto type)
```

setValue

```
public void setValue (Object value)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.17 SettingOptions

```
public enum SettingOptions
```

The `SettingOptions` contains available options that can be added to field setting.

Enum Constants**POSITIVE**

```
public static final SettingOptions POSITIVE
```

Ensure that a value in a given setting is a number and it has a positive value.

REQUIRE

```
public static final SettingOptions REQUIRE
```

Force setting a value for a given setting.

11.35.18 TrackingDto

public class **TrackingDto**

The `TrackingDto` contains information about which fields and what kind of actions should be logged.

Methods

addAction

public void **addAction** (*String action*)

addField

public void **addField** (*Number fieldId*)

equals

public boolean **equals** (*Object obj*)
{ @inheritDoc }

getActions

public *List<String>* **getActions** ()

getFields

public *List<Long>* **getFields** ()

hashCode

public int **hashCode** ()
{ @inheritDoc }

isAllowCreate

public boolean **isAllowCreate** ()

isAllowDelete

public boolean **isAllowDelete** ()

isAllowRead

public boolean **isAllowRead** ()

isAllowUpdate

```
public boolean isAllowUpdate ()
```

removeAction

```
public void removeAction (String action)
```

removeField

```
public void removeField (Number fieldId)
```

setActions

```
public void setActions (List<String> actions)
```

setFields

```
public void setFields (List<Long> fields)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.19 TypeDto

```
public class TypeDto
```

The `TypeDto` class contains information about an available field in an entity.

Fields**BLOB**

```
public static final TypeDto BLOB
```

Constant `BLOB` is a representation of the MDS BLOB type.

BOOLEAN

```
public static final TypeDto BOOLEAN
```

Constant `BOOLEAN` is a representation of the MDS Boolean type.

DATE

```
public static final TypeDto DATE
```

Constant `DATE` is a representation of the MDS Date type.

DATETIME

public static final [TypeDto](#) **DATETIME**

Constant **DATETIME** is a representation of the MDS DateTime type.

DOUBLE

public static final [TypeDto](#) **DOUBLE**

Constant **DOUBLE** is a representation of the MDS Decimal type.

INTEGER

public static final [TypeDto](#) **INTEGER**

Constant **INTEGER** is a representation of the MDS Integer type.

LIST

public static final [TypeDto](#) **LIST**

Constant **LIST** is a representation of the MDS Combobox type.

LOCAL_DATE

public static final [TypeDto](#) **LOCAL_DATE**

Constant **LOCAL_DATE** is a representation of the `org.joda.time.LocalDate` type.

LONG

public static final [TypeDto](#) **LONG**

Constant **LONG** is a representation of the MDS Long type.

MANY_TO_MANY_RELATIONSHIP

public static final [TypeDto](#) **MANY_TO_MANY_RELATIONSHIP**

MANY_TO_ONE_RELATIONSHIP

public static final [TypeDto](#) **MANY_TO_ONE_RELATIONSHIP**

MAP

public static final [TypeDto](#) **MAP**

Constant **MAP** is a representation of the MDS Map type.

ONE_TO_MANY_RELATIONSHIP

public static final [TypeDto](#) **ONE_TO_MANY_RELATIONSHIP**

ONE_TO_ONE_RELATIONSHIP

public static final [TypeDto](#) **ONE_TO_ONE_RELATIONSHIP**

PERIOD

public static final [TypeDto](#) **PERIOD**

Constant **PERIOD** is a representation of the MDS Period type.

STRING

public static final [TypeDto](#) **STRING**

Constant **STRING** is a representation of the MDS String type.

TIME

public static final [TypeDto](#) **TIME**

Constant **TIME** is a representation of the MDS Time type.

Constructors

[TypeDto](#)

public **TypeDto** ()

[TypeDto](#)

public **TypeDto** ([String](#) *displayName*, [String](#) *description*, [String](#) *defaultName*, [String](#) *typeClass*)

[TypeDto](#)

public **TypeDto** ([Long](#) *id*, [String](#) *displayName*, [String](#) *description*, [String](#) *defaultName*, [String](#) *typeClass*)

Methods

equals

public boolean **equals** ([Object](#) *obj*)
 { [@inheritDoc](#) }

getDefaultName

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

getDescription

```
public String getDescription ()
```

getDisplayName

```
public String getDisplayName ()
```

getId

```
public Long getId ()
```

getTypeClass

```
public String getTypeClass ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

isCombobox

```
public boolean isCombobox ()
```

isRelationship

```
public boolean isRelationship ()
```

setDefaultName

```
public void setDefaultName (String defaultName)
```

setDescription

```
public void setDescription (String description)
```

setDisplayName

```
public void setDisplayName (String displayName)
```

setId

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

setTypeClass

```
public void setTypeClass (String typeClass)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

11.35.20 ValidationCriterionDto

```
public class ValidationCriterionDto
```

The `ValidationCriterionDto` contains information about single criterion for field validation.

Constructors**ValidationCriterionDto**

```
public ValidationCriterionDto ()
```

ValidationCriterionDto

```
public ValidationCriterionDto (String displayName, TypeDto type)
```

ValidationCriterionDto

```
public ValidationCriterionDto (String displayName, TypeDto type, Object value, boolean enabled)
```

Methods**equals**

```
public boolean equals (Object obj)  
    { @inheritDoc }
```

getDisplayName

```
public String getDisplayName ()
```

getType

```
public TypeDto getType ()
```

getValue

```
public Object getValue ()
```

hashCode

```
public int hashCode ()  
    { @inheritDoc }
```

isEnabled

```
public boolean isEnabled ()
```

setDisplayName

```
public void setDisplayName (String displayName)
```

setEnabled

```
public void setEnabled (boolean enabled)
```

setType

```
public void setType (TypeDto type)
```

setValue

```
public void setValue (Object value)
```

toString

```
public String toString ()  
    { @inheritDoc }
```

valueAsString

```
public String valueAsString ()
```

11.36 org.motechproject.mds.enhancer

11.36.1 MdsJDOEnhancer

public class **MdsJDOEnhancer** extends JDOEnhancer

The `MdsJDOEnhancer` class is a wrapper for `org.datanucleus.api.jdo.JDOEnhancer` class. Its task is to add the missing information into created entity class.

Constructors

MdsJDOEnhancer

```
public MdsJDOEnhancer (Properties config, ClassLoader classLoader)
```

Methods

addClass

```
public void addClass (ClassData classData)
```

11.37 org.motechproject.mds.ex

11.37.1 CollectionResultFromLookupExpectedException

```
public class CollectionResultFromLookupExpectedException extends IllegalLookupReturnTypeException  
    Signals that a collection result was expected, but the lookup returned a single result.
```

Constructors

CollectionResultFromLookupExpectedException

```
public CollectionResultFromLookupExpectedException (String lookupName)
```

11.37.2 EmptyTrashException

```
public class EmptyTrashException extends MdsException  
    The EmptyTrashException exception signals a situation that there were some problems with cleaning the  
    module trash.
```

Constructors

EmptyTrashException

```
public EmptyTrashException (Throwable cause)  
    Constructs a new EmptyTrashException with mds.error.emptyTrashException as a message key.
```

Parameters

- **cause** – the cause of exception.

11.37.3 EntityAlreadyExistException

```
public class EntityAlreadyExistException extends MdsException  
    The EntityAlreadyExistException exception signals a situation in which a user wants to create a new  
    entity with a name that already exist in database.
```

Constructors

EntityAlreadyExistException

public **EntityAlreadyExistException** ()

Constructs a new EntityAlreadyExistException with *mds.error.entityAlreadyExist* as a message key.

11.37.4 EntityChangedException

public class **EntityChangedException** extends [MdsException](#)

This exception signals that an Entity was changed(presumably by another user).

Constructors

EntityChangedException

public **EntityChangedException** ()

11.37.5 EntityCreationException

public class **EntityCreationException** extends [MdsException](#)

The `EntityCreationException` exception signals a situation when there were problems with creating new entity class.

Constructors

EntityCreationException

public **EntityCreationException** ([Throwable](#) cause)

Constructs a new EntityCreationException with *mds.error.entityBuilderFailure* as a message key.

Parameters

- **cause** – the cause of exception.

EntityCreationException

public **EntityCreationException** ([String](#) messageKey)

11.37.6 EntityDeletedException

public class **EntityDeletedException** extends [MdsException](#)

This exception signals that the Entity was deleted(presumably by an another user).

Constructors

EntityDeletedException

```
public EntityDeletedException ()
```

11.37.7 EntityInfrastructureException

```
public class EntityInfrastructureException extends MdsException
```

The `EntityInfrastructureException` exception signals a situation when there were problems with creating repository/service interface/service class for entity.

Constructors

EntityInfrastructureException

```
public EntityInfrastructureException (Throwable cause)
```

Constructs a new `EntityInfrastructureException` with `mds.error.entityInfrastructureFailure` as a message key.

Parameters

- **cause** – the cause of exception.

11.37.8 EntityNotFoundException

```
public class EntityNotFoundException extends MdsException
```

The `EntityNotFoundException` exception signals a situation in which an entity with a given id does not exist in database.

Constructors

EntityNotFoundException

```
public EntityNotFoundException ()
```

Constructs a new `EntityNotFoundException` with `mds.error.entityNotFound` as a message key.

11.37.9 EntityReadOnlyException

```
public class EntityReadOnlyException extends MdsException
```

The `EntityReadOnlyException` exception signals a situation in which a user wants to make changes on an entity which is read only (it was created by a module).

Constructors

EntityReadOnlyException

```
public EntityReadOnlyException ()
```

Constructs a new `EntityReadOnlyException` with `mds.error.entityIsReadOnly` as a message key.

11.37.10 EntitySchemaMismatchException

public class **EntitySchemaMismatchException** extends [MdsException](#)

The `EntitySchemaMismatch` exception signals a situation in which a user wants to revert their instance to a version on a different schema version.

Constructors

EntitySchemaMismatchException

public **EntitySchemaMismatchException** ()

Constructs a new `EntitySchemaMismatch` with `mds.error.entitySchemaMismatch` as a message key.

11.37.11 FieldNotFoundException

public class **FieldNotFoundException** extends [MdsException](#)

This exception signals that a given field was not found for the Entity.

Constructors

FieldNotFoundException

public **FieldNotFoundException** ()

11.37.12 FieldUsedInLookupException

public class **FieldUsedInLookupException** extends [MdsException](#)

Exception indicating that a field cannot be removed, since it is used in a lookup.

Constructors

FieldUsedInLookupException

public **FieldUsedInLookupException** ([String](#) fieldName, [String](#) lookupNames)

11.37.13 IllegalLookupException

public class **IllegalLookupException** extends [RuntimeException](#)

Signales that the user defined an illegal lookup.

Constructors

IllegalLookupException

public **IllegalLookupException** ([String](#) message)

11.37.14 `IllegalLookupReturnTypeException`

public abstract class `IllegalLookupReturnTypeException` extends `IllegalArgumentException`
Signals that the exception returns one object but we expected a list(or vice-versa).

Constructors

`IllegalLookupReturnTypeException`

public `IllegalLookupReturnTypeException` (*String message*)

11.37.15 `LoaderException`

public class `LoaderException` extends `RuntimeException`
The `LoaderException` exception signals situations in which there were problems with correct loading the given class or its dependencies.

Constructors

`LoaderException`

public `LoaderException` (*Throwable cause*)

11.37.16 `LookupExecutionException`

public class `LookupExecutionException` extends `MdsException`
Signals that we were not able to execute a lookup for a given entity.

Constructors

`LookupExecutionException`

public `LookupExecutionException` ()

`LookupExecutionException`

public `LookupExecutionException` (*Throwable cause*)

11.37.17 `LookupExecutorException`

public class `LookupExecutorException` extends `RuntimeException`
Signals that an error occurred during lookup execution.

Constructors

LookupExecutorException

public **LookupExecutorException** (*String message*, *Throwable cause*)

11.37.18 LookupNameIsRepeatedException

public class **LookupNameIsRepeatedException** extends [MdsException](#)

The `LookupNameIsRepeatedException` exception signals a situation when are more than one lookups in the entity with the same name.

Constructors

LookupNameIsRepeatedException

public **LookupNameIsRepeatedException** ()

11.37.19 LookupNotFoundException

public class **LookupNotFoundException** extends [MdsException](#)

The `LookupNotFoundException` exception signals a situation in which a lookup with given id does not exist in database.

Constructors

LookupNotFoundException

public **LookupNotFoundException** ()

Constructs a new `LookupNotFoundException` with *mds.error.lookupNotFound* as a message key.

11.37.20 LookupReferencedException

public class **LookupReferencedException** extends [MdsException](#)

The `LookupReferencedException` exception signals a situation in which a lookup is used somewhere

Constructors

LookupReferencedException

public **LookupReferencedException** (*String lookups*)

11.37.21 LookupWrongParameterTypeException

public class **LookupWrongParameterTypeException** extends [RuntimeException](#)

Signals wrong type of lookup parameter

Constructors

LookupWrongParameterTypeException

public **LookupWrongParameterTypeException** (*String message*)

11.37.22 MdsException

public class **MdsException** extends *RuntimeException*

The *MdsException* exception is a basic class for all other exceptions defined in the mds module. It contains information about a message key which will be used on UI to present a message in appropriate language.

Constructors

MdsException

public **MdsException** (*String messageKey*)

Constructs a new mds exception with the specified message key.

Parameters

- **messageKey** – the message key used later to display message in appropriate language on UI.

MdsException

public **MdsException** (*String messageKey, String params*)

Constructs a new mds exception with the specified message key and params.

Parameters

- **messageKey** – the message key used later to display message in appropriate language on UI.
- **params** – the params used later to change placeholders in the message

MdsException

public **MdsException** (*String messageKey, String... params*)

Constructs a new mds exception with the specified message key and params.

Parameters

- **messageKey** – the message key used later to display message in appropriate language on UI.
- **params** – the params used later to change placeholders in the message

MdsException

public **MdsException** (*String messageKey, Throwable cause*)

Constructs a new mds exception with the specified message key and the specified cause.

Parameters

- **messageKey** – the message key used later to display message in appropriate language on UI.
- **cause** – the cause of exception.

MdsException

public **MdsException** (*String* messageKey, *String* params, *Throwable* cause)

Constructs a new mds exception with the specified message key and the specified cause.

Parameters

- **messageKey** – the message key used later to display message in appropriate language on UI.
- **params** – the params used later to change placeholders in the message
- **cause** – the cause of exception.

Methods

getMessageKey

public *String* **getMessageKey** ()

getParams

public *String* **getParams** ()

11.37.23 MdsSchedulerException

public class **MdsSchedulerException** extends *RuntimeException*

The *MdsSchedulerException* exception signals problems with scheduling MDS jobs

Constructors

MdsSchedulerException

public **MdsSchedulerException** (*String* message, *Throwable* cause)

11.37.24 NoSuchTypeException

public class **NoSuchTypeException** extends *MdsException*

An exception which signals that a given type does not exist in the database.

Constructors

NoSuchTypeException

public **NoSuchTypeException** ()

Constructs a new NoSuchTypeException with *mds.error.noSuchType* as a message key.

11.37.25 ObjectNotFoundException

public class **ObjectNotFoundException** extends [MdsException](#)

Signals that the expected object was not found in the database.

Constructors

ObjectNotFoundException

public **ObjectNotFoundException** ()

11.37.26 ObjectReadException

public class **ObjectReadException** extends [MdsException](#)

Signals that we were unable to we are unable to parse the object coming from the database.

Constructors

ObjectReadException

public **ObjectReadException** ()

ObjectReadException

public **ObjectReadException** ([Throwable](#) *cause*)

11.37.27 ObjectUpdateException

public class **ObjectUpdateException** extends [MdsException](#)

Signals that we were unable to update object instance from the provided data.

Constructors

ObjectUpdateException

public **ObjectUpdateException** ()

ObjectUpdateException

public **ObjectUpdateException** (*Throwable cause*)

11.37.28 ReservedKeywordException

public class **ReservedKeywordException** extends [MdsException](#)
Signals that field/lookup name is invalid because it is a java keyword.

Constructors

ReservedKeywordException

public **ReservedKeywordException** (*String keyword*)

11.37.29 SecurityException

public class **SecurityException** extends [MdsException](#)
The `SecurityException` exception signals a situation in which user wants to perform an operation on objects, they don't have access to.

Constructors

SecurityException

public **SecurityException** ()
Constructs a new `SecurityException` with `mds.error.securityError` as a message key.

Methods

getMessage

public [String](#) **getMessage** ()
Overrides a standard message to provide useful information about exception in logs.
Returns String message about encountered exception

11.37.30 ServiceNotFoundException

public class **ServiceNotFoundException** extends [MdsException](#)
Signals that service for a corresponding entity was not found. This most likely signals an issue with entities bundle.

Constructors

ServiceNotFoundException

```
public ServiceNotFoundException ()
```

11.37.31 SingleResultFromLookupExpectedException

public class **SingleResultFromLookupExpectedException** extends [IllegalLookupReturnTypeException](#)
Signals that the lookup returned a Collection, while a single result was expected.

Constructors

SingleResultFromLookupExpectedException

```
public SingleResultFromLookupExpectedException (String lookupName)
```

11.37.32 TrashClassNotFoundException

public class **TrashClassNotFoundException** extends [MdsException](#)
An Exception thrown when MDS fails to load trash class for an entity

Constructors

TrashClassNotFoundException

```
public TrashClassNotFoundException (String className)
```

11.37.33 TypeAlreadyExistsException

public class **TypeAlreadyExistsException** extends [MdsException](#)
The `TypeAlreadyExistsException` is thrown, if the user attempts to add a field type, with a display name that already exists in the database

Constructors

TypeAlreadyExistsException

```
public TypeAlreadyExistsException ()
```

11.37.34 TypeNotFoundException

public class **TypeNotFoundException** extends [RuntimeException](#)
The `TypeNotFoundException` exception signals a situation in which a type with given name does not exist in database.

Constructors

TypeNotFoundException

```
public NotFoundException (String msg)
```

11.37.35 TypeValidationAlreadyExistsException

```
public class TypeValidationAlreadyExistsException extends MdsException
```

The `TypeValidationAlreadyExistsException` is thrown, if the user attempts to add a type validation when there is already validation for given type

Constructors

TypeValidationAlreadyExistsException

```
public TypeValidationAlreadyExistsException ()
```

11.38 org.motechproject.mds.ex.rest

11.38.1 RestBadBodyFormatException

```
public class RestBadBodyFormatException extends RuntimeException
```

Signals that there were errors parsing the class from the provided body.

Constructors

RestBadBodyFormatException

```
public RestBadBodyFormatException (String message)
```

RestBadBodyFormatException

```
public RestBadBodyFormatException (String message, Throwable cause)
```

11.38.2 RestLookupExecutionForbiddenException

```
public class RestLookupExecutionForbiddenException extends RuntimeException
```

Signals that the lookup can not be executed through REST since it is not exposed. Thrown only for existing lookups that are not exposed.

Constructors

RestLookupExecutionForbiddenException

```
public RestLookupExecutionForbiddenException (String lookupName)
```


11.38.3 RestLookupNotFoundException

public class **RestLookupNotFoundException** extends [RuntimeException](#)
Signals that the lookup requested by REST does not exist.

Constructors

RestLookupNotFoundException

public **RestLookupNotFoundException** ([String lookupName](#))

11.38.4 RestNotSupportedException

public class **RestNotSupportedException** extends [RuntimeException](#)
Signals that the entity does not support rest.

Constructors

RestNotSupportedException

public **RestNotSupportedException** ([String entityName](#), [String moduleName](#), [String namespace](#))

Methods

getMessage

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

11.38.5 RestOperationNotSupportedException

public class **RestOperationNotSupportedException** extends [RuntimeException](#)
Signals that the given operation is not supported by the given entity.

Constructors

RestOperationNotSupportedException

public **RestOperationNotSupportedException** ([String message](#))

11.39 org.motechproject.mds.filter

11.39.1 Filter

public class **Filter** implements [Serializable](#)
Represents a filter on a field.

Constructors

Filter

public **Filter** ()

Filter

public **Filter** (*String field*, *FilterType type*)

Methods

filterForQuery

public *String* **filterForQuery** ()

getField

public *String* **getField** ()

getType

public *FilterType* **getType** ()

paramsDeclarationForQuery

public *String* **paramsDeclarationForQuery** ()

requiresFiltering

public boolean **requiresFiltering** ()

setField

public void **setField** (*String field*)

setType

public void **setType** (*FilterType type*)

valuesForQuery

public *Object*[] **valuesForQuery** ()

11.39.2 FilterType

public enum **FilterType**
Represents a method of filtering.

Enum Constants

ALL

public static final [FilterType](#) **ALL**

NO

public static final [FilterType](#) **NO**

PAST_7_DAYS

public static final [FilterType](#) **PAST_7_DAYS**

THIS_MONTH

public static final [FilterType](#) **THIS_MONTH**

THIS_YEAR

public static final [FilterType](#) **THIS_YEAR**

TODAY

public static final [FilterType](#) **TODAY**

YES

public static final [FilterType](#) **YES**

11.40 org.motechproject.mds.javassist

11.40.1 JavassistBuilder

public final class **JavassistBuilder**
Builder class for javassist related tasks. Helps with building appropriate elements of class e.g. fields, getters, field initializer

Methods

`createEnumInitializer`

public static CtField.Initializer **createEnumInitializer** (*String* enumType, *String* defaultValue)

`createField`

public static CtField **createField** (CtClass declaring, CtClass type, *String* name, *String* genericSignature)

`createGetter`

public static CtMethod **createGetter** (*String* fieldName, CtClass declaring, CtField field)

`createInitializer`

public static CtField.Initializer **createInitializer** (*String* typeClass, *String* defaultValueAsString)

`createListInitializer`

public static CtField.Initializer **createListInitializer** (*String* genericType, *Object* defaultValue)

`createLocaleInitializer`

public static CtField.Initializer **createLocaleInitializer** (*String* defaultValue)

`createSetter`

public static CtMethod **createSetter** (*String* fieldName, CtField field)

`createSimpleInitializer`

public static CtField.Initializer **createSimpleInitializer** (*String* type, *Object* defaultValue)

`createSimpleInitializer`

public static CtField.Initializer **createSimpleInitializer** (*String* type, *String* defaultValue)

`getGetterName`

public static *String* **getGetterName** (*String* fieldName, CtClass declaring)

`getSetterName`

public static *String* **getSetterName** (*String* fieldName)

11.40.2 JavassistHelper

public final class **JavassistHelper**

Helper class for javassist related tasks. Helps with generic signature generation, plus methods related with analyzing and loading javassist class representations.

Methods

containsDeclaredField

public static boolean **containsDeclaredField** (CtClass *ctClass*, String *fieldName*)

containsDeclaredMethod

public static boolean **containsDeclaredMethod** (CtClass *ctClass*, String *methodName*)

containsField

public static boolean **containsField** (CtClass *ctClass*, String *fieldName*)

containsMethod

public static boolean **containsMethod** (CtClass *ctClass*, String *methodName*)

findDeclaredField

public static CtField **findDeclaredField** (CtClass *ctClass*, String *fieldName*)

findDeclaredMethod

public static CtMethod **findDeclaredMethod** (CtClass *ctClass*, String *methodName*)

findField

public static CtField **findField** (CtClass *ctClass*, String *fieldName*)

findMethod

public static CtMethod **findMethod** (CtClass *ctClass*, String *methodName*)

genericSignature

public static String **genericSignature** (Class<?> *typeClass*, Class<?> *genericParam*)

genericSignature

public static **String genericSignature** (**Class**<?> *typeClass*, **String** *genericParam*)

genericSignature

public static **String genericSignature** (**String** *typeClass*, **String** *genericParam*)

hasInterface

public static boolean **hasInterface** (**CtClass** *ctClass*, **CtClass** *ctInterface*)

inheritsFromCustomClass

public static boolean **inheritsFromCustomClass** (**Class**<?> *clazz*)

loadClass

public static **CtClass loadClass** (**Bundle** *bundle*, **String** *className*, **ClassPool** *classPool*)

removeDeclaredFieldIfExists

public static void **removeDeclaredFieldIfExists** (**CtClass** *ctClass*, **String** *fieldName*)

removeDeclaredMethodIfExists

public static void **removeDeclaredMethodIfExists** (**CtClass** *ctClass*, **String** *methodName*)

removeFieldIfExists

public static void **removeFieldIfExists** (**CtClass** *ctClass*, **String** *fieldName*)

removeMethodIfExists

public static void **removeMethodIfExists** (**CtClass** *ctClass*, **String** *methodName*)

toClassPath

public static **String toClassPath** (**Class**<?> *clazz*)

toClassPath

public static **String toClassPath** (**String** *clazz*)

toClassPath

```
public static String toClassPath (String clazz, boolean extension)
```

toGenericParam

```
public static String toGenericParam (Class<?> clazz)
```

toGenericParam

```
public static String toGenericParam (String clazz)
```

11.40.3 JavassistLoader

```
public class JavassistLoader extends Loader<ClassData>
```

The JavassistLoader is a implementation of the `org.motechproject.mds.util.Loader` interface. It takes class information from instance of `org.motechproject.mds.domain.ClassData` and the missing classes are taken from `org.motechproject.mds.javassist.MotechClassPool`

See also: `org.motechproject.mds.util.Loader`, `org.motechproject.mds.domain.ClassData`, `org.motechproject.mds.javassist.MotechClassPool`

Constructors**JavassistLoader**

```
public JavassistLoader (MDSClassLoader classLoader)
```

Methods**doWhenClassNotFound**

```
public void doWhenClassNotFound (String name)
```

getClassDefinition

```
public Class<?> getClassDefinition (ClassData data)
```

loadClass

```
public Class<?> loadClass (ClassData arg)
```

11.40.4 MotechClassPool

public final class **MotechClassPool**

This class holds the javassist classpool, enriched by motech classes. All predefined additions to the ClassPool should take place here. The classpool should also be retrieved using this class, in order to be sure that the a initialization took place.

Methods

clearEnhancedData

public static void **clearEnhancedData** ()

getDefault

public static ClassPool **getDefault** ()

getEnhancedClassData

public static [ClassData](#) **getEnhancedClassData** ([String](#) *className*)

getEnhancedClasses

public static [Collection](#)<[ClassData](#)> **getEnhancedClasses** (boolean *includeInterfaces*)

getHistoryClassData

public static [ClassData](#) **getHistoryClassData** ([String](#) *className*)

getInterfaceName

public static [String](#) **getInterfaceName** ([String](#) *className*)

getRepositoryName

public static [String](#) **getRepositoryName** ([String](#) *className*)

getServiceImplName

public static [String](#) **getServiceImplName** ([String](#) *className*)

getTrashClassData

public static [ClassData](#) **getTrashClassData** ([String](#) *className*)

isDDEReady

public static boolean **isDDEReady** (*String className*)

isServiceInterfaceRegistered

public static boolean **isServiceInterfaceRegistered** (*String className*)

registerDDE

public static void **registerDDE** (*String className*)

registerEnhancedClassData

public static void **registerEnhancedClassData** (*ClassData enhancedClassData*)

registerEnum

public static void **registerEnum** (*String enumName*)

registerHistoryClassData

public static void **registerHistoryClassData** (*ClassData cData*)

registerServiceInterface

public static void **registerServiceInterface** (*String className*, *String interfaceName*)

registerTrashClassData

public static void **registerTrashClassData** (*ClassData cData*)

registeredEnums

public static *Collection<String>* **registeredEnums** ()

registeredInterfaces

public static *Collection<String>* **registeredInterfaces** ()

unregisterEnhancedData

public static void **unregisterEnhancedData** (*String className*)

11.41 org.motechproject.mds.jdo

11.41.1 AbstractObjectValueGenerator

public abstract class **AbstractObjectValueGenerator**<T> implements ObjectValueGenerator

Base class for other generator classes. It takes value of property (see `getPropertyName()` method) from object and modify it depending on the implementation. If the modified value is null then the `java.lang.IllegalStateException` is thrown.

Parameters

- <T> – type of property

Methods

generate

public **Object generate** (ExecutionContext *ec*, **Object obj**, ExtensionMetaData[] *extensions*)

getPropertyName

protected abstract **String** **getPropertyName** ()

modify

protected abstract **T** **modify** (T *value*)

Modifies the given value. This method cannot return null value.

Parameters

- **value** – the given value related with property.

Returns modified value.

11.41.2 CreationDateValueGenerator

public class **CreationDateValueGenerator** extends DateTimeValueGenerator

The `CreationDateValueGenerator` class is responsible for generating value for `org.motechproject.mds.util.Constants.Util.CREATION_DATE_FIELD_NAME` field.

Methods

getPropertyName

protected **String** **getPropertyName** ()

11.41.3 CreatorValueGenerator

public class **CreatorValueGenerator** extends [UsernameValueGenerator](#)

The `CreatorValueGenerator` class is responsible for generating value for `org.motechproject.mds.util.Constants.Util.CREATOR_FIELD_NAME` field.

Methods

`getPropertyName`

protected [String](#) **getPropertyName** ()

11.41.4 DateTimeValueGenerator

public abstract class **DateTimeValueGenerator** extends [AbstractObjectValueGenerator<DateTime>](#)

The `DateTimeValueGenerator` class modifies properties with `org.joda.time.DateTime` type. If the given value is null then the current time is returned; otherwise the given value is returned.

Methods

`modify`

protected [DateTime](#) **modify** ([DateTime](#) *value*)

11.41.5 MDSClassLoaderResolver

public class **MDSClassLoaderResolver** implements [ClassLoaderResolver](#)

This is a wrapper for `org.motechproject.mds.jdo.MDSClassLoaderResolverImpl`. All calls for the `org.datanucleus.ClassLoaderResolver` interface are passed to the current instance of the `ClassLoaderResolver` implementation. When we hit a `NullPointerException` originating in Felix, we can determine it is due to a synchronization bug after bundle updates - as a result of this `DataNucleus` has passed us `ClassLoaders` from the former Bundle version. In that case we reload the instance passing it the `ClassLoaders` from the new bundle.

Constructors

`MDSClassLoaderResolver`

public **MDSClassLoaderResolver** ()

`MDSClassLoaderResolver`

public **MDSClassLoaderResolver** ([ClassLoader](#) *pmLoader*)

Methods

classForName

public [Class](#) **classForName** ([String](#) *name*, [ClassLoader](#) *primary*)

classForName

public [Class](#) **classForName** ([String](#) *name*, [ClassLoader](#) *primary*, boolean *initialize*)

classForName

public [Class](#) **classForName** ([String](#) *name*)

classForName

public [Class](#) **classForName** ([String](#) *name*, boolean *initialize*)

getResource

public [URL](#) **getResource** ([String](#) *resourceName*, [ClassLoader](#) *primary*)

getResources

public [Enumeration](#)<[URL](#)> **getResources** ([String](#) *resourceName*, [ClassLoader](#) *primary*)

isAssignableFrom

public boolean **isAssignableFrom** ([String](#) *className*, [Class](#) *clazz*)

isAssignableFrom

public boolean **isAssignableFrom** ([Class](#) *clazz*, [String](#) *className*)

isAssignableFrom

public boolean **isAssignableFrom** ([String](#) *className1*, [String](#) *className2*)

registerUserClassLoader

public void **registerUserClassLoader** ([ClassLoader](#) *loader*)

setPrimary

public void **setPrimary** ([ClassLoader](#) *primary*)

setRuntimeClassLoader

```
public void setRuntimeClassLoader (ClassLoader loader)
```

unsetPrimary

```
public void unsetPrimary ()
```

11.41.6 MDSClassLoaderResolverImpl

class **MDSClassLoaderResolverImpl** extends `ClassLoaderResolverImpl`

The main purpose of the `MDSClassLoaderResolverImpl` class is to avoid situation in which standard datanucleus class loader resolver does not see classes that are saved in database. This is the main implementation that extends the standard `ClassLoaderResolverImpl` from datanucleus. Due to a synchronization bug in Felix, there are cases when we will instantiate this more then once (after we hit the bug).

Constructors

MDSClassLoaderResolverImpl

```
public MDSClassLoaderResolverImpl ()
```

MDSClassLoaderResolverImpl

```
public MDSClassLoaderResolverImpl (ClassLoader pmLoader)
```

Methods

classForName

```
public Class classForName (String name, ClassLoader primary)
```

11.41.7 MdsJdoAnnotationReader

public class **MdsJdoAnnotationReader** extends `JDOAnnotationReader`

MDS JDO annotation reader, extends the regular `org.datanucleus.api.jdo.metadata.JDOAnnotationReader`. This class was introduced because `org.datanucleus.api.jdo.metadata.JDOAnnotationReader` would not read field annotations for metadata if there was no class level JDO annotations. This extension will recognize the `org.motechproject.mds.annotations.Entity` annotation as an annotation indicating that the class is persistence capable.

Constructors

MdsJdoAnnotationReader

```
public MdsJdoAnnotationReader (MetaDataManager mgr)
```

Methods

isClassPersistenceCapable

protected AnnotationObject **isClassPersistenceCapable** (*Class cls*)

11.41.8 MdsTransactionManager

public class **MdsTransactionManager** extends *JdoTransactionManager*

We override springs transaction for classloader control. We store context classloaders as thread local variables, and switch them with the MDS classloader for the transaction. Since we only allow operations in transactions, this entry point for classloader switching is enough.

Methods

doBegin

protected void **doBegin** (*Object transaction*, *TransactionDefinition definition*)

doCleanupAfterCompletion

protected void **doCleanupAfterCompletion** (*Object transaction*)

11.41.9 ModificationDateValueGenerator

public class **ModificationDateValueGenerator** extends *DateTimeValueGenerator*

The *ModificationDateValueGenerator* class is responsible for generating value for `org.motechproject.mds.util.Constants.Util.MODIFICATION_DATE_FIELD_NAME` field.

Methods

getPropertyName

protected *String* **getPropertyName** ()

11.41.10 ModifiedByValueGenerator

public class **ModifiedByValueGenerator** extends *UsernameValueGenerator*

The *ModifiedByValueGenerator* class is responsible for generating value for `org.motechproject.mds.util.Constants.Util.MODIFIED_BY_FIELD_NAME` field.

Methods

getPropertyName

protected *String* **getPropertyName** ()

11.41.11 OwnerValueGenerator

public class **OwnerValueGenerator** extends [UsernameValueGenerator](#)

The `OwnerValueGenerator` class is responsible for generating value for `org.motechproject.mds.util.Constants.Util.OWNER_FIELD_NAME` field.

Methods

`getPropertyName`

protected [String](#) **getPropertyName** ()

11.41.12 SchemaGenerator

public class **SchemaGenerator** implements [InitializingBean](#)

The schema generator class is responsible for generating the table schema for entities upon start. Schema for all entity classes has to be generated, otherwise issues might arise in foreign key generation for example. This code runs in the generated entities bundle.

Constructors

`SchemaGenerator`

public **SchemaGenerator** (JDOPersistenceManagerFactory *persistenceManagerFactory*)

Methods

`afterPropertiesSet`

public void **afterPropertiesSet** ()

`generateSchema`

public void **generateSchema** ()

11.41.13 TimeTypeConverter

public class **TimeTypeConverter** implements [TypeConverter](#)<[Time](#), [String](#)>

This is datanucleus type converter we plug in. It is responsible for converting `org.motechproject.commons.date.model.Time` instances to Strings which are persisted.

Methods

`toDatastoreType`

public [String](#) **toDatastoreType** ([Time](#) *memberValue*)

toMemberType

```
public Time toMemberType (String datastoreValue)
```

11.41.14 UsernameValueGenerator

```
public abstract class UsernameValueGenerator extends AbstractObjectValueGenerator<String>
```

The UsernameValueGenerator class modifies properties with `java.lang.String` type. The given value is returned without any change if it is not blank. Otherwise the class tries to get current logged user name. If the user exists and name is not blank then this name is returned otherwise the empty string is returned.

Methods

modify

```
protected String modify (String value)
```

11.42 org.motechproject.mds.lookup

11.42.1 LookupExecutor

```
public class LookupExecutor
```

This class allowing executing lookups by providing the lookup name as a string and the lookup params in name-value map. Used both by the REST api and the Databrowser UI for executing lookups based on only metadata. The dataservice and metadata must be provided during construction.

Constructors

LookupExecutor

```
public LookupExecutor (MotechDataService dataService, LookupDto lookup, Map<Long, FieldDto> fieldsById)
```

Methods

execute

```
public Object execute (Map<String, ?> lookupMap)
```

execute

```
public Object execute (Map<String, ?> lookupMap, QueryParams queryParams)
```

executeCount

```
public long executeCount (Map<String, ?> lookupMap)
```


11.43 org.motechproject.mds.performance.domain

11.43.1 Sample

public class **Sample**

Constructors

Sample

public **Sample** ()

Sample

public **Sample** (*Integer testInt*, *String testString*)

Methods

getTestInt

public *Integer* **getTestInt** ()

getTestString

public *String* **getTestString** ()

setTestInt

public void **setTestInt** (*Integer testInt*)

setTestString

public void **setTestString** (*String testString*)

11.44 org.motechproject.mds.performance.service

11.44.1 MdsDummyDataGenerator

public interface **MdsDummyDataGenerator**

Methods

clearEntities

void **clearEntities** ()

generateDummyEntities

void **generateDummyEntities** (int *numberOfEntities*, boolean *regenerateBundle*)

Generate a given amount of empty entities

Parameters

- **numberOfEntities** – How many entities should be generated
- **regenerateBundle** – Should the data bundle be regenerated after entities are created

Throws

- **IOException** –

generateDummyEntities

void **generateDummyEntities** (int *numberOfEntities*, int *fieldsPerEntity*, int *lookupsPerEntity*, boolean *regenerateBundle*)

Generate a given amount of entities

Parameters

- **numberOfEntities** – How many entities should be generated
- **fieldsPerEntity** – How many fields should each entity have - type of field is assigned randomly
- **lookupsPerEntity** – How many lookups should each entity have
- **regenerateBundle** – Should the data bundle be regenerated after entities are created

Throws

- **IOException** –

generateDummyInstances

void **generateDummyInstances** (Long *entityId*, int *numberOfInstances*)

Generates a given amount of instances. For most common field types, values will be automatically assigned.

Parameters

- **entityId** – Id of an entity, for which we want to generate instances
- **numberOfInstances** – How many instances should be generated

Throws

- **IllegalAccessException** –
- **ClassNotFoundException** –
- **InstantiationException** –

generateDummyInstances

void **generateDummyInstances** (Long *entityId*, int *numberOfInstances*, int *numberOfHistoricalRevisions*, int *numberOfTrashInstances*)

Generates a given amount of instances, historical revisions and deleted instances. For most common field types, values will be automatically assigned.

Parameters

- **entityId** – Id of an entity, for which we want to generate instances
- **numberOfInstances** – How many instances should be generated
- **numberOfHistoricalRevisions** – How many historical revisions should be generated
- **numberOfTrashInstances** – How many instances in trash should be generated

Throws

- **IllegalAccessException** –
- **ClassNotFoundException** –
- **InstantiationException** –

getEntityPrefix

String **getEntityPrefix**()

getFieldPrefix

String **getFieldPrefix**()

getLookupPrefix

String **getLookupPrefix**()

getService

MotechDataService **getService**(BundleContext *bundleContext*, String *className*)

makeDummyInstance

Object **makeDummyInstance**(Long *entityId*)

Makes a dummy instance for a given entity. It won't insert the created instance into the database, so it can be used to prepare a large dataset of random instances (for example, to insert them manually and measure time)

Parameters

- **entityId** – Id of an entity, for which we want to generate instance

Throws

- **IllegalAccessException** –
- **ClassNotFoundException** –
- **InstantiationException** –

Returns Generated instance, with randomly assigned fields

setEntityPrefix

void **setEntityPrefix** ([String](#) *entityPrefix*)
Sets a prefix for the generated entities. Defaults to “Entity”

Parameters

- **entityPrefix** – The prefix for generated entities

setFieldPrefix

void **setFieldPrefix** ([String](#) *fieldPrefix*)
Sets the prefix for the generated fields. Defaults to “field”

Parameters

- **fieldPrefix** – The prefix for generated fields

setLookupPrefix

void **setLookupPrefix** ([String](#) *lookupPrefix*)
Sets a prefix for the generated lookups. Defaults to “Lookup”

Parameters

- **lookupPrefix** – The prefix for generated lookups

setUpSecurityContext

void **setUpSecurityContext** ()
Prepares a dummy security context.

11.44.2 SampleService

public interface **SampleService** extends [MotechDataService<Sample>](#)

11.45 org.motechproject.mds.performance.service.impl

11.45.1 MdsDummyDataGeneratorImpl

public class **MdsDummyDataGeneratorImpl** implements [MdsDummyDataGenerator](#)

The `MdsDummyDataGenerator` class, allows developers and testers to create any amount of dummy entities and instances in MDS.

Constructors

MdsDummyDataGeneratorImpl

public **MdsDummyDataGeneratorImpl** ([EntityService](#) *entityService*, [JarGeneratorService](#) *jarGeneratorService*, [BundleContext](#) *bundleContext*)

Methods

clearEntities

```
public void clearEntities ()
```

generateDummyEntities

```
public void generateDummyEntities (int number, boolean regenerateBundle)
```

generateDummyEntities

```
public void generateDummyEntities (int numberOfEntities, int fieldsPerEntity, int lookupsPerEntity,  
                                     boolean regenerateBundle)
```

generateDummyInstances

```
public void generateDummyInstances (Long entityId, int numberOfInstances)
```

generateDummyInstances

```
public void generateDummyInstances (Long entityId, int numberOfInstances, int numberOfHistorical-  
                                     Revisions, int numberOfTrashInstances)
```

getEntityPrefix

```
public String getEntityPrefix ()
```

getFieldPrefix

```
public String getFieldPrefix ()
```

getLookupPrefix

```
public String getLookupPrefix ()
```

getService

```
public MotechDataService getService (BundleContext bundleContext, String className)
```

makeDummyInstance

```
public Object makeDummyInstance (Long entityId)
```

setEntityPrefix

```
public void setEntityPrefix (String entityPrefix)
```

setFieldPrefix

```
public void setFieldPrefix (String fieldPrefix)
```

setLookupPrefix

```
public void setLookupPrefix (String lookupPrefix)
```

setUpSecurityContext

```
public void setUpSecurityContext ()
```

11.46 org.motechproject.mds.query

11.46.1 CollectionProperty

```
public class CollectionProperty extends Property<Collection>
```

The `CollectionProperty` class represent a property that will be used in JDO query and it has to have the given value(s).

Constructors**CollectionProperty**

```
public CollectionProperty (String name, Object value)
```

Methods**generateDeclareParameter**

```
public CharSequence generateDeclareParameter (int idx)
```

generateFilter

```
public CharSequence generateFilter (int idx)
```

shouldIgnoreThisProperty

```
protected boolean shouldIgnoreThisProperty ()
```

unwrap

```
public Collection unwrap ()
```

11.46.2 CustomOperatorProperty

```
public class CustomOperatorProperty<T> extends Property<T>
```

The `CustomOperatorProperty` class represents a property that will be used in JDO query. This class allows inserting a custom operator, such as `>`, `<=`, `matches()`, etc.

Parameters

- `<T>` – type of the passed value

Constructors

CustomOperatorProperty

```
public CustomOperatorProperty (String name, T value, String operator)
```

Methods

generateFilter

```
public CharSequence generateFilter (int idx)
```

getOperator

```
public String getOperator ()
```

isOperatorAMethod

```
public boolean isOperatorAMethod ()
```

11.46.3 EqualProperty

```
public class EqualProperty<T> extends Property<T>
```

The `EqualProperty` class represents a property that will be used in JDO query and it has to be equal to the given value.

Parameters

- `<T>` – type of the passed value

Constructors

EqualProperty

```
public EqualProperty (String name, T value)
```

Methods

generateFilter

```
public CharSequence generateFilter (int idx)
```

11.46.4 MatchesProperty

```
public class MatchesProperty extends CustomOperatorProperty<String>
```

A convenience extension of the `org.motechproject.mds.query.CustomOperatorProperty`.

The custom operator is “matches()” and the underlying type is String. The value passed will be wrapped inside `.*.*` for matching purposes.

Constructors

MatchesProperty

```
public MatchesProperty (String name, String value)
```

11.46.5 Property

```
public abstract class Property<T>
```

The `Property` class represents a property that will be used in JDO query. Classes that extend this class should define how that property should be used in WHERE section in JDO query.

Parameters

- `<T>` – type of the passed value

Constructors

Property

```
protected Property (String name, T value)
```

Methods

asDeclareParameter

```
public CharSequence asDeclareParameter (int idx)
```

asFilter

```
public CharSequence asFilter (int idx)
```

containsOnlyNullValues

```
protected boolean containsOnlyNullValues (Collection collection)
```


generateDeclareParameter

protected [CharSequence](#) **generateDeclareParameter** (int *idx*)

generateFilter

protected abstract [CharSequence](#) **generateFilter** (int *idx*)

getName

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

getValue

public T **getValue** ()

shouldIgnoreThisProperty

protected boolean **shouldIgnoreThisProperty** ()

unwrap

public [Collection](#) **unwrap** ()

11.46.6 PropertyBuilder

public final class **PropertyBuilder**

The `PropertyBuilder` class is a util class that helps create appropriate property class based on passed name and value.

Methods**create**

public static [Property](#) **create** ([Field](#) *field*, [Object](#) *value*)

create

public static [Property](#) **create** ([String](#) *name*, [Object](#) *value*)

create

public static [Property](#) **create** ([String](#) *name*, [Object](#) *value*, [String](#) *operator*)

11.46.7 QueryExecution

public interface **QueryExecution**<T>

Allows users to execute custom queries through Motech Data Services. Implementations need only to implement the execute method, which can operate directly on the `javax.jdo.Query` object. The return value type is left to the implementation.

Parameters

- <T> – the type that will be returned from this query

Methods

execute

T **execute** (*Query query*, *InstanceSecurityRestriction restriction*)

11.46.8 QueryExecutor

public final class **QueryExecutor**

The `QueryExecutor` util class provides methods that help execute a JDO query.

Methods

execute

public static Object **execute** (*Query query*, *InstanceSecurityRestriction restriction*)

execute

public static Object **execute** (*Query query*, *Object value*, *InstanceSecurityRestriction restriction*)

executeWithArray

public static Object **executeWithArray** (*Query query*, *Object[] values*, *InstanceSecurityRestriction restriction*)

executeWithArray

public static Object **executeWithArray** (*Query query*, *List<Property> properties*)

executeWithFilter

public static Object **executeWithFilter** (*Query query*, *Filter filter*, *InstanceSecurityRestriction restriction*)

11.46.9 QueryParams

public class **QueryParams** implements [Serializable](#)

Utility class containing parameters which control order and size of query results. Used mainly for paging/ordering queries from the UI.

Constructors

QueryParams

public **QueryParams** ([Integer](#) *page*, [Integer](#) *pageSize*)

QueryParams

public **QueryParams** ([Order](#) *order*)

QueryParams

public **QueryParams** ([Integer](#) *page*, [Integer](#) *pageSize*, [Order](#) *order*)

Methods

ascOrder

public static [QueryParams](#) **ascOrder** ([String](#) *field*)

descOrder

public static [QueryParams](#) **descOrder** ([String](#) *field*)

getOrder

public [Order](#) **getOrder** ()

getPage

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

getPageSize

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

isOrderSet

public boolean **isOrderSet** ()

isPagingSet

public boolean **isPagingSet** ()

11.46.10 QueryUtil

public final class **QueryUtil**

The `QueryUtil` util class provides methods that help developer to create a JDO query.

See also: `javax.jdo.Query`

Methods

asMatchesPattern

public static `String` **asMatchesPattern** (`String` *string*)

Returns the string in a form a pattern used for searching with the matches operator.

Parameters

- **string** – The string to search for.

Returns The string in format `.*<string>.*` or the string unchanged if it is empty.

setCountResult

public static void **setCountResult** (`Query` *query*)

setQueryParams

public static void **setQueryParams** (`Query` *query*, `QueryParams` *queryParams*)

useFilter

public static void **useFilter** (`Query` *query*, `Filter` *filter*)

useFilter

public static void **useFilter** (`Query` *query*, `String`[] *properties*, `Object`[] *values*)

useFilter

public static void **useFilter** (`Query` *query*, `String`[] *properties*, `Object`[] *values*, `InstanceSecurityRestriction` *restriction*)

useFilter

public static void **useFilter** (`Query` *query*, `List`<`Property`> *properties*)

useFilter

```
public static void useFilter (Query query, List<Property> properties, InstanceSecurityRestriction restriction)
```

useFilterFromPattern

```
public static void useFilterFromPattern (Query query, String pattern, List<Property> properties)
```

11.46.11 RangeProperty

```
public class RangeProperty<T> extends Property<Range<T>>
```

The `RangeProperty` class represents a property that will be used in JDO query and it has to be inside the given range.

Parameters

- `<T>` – type used in range.

Constructors

RangeProperty

```
public RangeProperty (String name, Range<T> value)
```

Methods

generateDeclareParameter

```
public CharSequence generateDeclareParameter (int idx)
```

generateFilter

```
public CharSequence generateFilter (int idx)
```

shouldIgnoreThisProperty

```
protected boolean shouldIgnoreThisProperty ()
```

unwrap

```
public Collection unwrap ()
```

11.46.12 RestrictionProperty

```
public class RestrictionProperty extends EqualProperty<String>
```

The `RestrictionProperty` class represents a property that will be used in JDO query and depends on restriction criteria the `creator` or `owner` field in an instance has to have the appropriate user name.

Constructors

RestrictionProperty

public **RestrictionProperty** ([InstanceSecurityRestriction](#) *restriction*, [String](#) *value*)

11.46.13 SetProperty

public class **SetProperty**<T> extends [Property](#)<[Set](#)<T>>

The `SetProperty` class represents a property that will be used in JDO query and it has to have one of the value from the given set.

Parameters

- <T> – type used in set.

Constructors

SetProperty

public **SetProperty** ([String](#) *name*, [Set](#)<T> *value*)

Methods

generateDeclareParameter

public [CharSequence](#) **generateDeclareParameter** (int *idx*)

generateFilter

public [CharSequence](#) **generateFilter** (int *idx*)

shouldIgnoreThisProperty

protected boolean **shouldIgnoreThisProperty** ()

unwrap

public [Collection](#) **unwrap** ()

11.46.14 SqlQueryExecution

public interface **SqlQueryExecution**<T>

Allows users to execute custom SQL queries through Motech Data Services. Implementations need to implement the `execute` method, which can operate directly on the `javax.jdo.Query` object and `getSqlQuery()` should return the sql query that will be executed. The return value type is left to the implementation. It is not advised to rely on raw SQL, however some use cases may require it.

Parameters

- `<T>` – the type that will be returned from this query

Methods

`execute`

`T execute (Query query)`

`getSqlQuery`

`String getSqlQuery ()`

11.47 org.motechproject.mds.repository

11.47.1 AllConfigSettings

public class **AllConfigSettings** extends `MotechDataRepository<ConfigSettings>`
`AllConfigSettings` is responsible for communication with database for MDS configuration.

Constructors

`AllConfigSettings`

public **AllConfigSettings** ()

Methods

`addOrUpdate`

public void **addOrUpdate** (`ConfigSettings record`)

11.47.2 AllEntities

public class **AllEntities** extends `MotechDataRepository<Entity>`
The `AllEntities` class is a repository class that operates on instances of `org.motechproject.mds.domain.Entity`.

Constructors

`AllEntities`

public **AllEntities** ()

Methods

contains

public boolean **contains** (*String className*)

create

public *Entity* **create** (*EntityDto dto*)

delete

public void **delete** (*Long id*)

retrieveByClassName

public *Entity* **retrieveByClassName** (*String className*)

retrieveById

public *Entity* **retrieveById** (*Long id*)

11.47.3 AllEntityAudits

public class **AllEntityAudits** extends *MotechDataRepository<EntityAudit>*

This a repository for persisting entity audits. It provides methods which created audits, by cloning the giving entity.

Constructors

AllEntityAudits

public **AllEntityAudits** ()

Methods

createAudit

public *EntityAudit* **createAudit** (*Entity entity*, *String username*)

11.47.4 AllEntityDrafts

public class **AllEntityDrafts** extends *MotechDataRepository<EntityDraft>*

This a repository for persisting entity drafts. It provides methods which created drafts, by cloning the giving entity.

Constructors

AllEntityDrafts

```
public AllEntityDrafts ()
```

Methods

create

```
public EntityDraft create (Entity entity, String username)
```

deleteAll

```
public void deleteAll (Entity entity)
```

retrieve

```
public EntityDraft retrieve (Entity entity, String username)
```

retrieveAll

```
public List<EntityDraft> retrieveAll (String username)
```

retrieveAll

```
public List<EntityDraft> retrieveAll (Entity entity)
```

setProperties

```
public void setProperties (EntityDraft draft, Entity entity)
```

setProperties

```
public void setProperties (EntityDraft draft, Entity entity, String username)
```

update

```
public EntityDraft update (EntityDraft draft)
```

11.47.5 AllTypeSettings

```
public class AllTypeSettings extends MotechDataRepository<TypeSetting>
```

The `AllTypeSettings` class is a repository class that operates on instances of `org.motechproject.mds.domain.TypeSetting`.

Constructors

AllTypeSettings

```
public AllTypeSettings ()
```

11.47.6 AllTypeValidations

```
public class AllTypeValidations extends MotechDataRepository<TypeValidation>
```

The `AllTypeValidations` class is a repository class that operates on instances of `org.motechproject.mds.domain.TypeValidation`.

Constructors

AllTypeValidations

```
public AllTypeValidations ()
```

11.47.7 AllTypes

```
public class AllTypes extends MotechDataRepository<Type>
```

The `AllTypes` repository class allows persistence and retrieving of Field Types in Data Services database.

Constructors

AllTypes

```
public AllTypes ()
```

Methods

retrieveByClassName

```
public Type retrieveByClassName (String className)
```

11.47.8 MetadataHolder

```
public class MetadataHolder
```

Holds the current JDO metadata for Seuss. Allows reloading the metadata and retrieval for modifications.

Methods

getJdoMetadata

```
public JDOMetadata getJdoMetadata ()
```

reloadMetadata

```
public JDOMetadata reloadMetadata ()
```

setPersistenceManagerFactory

```
public void setPersistenceManagerFactory (PersistenceManagerFactory persistenceManagerFactory)
```

11.47.9 MotechDataRepository

```
public abstract class MotechDataRepository<T>
```

This is a basic repository class with standard CRUD operations. It should be used by other repositories inside this package.

This class is also used as super class to create a repository related with the given entity schema in `org.motechproject.mds.builder.EntityInfrastructureBuilder`.

Parameters

- `<T>` – the type of class

Constructors**MotechDataRepository**

```
protected MotechDataRepository (Class<T> classType)
```

Methods**count**

```
public long count (InstanceSecurityRestriction restriction)
```

count

```
public long count (String[] properties, Object[] values, InstanceSecurityRestriction restriction)
```

count

```
public long count (List<Property> properties, InstanceSecurityRestriction restriction)
```

countForFilter

```
public long countForFilter (Filter filter)
```

countForFilter

public long **countForFilter** ([Filter](#) *filter*, [InstanceSecurityRestriction](#) *restriction*)

create

public T **create** (T *object*)

delete

public void **delete** (T *object*)

delete

public void **delete** ([String](#) *property*, [Object](#) *value*, [InstanceSecurityRestriction](#) *restriction*)

delete

public void **delete** ([String](#)[] *properties*, [Object](#)[] *values*, [InstanceSecurityRestriction](#) *restriction*)

deleteAll

public void **deleteAll** ([String](#) *property*, [Object](#) *value*)

deleteAll

public void **deleteAll** ([String](#) *property*, [Object](#) *value*, [InstanceSecurityRestriction](#) *restriction*)

deleteAll

public void **deleteAll** ([String](#)[] *properties*, [Object](#)[] *values*, [InstanceSecurityRestriction](#) *restriction*)

exists

public boolean **exists** ([String](#) *property*, [Object](#) *value*)

exists

public boolean **exists** ([String](#)[] *properties*, [Object](#)[] *values*)

filter

public [List](#)<T> **filter** ([Filter](#) *filter*, [QueryParams](#) *queryParams*)

filter

```
public List<T> filter (Filter filter, QueryParams queryParams, InstanceSecurityRestriction restriction)
```

getClassType

```
public Class<T> getClassType ()
```

getDetachedField

```
public Object getDetachedField (T instance, String field)
```

getPersistenceManager

```
public PersistenceManager getPersistenceManager ()
```

retrieve

```
public T retrieve (Object key)
```

retrieve

```
public T retrieve (String property, Object value)
```

retrieve

```
public T retrieve (String property, Object value, InstanceSecurityRestriction restriction)
```

retrieve

```
public T retrieve (String[] properties, Object[] values)
```

retrieve

```
public T retrieve (String[] properties, Object[] values, InstanceSecurityRestriction restriction)
```

retrieveAll

```
public List<T> retrieveAll ()
```

retrieveAll

```
public List<T> retrieveAll (InstanceSecurityRestriction restriction)
```

retrieveAll

```
public List<T> retrieveAll (String property, Object value)
```

retrieveAll

```
public List<T> retrieveAll (String property, Object value, InstanceSecurityRestriction restriction)
```

retrieveAll

```
public List<T> retrieveAll (String[] properties, Object[] values, InstanceSecurityRestriction restriction)
```

retrieveAll

```
public List<T> retrieveAll (String[] properties, Object[] values, QueryParams queryParams, InstanceSecurityRestriction restriction)
```

retrieveAll

```
public List<T> retrieveAll (QueryParams queryParams, InstanceSecurityRestriction restriction)
```

retrieveAll

```
public List<T> retrieveAll (List<Property> properties, InstanceSecurityRestriction restriction)
```

retrieveAll

```
public List<T> retrieveAll (List<Property> properties, QueryParams queryParams, InstanceSecurityRestriction restriction)
```

setPersistenceManagerFactory

```
public void setPersistenceManagerFactory (PersistenceManagerFactory persistenceManagerFactory)
```

update

```
public T update (T object)
```

11.48 org.motechproject.mds.rest

11.48.1 MdsRestFacade

```
public interface MdsRestFacade<T>
```

Interface called by the REST controller REST operations. Should be exposed as

an OSGi service for each MDS Entity. If rest is not supported, it throws a `org.motechproject.mds.ex.rest.RestNotSupportedException`.

Parameters

- `<T>` – the entity class.

Methods

create

`T create (InputStream instanceBody)`

delete

`void delete (Long id)`

executeLookup

`Object executeLookup (String lookupName, Map<String, String> lookupMap, QueryParams queryParams)`

get

`List<T> get (QueryParams queryParams)`

get

`T get (Long id)`

update

`T update (InputStream instanceBody)`

11.48.2 MdsRestFacadeImpl

public class **MdsRestFacadeImpl**<T> implements `MdsRestFacade`<T>

This `org.motechproject.mds.rest.MdsRestFacade` implementation retrieves REST related metadata on initialization. It uses an instance of `org.motechproject.mds.service.MotechDataService` for operations and the jackson JSON library for parsing InputStreams.

Parameters

- `<T>` –

Methods

create

public T **create** (*InputStream instanceBody*)

delete

public void **delete** (*Long id*)

executeLookup

public Object **executeLookup** (*String lookupName*, *Map<String, String> lookupMap*, *QueryParams queryParams*)

get

public List<T> **get** (*QueryParams queryParams*)

get

public T **get** (*Long id*)

init

public void **init** ()

setAllEntities

public void **setAllEntities** (*AllEntities allEntities*)

setDataService

public void **setDataService** (*MotechDataService<T> dataService*)

update

public T **update** (*InputStream instanceBody*)

11.49 org.motechproject.mds.service

11.49.1 DefaultMotechDataService

public abstract class **DefaultMotechDataService**<T> implements [MotechDataService](#)<T>

This is a basic implementation of [org.motechproject.mds.service.MotechDataService](#).
Mainly it is used as super class to create a service related with the given entity schema in [org.motechproject.mds.builder.EntityInfrastructureBuilder](#) but it can be also used by other services inside this package.

Parameters

- <T> – the type of entity schema.

Methods

count

public long **count** ()

count

protected long **count** ([List](#)<[Property](#)> *properties*)

countForFilter

public long **countForFilter** ([Filter](#) *filter*)

create

public T **create** (T *object*)

delete

public void **delete** (T *object*)

delete

public void **delete** ([String](#) *primaryKeyName*, [Object](#) *value*)

deleteAll

public void **deleteAll** ()

doInTransaction

public <R> R **doInTransaction** ([TransactionCallback](#)<R> *transactionCallback*)

executeQuery

```
public <R> R executeQuery (QueryExecution<R> queryExecution)
```

executeSQLQuery

```
public <R> R executeSQLQuery (SqlQueryExecution<R> queryExecution)
```

filter

```
public List<T> filter (Filter filter)
```

filter

```
public List<T> filter (Filter filter, QueryParams queryParams)
```

findById

```
public T findById (Long id)
```

findTrashInstanceById

```
public T findTrashInstanceById (Object instanceId, Object entityId)
```

getClassType

```
public Class<T> getClassType ()
```

getDetachedField

```
public Object getDetachedField (T instance, String fieldName)
```

getId

```
protected Object getId (T instance)
```

getRepository

```
protected MotechDataRepository<T> getRepository ()
```

initializeSecurityState

```
public void initializeSecurityState ()
```

retrieve

public T **retrieve** (*String* *primaryKeyName*, *Object* *value*)

retrieveAll

public *List*<T> **retrieveAll** ()

retrieveAll

public *List*<T> **retrieveAll** (*QueryParams* *queryParams*)

retrieveAll

protected *List*<T> **retrieveAll** (*List*<*Property*> *properties*)

retrieveAll

protected *List*<T> **retrieveAll** (*List*<*Property*> *properties*, *QueryParams* *queryParams*)

revertFromTrash

public void **revertFromTrash** (*Object* *newInstance*, *Object* *trash*)

setAllEntities

public void **setAllEntities** (*AllEntities* *allEntities*)

setEntityService

public void **setEntityService** (*EntityService* *entityService*)

setHistoryService

public void **setHistoryService** (*HistoryService* *historyService*)

setRepository

public void **setRepository** (*MotechDataRepository*<T> *repository*)

setTransactionManager

public void **setTransactionManager** (*JdoTransactionManager* *transactionManager*)

setTrashService

```
public void setTrashService (TrashService trashService)
```

update

```
public T update (T object)
```

updateFromTransient

```
public T updateFromTransient (T transientObject)
```

validateCredentials

```
protected InstanceSecurityRestriction validateCredentials ()
```

validateCredentials

```
protected InstanceSecurityRestriction validateCredentials (T instance)
```

11.49.2 EntityService

```
public interface EntityService
```

This interface provides methods related with executing actions on an entity.

Methods

abandonChanges

```
void abandonChanges (Long entityId)
```

addDisplayedFields

```
void addDisplayedFields (EntityDto entityDto, Map<String, Long> positions)
```

addFields

```
void addFields (EntityDto entity, Collection<FieldDto> fields)
```

addFilterableFields

```
void addFilterableFields (EntityDto entityDto, Collection<String> fieldNames)
```

addLookups

void **addLookups** (Long *entityId*, Collection<LookupDto> *lookups*)

commitChanges

void **commitChanges** (Long *entityId*)

commitChanges

void **commitChanges** (Long *entityId*, String *changesOwner*)

createEntity

EntityDto **createEntity** (EntityDto *entityDto*)

deleteEntity

void **deleteEntity** (Long *entityId*)

findEntityFieldByName

FieldDto **findEntityFieldByName** (Long *entityId*, String *name*)

findFieldByName

FieldDto **findFieldByName** (Long *entityId*, String *name*)

getAdvancedSettings

AdvancedSettingsDto **getAdvancedSettings** (Long *entityId*)

getAdvancedSettings

AdvancedSettingsDto **getAdvancedSettings** (Long *entityId*, boolean *committed*)

getCurrentSchemaVersion

Long **getCurrentSchemaVersion** (String *entityClassName*)

getDisplayFields

List<FieldDto> **getDisplayFields** (Long *entityId*)

getEntitiesWithLookups

List<EntityDto> **getEntitiesWithLookups** ()

getEntity

EntityDto **getEntity** (Long *entityId*)

getEntityByClassName

EntityDto **getEntityByClassName** (String *className*)

getEntityDraft

EntityDraft **getEntityDraft** (Long *entityId*)

getEntityDraft

EntityDraft **getEntityDraft** (Long *entityId*, String *username*)

getEntityFields

List<FieldDto> **getEntityFields** (Long *entityId*)

getEntityForEdit

EntityDto **getEntityForEdit** (Long *entityId*)

getEntityLookups

List<LookupDto> **getEntityLookups** (Long *entityId*)

getFields

List<FieldDto> **getFields** (Long *entityId*)

getLookupByName

LookupDto **getLookupByName** (Long *entityId*, String *lookupName*)

listEntities

List<EntityDto> **listEntities** ()

listEntities

List<EntityDto> **listEntities** (boolean *withSecurityCheck*)

listWorkInProgress

List<EntityDto> **listWorkInProgress** ()

saveDraftEntityChanges

DraftResult **saveDraftEntityChanges** (Long *entityId*, DraftData *draftData*, String *username*)

saveDraftEntityChanges

DraftResult **saveDraftEntityChanges** (Long *entityId*, DraftData *draftData*)

updateComboboxValues

void **updateComboboxValues** (Long *entityId*, Map<String, Collection> *fieldValuesToUpdate*)

updateDraft

EntityDto **updateDraft** (Long *entityId*)

updateRestOptions

void **updateRestOptions** (Long *entityId*, RestOptionsDto *restOptionsDto*)

11.49.3 HistoryService

public interface **HistoryService**

The **HistoryService** provides methods related with processing historical changes on the given instance of entity.

Methods**countHistoryRecords**

long **countHistoryRecords** (Object *instance*)

Returns a number of historical revisions for the given instance.

Parameters

- **instance** – an instance to count historical revisions for

Returns a number of historical revisions of the instance

getHistoryForInstance

List **getHistoryForInstance** (*Object instance*, *QueryParams queryParams*)

Returns the historical data for the given instance. This method return historical data only for objects that are not in the MDS trash. For trash instances the return value will be incorrect.

Parameters

- **instance** – an instance created from the given entity definition.
- **queryParams** – Query parameters such as page number, size of page and sort direction. If null method will return all history records.

Returns a list of historical data related with the given instance.

getSingleHistoryInstance

Object **getSingleHistoryInstance** (*Object instance*, *Long historyId*)

Returns a single historical revision for the given instance.

Parameters

- **instance** – an instance to retrieve historical revisions from
- **historyId** – id of a historical revision

Returns Historical revision or null if not found

record

void **record** (*Object instance*)

Records changes made on the given instance of entity. The first historical data should be equal to data inside the given instance. Two instance of historical data should be connected using appropriate fields (defined in history class definition). This method should be used only for instances that are not in the MDS trash.

Parameters

- **instance** – an instance created from the given entity definition.

remove

void **remove** (*Object instance*)

Removes all historical data with information what changes were made on the given instance of entity.

Parameters

- **instance** – an instance created from the given entity definition.

setTrashFlag

void **setTrashFlag** (*Object instance*, *Object trash*, *boolean flag*)

Sets the trash flag for historical data related with the given instance object.

Parameters

- **instance** – an instance created from the given entity definition.

- **flag** – true if instance was moved to trash; otherwise false.

11.49.4 JarGeneratorService

public interface **JarGeneratorService**

This interface provides methods to create a bundle jar with all entities defined in MDS module.

Fields

BLUEPRINT_TEMPLATE

String **BLUEPRINT_TEMPLATE**

BLUEPRINT_XML

String **BLUEPRINT_XML**

BUNDLE_IMPORTS

String **BUNDLE_IMPORTS**

DATANUCLEUS_PROPERTIES

String **DATANUCLEUS_PROPERTIES**

ENTITY_LIST_FILE

String **ENTITY_LIST_FILE**

MDS_COMMON_CONTEXT

String **MDS_COMMON_CONTEXT**

MDS_ENTITIES_CONTEXT

String **MDS_ENTITIES_CONTEXT**

MDS_ENTITIES_CONTEXT_TEMPLATE

String **MDS_ENTITIES_CONTEXT_TEMPLATE**

MOTECH_MDS_PROPERTIES

String **MOTECH_MDS_PROPERTIES**

PACKAGE_JDO

String **PACKAGE_JDO**

Methods

generate

File **generate** ()

Generates a jar file that contains entity class definitions, repositories, interfaces, implementations of these interfaces. The jar should also contains class related with historical data and trash.

Throws

- **IOException** – if an I/O error occurs while the jar is creating.

Returns file that point to an entitites bundle jar.

regenerateMdsDataBundle

void **regenerateMdsDataBundle** (boolean *buildDDE*)

Constructs entities, builds and starts the entities bundle jar

Parameters

- **buildDDE** – `true` if class definitions for entities from outside bundles should also be created; otherwise `false`.

See also: `.generate()`

regenerateMdsDataBundle

void **regenerateMdsDataBundle** (boolean *buildDDE*, boolean *startBundle*)

Constructs entities, builds the entities bundle jar. The generated bundle will start only if the **startBundle** will be set to `true`.

Parameters

- **buildDDE** – `true` if class definitions for entities from outside bundles should also be created; otherwise `false`.
- **startBundle** – `true` if the generated bundle should start; otherwise `false`.

See also: `.generate()`

regenerateMdsDataBundleAfterDdeEnhancement

void **regenerateMdsDataBundleAfterDdeEnhancement** (String *moduleName*)

Constructs entities, builds and starts the entities bundle jar. This method should be used after DDE enhancement. It will build all DDE classes and refresh the module from which the DDE being enhanced comes from.

Parameters

- **moduleName** – module name of the entity from which the enhanced DDE comes from

See also: `.generate()`

11.49.5 MDSLlookupService

public interface **MDSLlookupService**

This service allows executing lookups on entities given their classes or class names and lookup names as Strings. Allows generic access to any entity in MDS. This is just a facade and all data access goes through the underlying data service.

Methods

count

long **count** (Class entityClass, String lookupName, Map<String, ?> lookupParams)

count

long **count** (String entityClassName, String lookupName, Map<String, ?> lookupParams)

countAll

long **countAll** (Class entityClass)

countAll

long **countAll** (String entityClassName)

findMany

<T> List<T> **findMany** (Class<T> entityClass, String lookupName, Map<String, ?> lookupParams)

findMany

<T> List<T> **findMany** (String entityClassName, String lookupName, Map<String, ?> lookupParams)

findMany

<T> List<T> **findMany** (Class<T> entityClass, String lookupName, Map<String, ?> lookupParams, QueryParams queryParams)

findMany

<T> List<T> **findMany** (String entityClassName, String lookupName, Map<String, ?> lookupParams, QueryParams queryParams)

findOne

<T> T **findOne** (Class<T> entityClass, String lookupName, Map<String, ?> lookupParams)

findOne

```
<T> T findOne (String entityClassName, String lookupName, Map<String, ?> lookupParams)
```

retrieveAll

```
<T> List<T> retrieveAll (Class<T> entityClass)
```

retrieveAll

```
<T> List<T> retrieveAll (String entityClassName)
```

retrieveAll

```
<T> List<T> retrieveAll (Class<T> entityClass, QueryParams queryParams)
```

retrieveAll

```
<T> List<T> retrieveAll (String entityClassName, QueryParams queryParams)
```

11.49.6 MdsSchedulerService

public interface **MdsSchedulerService**

Methods

scheduleRepeatingJob

void **scheduleRepeatingJob** (long *interval*)

unscheduleRepeatingJob

void **unscheduleRepeatingJob** ()

11.49.7 MotechDataService

public interface **MotechDataService**<T>

This is a basic service interface with CRUD operations. Mainly it is used as super interface to create service interface related with the given entity schema in `org.motechproject.mds.builder.EntityInfrastructureBuilder` but it can be also used by other service interfaces inside this package.

Parameters

- <T> – the type of entity schema.

Methods

count

long **count** ()

countForFilter

long **countForFilter** ([Filter](#) *filter*)

create

T **create** (T *object*)

delete

void **delete** (T *object*)

delete

void **delete** ([String](#) *primaryKeyName*, [Object](#) *value*)

deleteAll

void **deleteAll** ()

doInTransaction

<R> R **doInTransaction** ([TransactionCallback](#)<R> *transactionCallback*)

executeQuery

<R> R **executeQuery** ([QueryExecution](#)<R> *queryExecution*)

executeSQLQuery

<R> R **executeSQLQuery** ([SqlQueryExecution](#)<R> *queryExecution*)

filter

List<T> **filter** ([Filter](#) *filter*)

filter

List<T> **filter** ([Filter](#) *filter*, [QueryParams](#) *queryParams*)

findById

T findById (*Long id*)

findTrashInstanceById

T findTrashInstanceById (*Object instanceId, Object entityId*)

getClassType

Class<T> getClassType ()

getDetachedField

Object getDetachedField (*T instance, String fieldName*)

retrieve

T retrieve (*String primaryKeyName, Object value*)

retrieveAll

List<T> retrieveAll ()

retrieveAll

List<T> retrieveAll (*QueryParams queryParams*)

revertFromTrash

void revertFromTrash (*Object newInstance, Object trash*)

update

T update (*T object*)

updateFromTransient

T updateFromTransient (*T transientObject*)

11.49.8 TransactionalMotechDataService

public abstract class **TransactionalMotechDataService**<T> extends [DefaultMotechDataService](#)<T>

The main goal of the `TransactionalMotechDataService` class is to resolve problems with transaction annotations not working for generated lookups. We use the traditional transaction callback instead.

Parameters

- <T> – the type of entity schema.

Methods

count

protected long **count** ([List](#)<[Property](#)> *properties*)

retrieveAll

protected [List](#)<T> **retrieveAll** ([List](#)<[Property](#)> *properties*)

retrieveAll

protected [List](#)<T> **retrieveAll** ([List](#)<[Property](#)> *properties*, [QueryParams](#) *queryParams*)

11.49.9 TrashService

public interface **TrashService**

The `TrashService` provides methods related with the module trash mode (by default the mode is active and it can be turned off by the user).

Methods

countTrashRecords

long **countTrashRecords** ([String](#) *className*)

emptyTrash

void **emptyTrash** ()

Cleans the module trash. All instances in trash should be removed permanently and if they contain any historical data they should also be removed permanently.

This method should only be executed by the job created in the `scheduleEmptyTrashJob()` method.

findTrashById

Object **findTrashById** (Object *instanceId*, Object *entityId*)

Return instance with given id from trash.

Parameters

- **instanceId** – id of instance
- **entityId** – id of instance entity

getInstancesFromTrash

Collection **getInstancesFromTrash** (String *entityName*, QueryParams *queryParams*)

Returns the collection of instances from trash of a certain entity. Returned collection contains only instances that are on the current schema version.

Parameters

- **entityName** – Instances of what entity should be looked for
- **queryParams** – Query parameters such as page number, size of page and sort direction. If null method will return all records in trash.

Returns Collection of instances on the current schema version in trash

isTrashMode

boolean **isTrashMode** ()

Checks if trash mode is active. This method should be used before executing the `moveToTrash(Object, Long)` method to resolve whether the given instance should be moved to trash or removed permanently.

Returns true if delete mode is equal to `org.motechproject.mds.config.DeleteMode.TRASH`; false otherwise.

moveFromTrash

void **moveFromTrash** (Object *newInstance*, Object *trash*)

Sets history for given trashed instance to match the new one and deletes trashed one from trash.

Parameters

- **newInstance** – instance to be returned from trash
- **trash** – trashed instance to be removed

moveToTrash

void **moveToTrash** (Object *instance*, Long *schemaVersion*)

Moves the given instance to the trash. This method should only be executed, when the module trash mode is active.

Parameters

- **instance** – an instance created from the given entity definition.

See also: `.isTrashMode()`

scheduleEmptyTrashJob

void **scheduleEmptyTrashJob** ()

Sets the repeating schedule job that will be executed from time to time. Execution time depends on the value of time value and time unit (defined in `org.motechproject.mds.util.Constants.Config.MODULE_FILE`).

Before scheduling new job, the old one should be unscheduled to prevent the errors.

11.49.10 TypeService

public interface **TypeService**

The `TypeService` is an interface defining available methods to execute various actions on Field Types.

Methods

findType

`TypeDto` **findType** (`Class<?> clazz`)

findValidations

`List<TypeValidation>` **findValidations** (`TypeDto type`, `Class<? extends Annotation> aClass`)

getAllTypes

`List<TypeDto>` **getAllTypes** ()

getType

`Type` **getType** (`TypeValidation validation`)

11.50 org.motechproject.mds.service.impl

11.50.1 EntityServiceImpl

public class **EntityServiceImpl** implements `EntityService`

Default implementation of `org.motechproject.mds.service.EntityService` interface.

Methods

abandonChanges

public void **abandonChanges** (`Long entityId`)

addDisplayedFields

public void **addDisplayedFields** ([EntityDto](#) *entityDto*, [Map](#)<[String](#), [Long](#)> *positions*)

addFields

public void **addFields** ([EntityDto](#) *entityDto*, [Collection](#)<[FieldDto](#)> *fields*)

addFilterableFields

public void **addFilterableFields** ([EntityDto](#) *entityDto*, [Collection](#)<[String](#)> *fieldNames*)

addLookups

public void **addLookups** ([Long](#) *entityId*, [Collection](#)<[LookupDto](#)> *lookups*)

commitChanges

public void **commitChanges** ([Long](#) *entityId*, [String](#) *changesOwner*)

commitChanges

public void **commitChanges** ([Long](#) *entityId*)

createEntity

public [EntityDto](#) **createEntity** ([EntityDto](#) *entityDto*)

deleteEntity

public void **deleteEntity** ([Long](#) *entityId*)

findEntityFieldByName

public [FieldDto](#) **findEntityFieldByName** ([Long](#) *entityId*, [String](#) *name*)

findFieldByName

public [FieldDto](#) **findFieldByName** ([Long](#) *entityId*, [String](#) *name*)

getAdvancedSettings

public [AdvancedSettingsDto](#) **getAdvancedSettings** ([Long](#) *entityId*)

getAdvancedSettings

```
public AdvancedSettingsDto getAdvancedSettings (Long entityId, boolean committed)
```

getCurrentSchemaVersion

```
public Long getCurrentSchemaVersion (String className)
```

getDisplayFields

```
public List<FieldDto> getDisplayFields (Long entityId)
```

getEntitiesWithLookups

```
public List<EntityDto> getEntitiesWithLookups ()
```

getEntity

```
public EntityDto getEntity (Long entityId)
```

getEntityByClassName

```
public EntityDto getEntityByClassName (String className)
```

getEntityDraft

```
public EntityDraft getEntityDraft (Long entityId)
```

getEntityDraft

```
public EntityDraft getEntityDraft (Long entityId, String username)
```

getEntityFields

```
public List<FieldDto> getEntityFields (Long entityId)
```

getEntityForEdit

```
public EntityDto getEntityForEdit (Long entityId)
```

getEntityLookups

```
public List<LookupDto> getEntityLookups (Long entityId)
```

getFields

```
public List<FieldDto> getFields (Long entityId)
```

getLookupByName

```
public LookupDto getLookupByName (Long entityId, String lookupName)
```

listEntities

```
public List<EntityDto> listEntities ()
```

listEntities

```
public List<EntityDto> listEntities (boolean withSecurityCheck)
```

listWorkInProgress

```
public List<EntityDto> listWorkInProgress ()
```

saveDraftEntityChanges

```
public DraftResult saveDraftEntityChanges (Long entityId, DraftData draftData, String username)
```

saveDraftEntityChanges

```
public DraftResult saveDraftEntityChanges (Long entityId, DraftData draftData)
```

setAllEntities

```
public void setAllEntities (AllEntities allEntities)
```

setAllEntityAudits

```
public void setAllEntityAudits (AllEntityAudits allEntityAudits)
```

setAllEntityDrafts

```
public void setAllEntityDrafts (AllEntityDrafts allEntityDrafts)
```

setAllTypes

```
public void setAllTypes (AllTypes allTypes)
```

setBundleContext

public void **setBundleContext** ([BundleContext](#) *bundleContext*)

setEntityValidator

public void **setEntityValidator** ([EntityValidator](#) *entityValidator*)

setMDSConstructor

public void **setMDSConstructor** ([MDSConstructor](#) *mdsConstructor*)

updateComboboxValues

public void **updateComboboxValues** ([Long](#) *entityId*, [Map](#)<[String](#), [Collection](#)> *fieldValuesToUpdate*)

updateDraft

public [EntityDto](#) **updateDraft** ([Long](#) *entityId*)

updateRestOptions

public void **updateRestOptions** ([Long](#) *entityId*, [RestOptionsDto](#) *restOptionsDto*)

11.50.2 JarGeneratorServiceImpl

public class **JarGeneratorServiceImpl** implements [JarGeneratorService](#)

Default implementation of [org.motechproject.mds.service.JarGeneratorService](#) interface.

Methods**generate**

public [File](#) **generate** ()

regenerateMdsDataBundle

public synchronized void **regenerateMdsDataBundle** (boolean *buildDDE*)

regenerateMdsDataBundle

public void **regenerateMdsDataBundle** (boolean *buildDDE*, boolean *startBundle*)

regenerateMdsDataBundleAfterDdeEnhancement

```
public void regenerateMdsDataBundleAfterDdeEnhancement (String moduleName)
```

setAllEntities

```
public void setAllEntities (AllEntities allEntities)
```

setBundleContext

```
public void setBundleContext (BundleContext bundleContext)
```

setMdsConstructor

```
public void setMdsConstructor (MDSConstructor mdsConstructor)
```

setMdsDataProvider

```
public void setMdsDataProvider (MDSDataProvider mdsDataProvider)
```

setMetadataHolder

```
public void setMetadataHolder (MetadataHolder metadataHolder)
```

setMonitor

```
public void setMonitor (EntitiesBundleMonitor monitor)
```

setVelocityEngine

```
public void setVelocityEngine (VelocityEngine velocityEngine)
```

11.50.3 MdsLookupServiceImpl

```
public class MdsLookupServiceImpl implements MDSLookupService
```

Implementation of the `org.motechproject.mds.service.MDSLookupService`. This runs in the MDS context(not entities context). All calls are delegated to the respective data service for the entity.

Methods**count**

```
public long count (Class entityClass, String lookupName, Map<String, ?> lookupParams)
```

count

```
public long count (String entityClassName, String lookupName, Map<String, ?> lookupParams)
```

countAll

```
public long countAll (Class entityClass)
```

countAll

```
public long countAll (String entityClassName)
```

findMany

```
public <T> List<T> findMany (Class<T> entityClass, String lookupName, Map<String, ?> lookupParams)
```

findMany

```
public <T> List<T> findMany (String entityClassName, String lookupName, Map<String, ?> lookupParams)
```

findMany

```
public <T> List<T> findMany (Class<T> entityClass, String lookupName, Map<String, ?> lookupParams, QueryParams queryParams)
```

findMany

```
public <T> List<T> findMany (String entityClassName, String lookupName, Map<String, ?> lookupParams, QueryParams queryParams)
```

findOne

```
public <T> T findOne (Class<T> entityClass, String lookupName, Map<String, ?> lookupParams)
```

findOne

```
public <T> T findOne (String entityClassName, String lookupName, Map<String, ?> lookupParams)
```

retrieveAll

```
public <T> List<T> retrieveAll (Class<T> entityClass)
```

retrieveAll

```
public <T> List<T> retrieveAll (String entityClassName)
```

retrieveAll

```
public <T> List<T> retrieveAll (Class<T> entityClass, QueryParams queryParams)
```

retrieveAll

```
public <T> List<T> retrieveAll (String entityClassName, QueryParams queryParams)
```

11.50.4 MdsScheduledJob

```
public class MdsScheduledJob implements Job  
    Job responsible for emptying MDS trash.
```

Methods**execute**

```
public void execute (JobExecutionContext jobExecutionContext)
```

11.50.5 MdsSchedulerServiceImpl

```
public class MdsSchedulerServiceImpl implements MdsSchedulerService
```

Fields**DEFAULT_WAIT_TIME**

```
public static final int DEFAULT_WAIT_TIME
```

JOB_GROUP_NAME

```
public static final String JOB_GROUP_NAME
```

MAX_REPEAT_COUNT

```
public static final int MAX_REPEAT_COUNT
```

RETRIEVAL_RETRIES_COUNT

```
public static final int RETRIEVAL_RETRIES_COUNT
```


SCHEDULER_SYMBOLIC_NAME

public static final [String](#) **SCHEDULER_SYMBOLIC_NAME**

Constructors

MdsSchedulerServiceImpl

public **MdsSchedulerServiceImpl** ([BundleContext](#) *bundleContext*)

Methods

scheduleRepeatingJob

public void **scheduleRepeatingJob** (long *interval*)

unscheduleRepeatingJob

public void **unscheduleRepeatingJob** ()

11.50.6 TypeServiceImpl

public class **TypeServiceImpl** implements [TypeService](#)

Default implementation of [org.motechproject.mds.service.TypeService](#) interface

Methods

findType

public [TypeDto](#) **findType** ([Class](#)<?> *clazz*)

findValidations

public [List](#)<[TypeValidation](#)> **findValidations** ([TypeDto](#) *type*, [Class](#)<? extends [Annotation](#)> *aClass*)

getAllTypes

public [List](#)<[TypeDto](#)> **getAllTypes** ()

getType

public [Type](#) **getType** ([TypeValidation](#) *validation*)

setAllTypeValidations

```
public void setAllTypeValidations (AllTypeValidations allTypeValidations)
```

setAllTypes

```
public void setAllTypes (AllTypes allTypes)
```

11.51 org.motechproject.mds.service.impl.history

11.51.1 BasePersistenceService

```
public abstract class BasePersistenceService
```

The `BasePersistenceService` class provides utility methods for communication with the database for `HistoryServiceImpl` and `TrashServiceImpl`. It allows to create and retrieve instances, load proper classes and parse values.

Methods**create**

```
protected <T> Object create (Class<T> clazz, Object instance, EntityType type, ValueGetter valueGetter)
```

create

```
protected <T> Object create (Class<T> clazz, Object instance, EntityType type, ValueGetter valueGetter,  
                             ObjectReference objectReference)
```

getAllEntities

```
protected AllEntities getAllEntities ()
```

getBundleContext

```
protected BundleContext getBundleContext ()
```

getCurrentSchemaVersion

```
protected Long getCurrentSchemaVersion (String className)
```

getEntities

```
protected List<Entity> getEntities ()
```

getEntity

protected [Entity](#) **getEntity** ([Long](#) *id*)

getEntitySchemaVersion

protected [Long](#) **getEntitySchemaVersion** ([Object](#) *src*)

getInstanceId

protected [Long](#) **getInstanceId** ([Object](#) *instance*)

getPersistenceManagerFactory

protected [PersistenceManagerFactory](#) **getPersistenceManagerFactory** ()

setAllEntities

public void **setAllEntities** ([AllEntities](#) *allEntities*)

setBundleContext

public void **setBundleContext** ([BundleContext](#) *bundleContext*)

setPersistenceManagerFactory

public void **setPersistenceManagerFactory** ([PersistenceManagerFactory](#) *persistenceManagerFactory*)

11.51.2 HistoryServiceImpl

public class **HistoryServiceImpl** extends [BasePersistenceService](#) implements [HistoryService](#)
Default implementation of [org.motechproject.mds.service.HistoryService](#) interface.

Methods**countHistoryRecords**

public long **countHistoryRecords** ([Object](#) *instance*)

getHistoryForInstance

public [List](#) **getHistoryForInstance** ([Object](#) *instance*, [QueryParams](#) *queryParams*)

getSingleHistoryInstance

```
public Object getSingleHistoryInstance (Object instance, Long historyId)
```

record

```
public void record (Object instance)
```

remove

```
public void remove (Object instance)
```

setTrashFlag

```
public void setTrashFlag (Object instance, Object trash, boolean flag)
```

11.51.3 HistoryTrashClassHelper

```
public final class HistoryTrashClassHelper
```

Contains utility methods for dealing with history and trash classes

Methods**currentVersion**

```
public static String currentVersion (Class<?> historyClass)
```

getClass

```
public static Class<?> getClass (Object src, EntityType type, BundleContext bundleContext)
```

getClass

```
public static Class<?> getClass (String srcClassName, EntityType type, BundleContext bundleContext)
```

getInstanceClassName

```
public static String getInstanceClassName (Object instance)
```

getMethodParameterType

```
public static String getMethodParameterType (Type type, ComboBoxHolder holder)
```

schemaVersion

```
public static String schemaVersion (Class<?> historyClass)
```

trashFlag

```
public static String trashFlag (Class<?> historyClass)
```

11.51.4 TrashServiceImpl

public class **TrashServiceImpl** extends **BasePersistenceService** implements **TrashService**
Default implementation of `org.motechproject.mds.service.TrashService` interface.

Methods**countTrashRecords**

```
public long countTrashRecords (String className)
```

emptyTrash

```
public void emptyTrash ()
```

findTrashById

```
public Object findTrashById (Object instanceId, Object entityId)
```

getInstancesFromTrash

```
public Collection getInstancesFromTrash (String className, QueryParams queryParams)
```

init

```
public void init ()
```

isTrashMode

```
public boolean isTrashMode ()
```

moveFromTrash

```
public void moveFromTrash (Object newInstance, Object trash)
```

moveToTrash

public void **moveToTrash** (*Object instance*, *Long entityVersion*)

scheduleEmptyTrashJob

public void **scheduleEmptyTrashJob** ()

setHistoryService

public void **setHistoryService** (*HistoryService historyService*)

setMdsSchedulerService

public void **setMdsSchedulerService** (*MdsSchedulerService mdsSchedulerService*)

setSettingsService

public void **setSettingsService** (*SettingsService settingsService*)

11.51.5 ValueGetter

public class **ValueGetter**

This class is required for retrieving values from entities. Since the history implementation makes additional changes to records involved in relationships and sets additional field, this class gives it an easy way to override the default behaviour.

Constructors

ValueGetter

public **ValueGetter** (*BasePersistenceService persistenceService*, *BundleContext bundleContext*)

Methods

findService

protected *MotechDataService* **findService** (*Class<?> clazz*)

getRelationshipValue

protected *Object* **getRelationshipValue** (*ObjectReference reference*)

getValue

public Object **getValue** (Field *field*, Object *instance*, Object *recordInstance*, EntityType *type*, ObjectReference *objectReference*)

resolveManyToManyRelationship

public Object **resolveManyToManyRelationship** (Field *field*, Object *instance*, Object *recordInstance*)

updateRecordFields

protected void **updateRecordFields** (Object *newHistoryRecord*, Object *realCurrentObj*)

Updates fields of a new record after it is created. Called for newly created relationship records as well.

Parameters

- **newHistoryRecord** – the newly created record
- **realCurrentObj** – the actual current object

11.52 org.motechproject.mds.util

11.52.1 ClassName

public final class **ClassName**

The `ClassName` util provides several methods which should help for example with getting class name or package from string representation of class. There is also methods related with creating names for repository, service interface and implementation of this service interface.

Methods

getEntityName

public static String **getEntityName** (String *className*)

getEntityTypeSuffix

public static String **getEntityTypeSuffix** (String *name*)

getHistoryClassName

public static String **getHistoryClassName** (String *className*)

getInterfaceName

public static String **getInterfaceName** (String *className*)

getPackage

```
public static String getPackage (String className)
```

getRepositoryName

```
public static String getRepositoryName (String className)
```

getServiceName

```
public static String getServiceName (String className)
```

getSimpleName

```
public static String getSimpleName (String className)
```

getTrashClassName

```
public static String getTrashClassName (String className)
```

isHistoryClassName

```
public static boolean isHistoryClassName (String className)
```

isTrashClassName

```
public static boolean isTrashClassName (String className)
```

moduleNameForRest

```
public static String moduleNameForRest (String moduleName)
```

restId

```
public static String restId (String entityName, String module, String namespace)
```

trimTrashHistorySuffix

```
public static String trimTrashHistorySuffix (String name)
```

11.52.2 Constants

```
public final class Constants
```

The `Constants` contains constant values used in MDS module. They are grouped by their role.

11.52.3 Constants.AnnotationFields

public static final class **AnnotationFields**

The AnnotationFields contains constant values related with attributes names in mds annotations.

See also: `org.motechproject.mds.annotations.Entity`, `org.motechproject.mds.annotations.Field`, `org.motechproject.mds.annotations.Ignore`, `org.motechproject.mds.annotations.Lookup`, `org.motechproject.mds.annotations.LookupField`

Fields

DELETE

public static final `String` **DELETE**

Constant DELETE corresponding to the attribute name delete

DISPLAY_NAME

public static final `String` **DISPLAY_NAME**

Constant DISPLAY_NAME corresponding to the primitive value displayName

FRACTION

public static final `String` **FRACTION**

Constant FRACTION corresponding to the primitive value fraction

INTEGER

public static final `String` **INTEGER**

Constant INTEGER corresponding to the primitive value integer

MAX

public static final `String` **MAX**

Constant MAX corresponding to the primitive value max

MIN

public static final `String` **MIN**

Constant MIN corresponding to the primitive value min

MODULE

public static final `String` **MODULE**

Constant MODULE corresponding to the attribute name module

NAME

public static final [String](#) **NAME**
Constant NAME corresponding to the attribute name name

NAMESPACE

public static final [String](#) **NAMESPACE**
Constant NAMESPACE corresponding to the attribute name namespace

PERSIST

public static final [String](#) **PERSIST**
Constant PERSIST corresponding to the attribute name persist

REGEXP

public static final [String](#) **REGEXP**
Constant REGEXP corresponding to the primitive value regexp

UPDATE

public static final [String](#) **UPDATE**
Constant UPDATE corresponding to the attribute name update

VALUE

public static final [String](#) **VALUE**
Constant VALUE corresponding to the primitive value value

11.52.4 Constants.BundleNames

public static final class **BundleNames**
The names of the mds bundles.

Fields

MDS_BUNDLE_NAME

public static final [String](#) **MDS_BUNDLE_NAME**

MDS_BUNDLE_SYMBOLIC_NAME

public static final [String](#) **MDS_BUNDLE_SYMBOLIC_NAME**

MDS_ENTITIES_NAME

```
public static final String MDS_ENTITIES_NAME
```

MDS_ENTITIES_SYMBOLIC_NAME

```
public static final String MDS_ENTITIES_SYMBOLIC_NAME
```

MDS_MIGRATION_NAME

```
public static final String MDS_MIGRATION_NAME
```

MDS_MIGRATION_SYMBOLIC_NAME

```
public static final String MDS_MIGRATION_SYMBOLIC_NAME
```

SYMBOLIC_NAME_PREFIX

```
public static final String SYMBOLIC_NAME_PREFIX
```

11.52.5 Constants.Config

```
public static final class Config
```

The `Config` contains constant values related with properties inside files:

- `datanucleus.properties`
- `motech-mds.properties`

Fields

DATANUCLEUS_FILE

```
public static final String DATANUCLEUS_FILE
```

Constant `DATANUCLEUS_FILE` presents the file name with configuration for datanucleus.

EMPTY_TRASH_JOB

```
public static final String EMPTY_TRASH_JOB
```

Constant `EMPTY_TRASH_JOB` presents a name of job scheduled by scheduler module.

MDS_DELETE_MODE

```
public static final String MDS_DELETE_MODE
```

Constant `MDS_DELETE_MODE` presents what should happen with objects when there are deleted. They can be deleted permanently or moved to the trash. The following values are valid for this property:

- `delete`

- trash

MDS_EMPTY_TRASH

public static final [String](#) **MDS_EMPTY_TRASH**

The boolean property that specifies if the trash should be empty after some time.

See also: `.MDS_DELETE_MODE`, `.MDS_TIME_VALUE`, `.MDS_TIME_UNIT`

MDS_TIME_UNIT

public static final [String](#) **MDS_TIME_UNIT**

The property that specifies what time unit should be used to specify time when trash should be cleaned. The following values are valid for this property:

- Hours
- Days
- Weeks
- Months
- Years

See also: `.MDS_DELETE_MODE`, `.MDS_EMPTY_TRASH`, `.MDS_TIME_VALUE`

MDS_TIME_VALUE

public static final [String](#) **MDS_TIME_VALUE**

The integer property that specifies after what time (according with correct time unit) trash should be cleaned.

See also: `.MDS_DELETE_MODE`, `.MDS_EMPTY_TRASH`, `.MDS_TIME_UNIT`

MODULE_FILE

public static final [String](#) **MODULE_FILE**

Constant `MODULE_FILE` presents the file name with configuration for MDS module.

11.52.6 Constants.Manifest

public static final class **Manifest**

The `Manifest` contains constant values related with attributes inside the `motech-platform-dataservices-entities` bundle manifest.

See also: `org.motechproject.mds.service.JarGeneratorService`,
`org.motechproject.mds.service.impl.JarGeneratorServiceImpl`

Fields

BUNDLE_MANIFESTVERSION

public static final `String` **BUNDLE_MANIFESTVERSION**
Constant BUNDLE_MANIFESTVERSION presents a version of bundle manifest.

BUNDLE_NAME_SUFFIX

public static final `String` **BUNDLE_NAME_SUFFIX**
Constant BUNDLE_NAME_SUFFIX presents suffix of the name of bundle that will be created by implementation of `org.motechproject.mds.service.JarGeneratorService` interface.

MANIFEST_VERSION

public static final `String` **MANIFEST_VERSION**
Constant MANIFEST_VERSION presents a version of jar manifest.

SYMBOLIC_NAME_SUFFIX

public static final `String` **SYMBOLIC_NAME_SUFFIX**
Constant SYMBOLIC_NAME_SUFFIX presents suffix of the bundle symbolic name of bundle that will be created by implementation of `org.motechproject.mds.service.JarGeneratorService` interface.

11.52.7 Constants.MetadataKeys

public static final class **MetadataKeys**
The keys used in fields metadata

Fields

DATABASE_COLUMN_NAME

public static final `String` **DATABASE_COLUMN_NAME**

ENUM_CLASS_NAME

public static final `String` **ENUM_CLASS_NAME**

MAP_KEY_TYPE

public static final `String` **MAP_KEY_TYPE**

MAP_VALUE_TYPE

public static final `String` **MAP_VALUE_TYPE**

OWNING_SIDE

public static final [String](#) **OWNING_SIDE**

RELATED_CLASS

public static final [String](#) **RELATED_CLASS**

RELATED_FIELD

public static final [String](#) **RELATED_FIELD**

11.52.8 Constants.Operators

public static final class **Operators**
Operators that users can use in lookups.

Fields

ENDS_WITH

public static final [String](#) **ENDS_WITH**

EQ

public static final [String](#) **EQ**

EQ_IGNORE_CASE

public static final [String](#) **EQ_IGNORE_CASE**

GT

public static final [String](#) **GT**

GT_EQ

public static final [String](#) **GT_EQ**

LT

public static final [String](#) **LT**

LT_EQ

public static final `String` **LT_EQ**

MATCHES

public static final `String` **MATCHES**

NEQ

public static final `String` **NEQ**

STARTS_WITH

public static final `String` **STARTS_WITH**

11.52.9 Constants.Packages

public static final class **Packages**

The `Packages` contains constant values related with packages inside MDS module.

Fields**BASE**

public static final `String` **BASE**

Constant `BASE` presents the base package for all packages inside MDS module.

ENTITY

public static final `String` **ENTITY**

Constant `ENTITY` presents a package for entity classes.

See also: `.BASE`

REPOSITORY

public static final `String` **REPOSITORY**

Constant `REPOSITORY` presents a package for repository classes.

See also: `.BASE`

SERVICE

public static final `String` **SERVICE**

Constant `SERVICE` presents a package for service interfaces.

See also: `.BASE`

SERVICE_IMPL

public static final [String](#) **SERVICE_IMPL**

Constant `SERVICE_IMPL` presents a package for implementation of interfaces defined in [SERVICE](#) package.

See also: `.BASE`, `.SERVICE`

11.52.10 Constants.PackagesGenerated

public static final class **PackagesGenerated**

Fields

ENTITY

public static final [String](#) **ENTITY**

Constant `ENTITY` presents a package for generated entity classes.

REPOSITORY

public static final [String](#) **REPOSITORY**

Constant `REPOSITORY` presents a package for generated repository classes.

See also: `.ENTITY`

SERVICE

public static final [String](#) **SERVICE**

Constant `SERVICE` presents a package for generated service interfaces.

See also: `.ENTITY`

SERVICE_IMPL

public static final [String](#) **SERVICE_IMPL**

Constant `SERVICE_IMPL` presents a package for generated implementation of interfaces defined in [SERVICE](#) package.

See also: `.SERVICE`

11.52.11 Constants.Roles

public static final class **Roles**

The `Roles` contains constant values related with security roles.

Fields

DATA_ACCESS

public static final [String](#) **DATA_ACCESS**

Users with ‘Data Access’ have the ability to view the Data Browser tab. From that tab then can search for objects within the system, view and modify the data stored in the system.

HAS_ANY_MDS_ROLE

public static final [String](#) **HAS_ANY_MDS_ROLE**

Spring security el expression to check if the given user has any of the MDS roles.

See also: `.SCHEMA_ACCESS`, `.SETTINGS_ACCESS`, `.DATA_ACCESS`

HAS_DATA_ACCESS

public static final [String](#) **HAS_DATA_ACCESS**

Spring security el expression to check if the given user has the ‘Data Access’ role.

See also: `.DATA_ACCESS`

HAS_DATA_OR_SCHEMA_ACCESS

public static final [String](#) **HAS_DATA_OR_SCHEMA_ACCESS**

Spring security el expression to check if the given user has the ‘Schema Access’ or ‘Data Access’ roles.

See also: `.SCHEMA_ACCESS`, `.DATA_ACCESS`

HAS_SCHEMA_ACCESS

public static final [String](#) **HAS_SCHEMA_ACCESS**

Spring security el expression to check if the given user has the ‘Schema Access’ role.

See also: `.SCHEMA_ACCESS`

HAS_SETTINGS_ACCESS

public static final [String](#) **HAS_SETTINGS_ACCESS**

Spring security el expression to check if the given user has the ‘Settings Access’ role.

See also: `.SETTINGS_ACCESS`

SCHEMA_ACCESS

public static final [String](#) **SCHEMA_ACCESS**

Users with ‘Schema Access’ have the ability to view the Schema Editor tab of the UI. Then can add new objects, delete existing objects and modify the fields on existing objects.

SETTINGS_ACCESS

public static final [String](#) **SETTINGS_ACCESS**

Users with ‘Settings Access’ have the ability to view the Settings tab. From that tab then can modify data retention policies as well as import and export schema and data.

11.52.12 Constants.Settings

public static final class **Settings**

Keys for entity settings.

Fields

ALLOW_MULTIPLE_SELECTIONS

public static final [String](#) **ALLOW_MULTIPLE_SELECTIONS**

ALLOW_USER_SUPPLIED

public static final [String](#) **ALLOW_USER_SUPPLIED**

COMBOBOX_VALUES

public static final [String](#) **COMBOBOX_VALUES**

STRING_MAX_LENGTH

public static final [String](#) **STRING_MAX_LENGTH**

11.52.13 Constants.Util

public static final class **Util**

The `Util` contains constant values to help avoid string literal repetition.

See also: [pmd](#)

Fields

AUTO_GENERATED

public static final [String](#) **AUTO_GENERATED**

AUTO_GENERATED_EDITABLE

public static final [String](#) **AUTO_GENERATED_EDITABLE**

CREATION_DATE_DISPLAY_FIELD_NAME

```
public static final String CREATION_DATE_DISPLAY_FIELD_NAME
```

CREATION_DATE_FIELD_NAME

```
public static final String CREATION_DATE_FIELD_NAME
```

CREATOR_DISPLAY_FIELD_NAME

```
public static final String CREATOR_DISPLAY_FIELD_NAME
```

CREATOR_FIELD_NAME

```
public static final String CREATOR_FIELD_NAME
```

DATANUCLEUS

```
public static final String DATANUCLEUS
```

ENTITY

```
public static final String ENTITY
```

Constant ENTITY corresponding to the field name of the class that want to create a bidirectional connection with instane of `org.motechproject.mds.domain.Entity`

FALSE

```
public static final String FALSE
```

Constant FALSE corresponding to the primitive value `false`

GENERATED_FIELD_NAMES

```
public static final String[] GENERATED_FIELD_NAMES
```

ID_DISPLAY_FIELD_NAME

```
public static final String ID_DISPLAY_FIELD_NAME
```

ID_FIELD_NAME

```
public static final String ID_FIELD_NAME
```

MODIFICATION_DATE_DISPLAY_FIELD_NAME

public static final `String` **MODIFICATION_DATE_DISPLAY_FIELD_NAME**

MODIFICATION_DATE_FIELD_NAME

public static final `String` **MODIFICATION_DATE_FIELD_NAME**

MODIFIED_BY_DISPLAY_FIELD_NAME

public static final `String` **MODIFIED_BY_DISPLAY_FIELD_NAME**

MODIFIED_BY_FIELD_NAME

public static final `String` **MODIFIED_BY_FIELD_NAME**

OWNER_DISPLAY_FIELD_NAME

public static final `String` **OWNER_DISPLAY_FIELD_NAME**

OWNER_FIELD_NAME

public static final `String` **OWNER_FIELD_NAME**

SQL_QUERY

public static final `String` **SQL_QUERY**

TRUE

public static final `String` **TRUE**
Constant TRUE corresponding to the primitive value `true`

VALUE_GENERATOR

public static final `String` **VALUE_GENERATOR**

11.52.14 EnumHelper

public final class **EnumHelper**
This is a helper class, used while generating enums in MDS.

Methods

prefixEnumValue

public static `String` **prefixEnumValue** (`String` *value*)

Prefixes enum values, if they start with illegal character (other than letter, dollar sign or underscore)

Parameters

- **value** – an enum value

Returns either the same value, if the value is legal, or value prefixed with underscore, if the value is illegal

prefixEnumValues

public static `List<String>` **prefixEnumValues** (`List<String>` *values*)

Prefixes a list of enum values. For each value in the list, `prefixEnumValue(java.lang.String)` is called.

Parameters

- **values** – a list of enum values

Returns a list of prefixed values

11.52.15 FieldHelper

public final class **FieldHelper**

Utility class handling dynamic setting of field values

Methods

setField

public static void **setField** (`Object` *current*, `String` *path*, `List` *value*)

11.52.16 InstanceSecurityRestriction

public class **InstanceSecurityRestriction**

Represents a restriction on entity instances

Methods

isByCreator

public boolean **isByCreator** ()

isByOwner

public boolean **isByOwner** ()

isEmpty

```
public boolean isEmpty ()
```

setByCreator

```
public void setByCreator (boolean byCreator)
```

setByOwner

```
public void setByOwner (boolean byOwner)
```

11.52.17 Loader

```
public abstract class Loader<T>
```

The **Loader** is an abstract class that checks if all class dependencies to the given class definition are resolved. If not then the missing class name is taken from exception and the `doWhenClassNotFound(String)` method is executed.

Parameters

- `<T>` – the type of argument data

Methods**doWhenClassNotFound**

```
public abstract void doWhenClassNotFound (String name)
```

getClassDefinition

```
public abstract Class<?> getClassDefinition (T arg)
```

loadClass

```
public Class<?> loadClass (T arg)
```

11.52.18 LookupName

```
public final class LookupName
```

Utility class for dealing with lookup names.

Methods**lookupCountMethod**

```
public static String lookupCountMethod (String lookupNameOrMethodName)
```

lookupMethod

public static [String](#) **lookupMethod** ([String](#) *lookupName*)

11.52.19 MDSCClassLoader

public class **MDSCClassLoader** extends [ClassLoader](#)

The MDSCClassLoader class is a mds wrapper for [ClassLoader](#).

Constructors

MDSCClassLoader

protected **MDSCClassLoader** ()

MDSCClassLoader

protected **MDSCClassLoader** ([ClassLoader](#) *parent*)

Methods

defineClass

public [Class](#)<?> **defineClass** ([String](#) *name*, byte[] *bytecode*)

getInstance

public static [MDSCClassLoader](#) **getInstance** ()

getStandaloneInstance

public static [MDSCClassLoader](#) **getStandaloneInstance** ()

getStandaloneInstance

public static [MDSCClassLoader](#) **getStandaloneInstance** ([ClassLoader](#) *parent*)

reloadClassLoader

public static void **reloadClassLoader** ()

safeDefineClass

public [Class](#)<?> **safeDefineClass** ([String](#) *name*, byte[] *bytecode*)

11.52.20 MdsBundleHelper

public final class **MdsBundleHelper**

Methods

findMdsBundle

public static [Bundle](#) **findMdsBundle** ([BundleContext](#) *bundleContext*)

findMdsEntitiesBundle

public static [Bundle](#) **findMdsEntitiesBundle** ([BundleContext](#) *bundleContext*)

isBundleMdsDependent

public static boolean **isBundleMdsDependent** ([Bundle](#) *bundle*)

isFrameworkBundle

public static boolean **isFrameworkBundle** ([Bundle](#) *bundle*)

isMdsBundle

public static boolean **isMdsBundle** ([Bundle](#) *bundle*)

isMdsEntitiesBundle

public static boolean **isMdsEntitiesBundle** ([Bundle](#) *bundle*)

11.52.21 MemberUtil

public final class **MemberUtil**

Fields

BOOLEAN_GETTER_PREFIX

public static final [String](#) **BOOLEAN_GETTER_PREFIX**

GETTER_PREFIX

public static final [String](#) **GETTER_PREFIX**

GET_OR_SET_END_INDEX

public static final int **GET_OR_SET_END_INDEX**

IS_END_INDEX

public static final int **IS_END_INDEX**

SETTER_PREFIX

public static final String **SETTER_PREFIX**

Methods

getCorrectType

public static Class<?> **getCorrectType** (AnnotatedElement *object*)

getCorrectType

public static Class<?> **getCorrectType** (Member *object*)

getDeclaringClass

public static Class<?> **getDeclaringClass** (AccessibleObject *ac*)

getFieldAndAccessorsForElement

public static List<AccessibleObject> **getFieldAndAccessorsForElement** (AccessibleObject *ao*)

getFieldName

public static String **getFieldName** (AnnotatedElement *object*)

getFieldName

public static String **getFieldName** (Member *object*)

getFieldNameFromGetterSetterName

public static String **getFieldNameFromGetterSetterName** (String *getterSetterName*)

getGenericType

public static Class<?> **getGenericType** (AnnotatedElement *object*)

getGenericType

```
public static Class<?> getGenericType (AnnotatedElement object, int typeNumber)
```

getGenericType

```
public static Class<?> getGenericType (Member object, int typeNumber)
```

getMembers

```
public static List<Member> getMembers (Class<?> clazz, Predicate memberPredicate)
```

isGetter

```
public static boolean isGetter (Member member)
```

isSetter

```
public static boolean isSetter (Member member)
```

11.52.22 NumberPredicate

```
public class NumberPredicate implements Predicate
```

Constructors**NumberPredicate**

```
public NumberPredicate (Number element)
```

Methods**evaluate**

```
public boolean evaluate (Object candidate)
```

11.52.23 ObjectReference

```
public class ObjectReference
```

Represents an object reference. It holds an information about related field name, as well as the object that the field should reference to.

Constructors

ObjectReference

```
public ObjectReference (String fieldName, Object reference, String mappingFieldName)
```

Methods

getFieldName

```
public String getFieldName ()
```

getMappingFieldName

```
public String getMappingFieldName ()
```

getReference

```
public Object getReference ()
```

setFieldName

```
public void setFieldName (String fieldName)
```

setMappingFieldName

```
public void setMappingFieldName (String mappingFieldName)
```

setReference

```
public void setReference (Object reference)
```

11.52.24 Order

```
public class Order implements Serializable  
    Represents an order in a query
```

Constructors

Order

```
public Order (String field)
```

Order

```
public Order (String field, String direction)
```

Order

```
public Order (String field, Direction direction)
```

Methods

getDirection

```
public Direction getDirection ()
```

getField

```
public String getField ()
```

toString

```
public String toString ()
```

11.52.25 Order.Direction

```
public static enum Direction
```

Enum Constants

ASC

```
public static final Order.Direction ASC
```

DESC

```
public static final Order.Direction DESC
```

11.52.26 Pair

```
public interface Pair<N, V>
```

The `Pair` util interface should use everywhere where developer needs a pair of key-value

Parameters

- `<N>` – type of key
- `<V>` – type of value

See also: `org.motechproject.mds.domain.FieldMetadata`, `org.motechproject.mds.domain.FieldSet`, `org.motechproject.mds.dto.MetadataDto`, `org.motechproject.mds.dto.SettingDto`

Methods

`getKey`

`N getKey ()`

`getValue`

`V getValue ()`

11.52.27 PropertyUtil

public final class **PropertyUtil** extends `PropertyUtils`

The `PropertyUtil` util class provides the same method like `org.apache.commons.beanutils.PropertyUtils` and two additional methods for safe writing and reading property in the given bean.

Methods

`copyPropertiesFromTransient`

public static void **copyPropertiesFromTransient** (`Object objFromDb`, `Object transientObj`)

`getPropertyDescriptors`

public static `PropertyDescriptor[]` **getPropertyDescriptors** (`Object bean`)

`safeGetProperty`

public static `Object` **safeGetProperty** (`Object bean`, `String name`)

`safeSetProperty`

public static void **safeSetProperty** (`Object bean`, `String name`, `Object value`)

11.52.28 SecurityMode

public enum **SecurityMode**

This enum describes security mode for an entity

Enum Constants

CREATOR

public static final [SecurityMode](#) **CREATOR**

EVERYONE

public static final [SecurityMode](#) **EVERYONE**

OWNER

public static final [SecurityMode](#) **OWNER**

ROLES

public static final [SecurityMode](#) **ROLES**

USERS

public static final [SecurityMode](#) **USERS**

11.52.29 SecurityUtil

public final class **SecurityUtil**

The `SecurityUtil` class provides helper methods to retrieve logged user details, such as username or roles

Methods

`getUserRoles`

public static [List<String>](#) **getUserRoles** ()

`getUsername`

public static [String](#) **getUsername** ()

11.52.30 ServiceUtil

public final class **ServiceUtil**

The `ServiceUtil` class provides utility methods to work with services

Methods

getServiceForInterfaceName

```
public static <S> S getServiceForInterfaceName (BundleContext bundleContext, String interface-  
Name)
```

11.52.31 TypeHelper

```
public final class TypeHelper
```

A helper class for parsing and formatting mds supported types.

Methods

breakString

```
public static String[] breakString (String str)
```

breakString

```
public static String[] breakString (String str, String[] removes, String[] search, String[] replacement, String  
separator)
```

breakStringForList

```
public static String[] breakStringForList (String str)
```

format

```
public static String format (Object obj)
```

getPrimitive

```
public static Class<?> getPrimitive (Class<?> clazz)
```

getWrapperForPrimitive

```
public static Class<?> getWrapperForPrimitive (Class<?> clazz)
```

hasPrimitive

```
public static boolean hasPrimitive (Class<?> clazz)
```

isPrimitive

```
public static boolean isPrimitive (Class<?> clazz)
```

isPrimitive

public static boolean **isPrimitive** (*String className*)

parse

public static Object **parse** (*Object val*, *Class<?> toClass*)

parse

public static Object **parse** (*Object val*, *String toClass*)

parse

public static Object **parse** (*Object val*, *String toClass*, *ClassLoader classLoader*)

parse

public static Object **parse** (*Object val*, *String toClass*, *String genericType*)

parse

public static Object **parse** (*Object val*, *String toClass*, *String genericType*, *ClassLoader classLoader*)

parseIntToBool

public static boolean **parseIntToBool** (*Integer val*)

parseList

public static List **parseList** (*List val*, *Class<?> generic*)

parseNumber

public static Number **parseNumber** (*Object val*, *String toClass*)

parseString

public static Object **parseString** (*String str*, *Class<?> toClass*)

parseString

public static Object **parseString** (*String str*, *String toClass*)

parseString

public static **Object** **parseString** (*String str*, *Class<?> toClass*, *Class<?> generic*)

parseStringToMap

public static **Map** **parseStringToMap** (*String str*)

toRange

public static **Range** **toRange** (*Object object*, *String typeClass*)

toSet

public static **Set** **toSet** (*Object object*, *String typeClass*)

11.52.32 ValidationUtil

public final class **ValidationUtil**
Common validation utils for mds.

Methods

validateNoJavaKeyword

public static void **validateNoJavaKeyword** (*String str*)

11.53 org.motechproject.osgi.web

11.53.1 Activator

public class **Activator** implements *BundleActivator*

Methods

resourceMappings

protected **Map**<*String*, *String*> **resourceMappings** ()

start

public void **start** (*BundleContext context*)

stop

public void **stop** ([BundleContext](#) *context*)

11.53.2 ApplicationContextTracker

public abstract class **ApplicationContextTracker** extends [ServiceTracker](#)

Base class for every class that wishes to track Spring application context. Contains a methods that help with synchronous processing.

Constructors

ApplicationContextTracker

public **ApplicationContextTracker** ([BundleContext](#) *context*)

Methods

contextInvalidOrProcessed

protected boolean **contextInvalidOrProcessed** ([ServiceReference](#) *serviceReference*, [ApplicationContext](#) *applicationContext*)

getLock

protected [Object](#) **getLock** ()

markAsProcessed

protected void **markAsProcessed** ([ApplicationContext](#) *applicationContext*)

removeFromProcessed

protected void **removeFromProcessed** ([ApplicationContext](#) *applicationContext*)

11.53.3 BlueprintActivator

public class **BlueprintActivator** implements [BundleActivator](#)

Methods

start

public void **start** ([BundleContext](#) *context*)

stop

public void **stop** (*BundleContext context*)

11.53.4 BlueprintApplicationContextTracker

public class **BlueprintApplicationContextTracker** extends *ApplicationContextTracker*

The *BlueprintApplicationContextTracker* class tracks application contexts, which are registered as services.

Constructors

BlueprintApplicationContextTracker

public **BlueprintApplicationContextTracker** (*BundleContext context*)

Methods

addingService

public *Object* **addingService** (*ServiceReference serviceReference*)

removedService

public void **removedService** (*ServiceReference reference*, *Object service*)

11.53.5 BundleContextWrapper

public class **BundleContextWrapper** implements *BundleContextAware*

Fields

CONTEXT_SERVICE_NAME

public static final *String* **CONTEXT_SERVICE_NAME**

Constructors

BundleContextWrapper

public **BundleContextWrapper** ()

BundleContextWrapper

public **BundleContextWrapper** (*BundleContext context*)

Methods

getBundleApplicationContext

```
public ApplicationContext getBundleApplicationContext ()
```

getBundleContext

```
public BundleContext getBundleContext ()
```

getCurrentBundleSymbolicName

```
public String getCurrentBundleSymbolicName ()
```

getService

```
public <T> T getService (Class<T> clazz)
```

setBundleContext

```
public void setBundleContext (BundleContext bundleContext)
```

11.53.6 BundleRegister

```
public final class BundleRegister
```

The `BundleRegister` Singleton class is used for recording bundles. This class will help to reconfigure logger's levels.

Methods

addBundle

```
public void addBundle (Bundle bundle)
```

getBundleList

```
public List<Bundle> getBundleList ()
```

getInstance

```
public static BundleRegister getInstance ()
```

11.53.7 BundledJspView

```
public class BundledJspView extends JstlView implements BundleContextAware
```

Methods

render

```
public void render (Map<String, ?> model, HttpServletRequest request, HttpServletResponse response)
```

setBundleContext

```
public void setBundleContext (BundleContext bundleContext)
```

11.53.8 HttpServiceTracker

```
public class HttpServiceTracker extends ServiceTracker
```

Constructors

HttpServiceTracker

```
public HttpServiceTracker (BundleContext context, Map<String, String> resourceMapping)
```

Methods

addingService

```
public Object addingService (ServiceReference serviceReference)
```

removedService

```
public void removedService (ServiceReference ref, Object service)
```

start

```
public void start ()
```

unregister

```
public void unregister ()
```

11.53.9 HttpServiceTrackers

```
public class HttpServiceTrackers
```

Methods

addTrackerFor

public [HttpServiceTracker](#) **addTrackerFor** ([Bundle](#) *bundle*)

isBeingTracked

public boolean **isBeingTracked** ([Bundle](#) *bundle*)

removeTrackerFor

public [HttpServiceTracker](#) **removeTrackerFor** ([Bundle](#) *bundle*)

11.53.10 Log4JBundleLoader

public class **Log4JBundleLoader**

This `Log4JBundleLoader` class is responsible for loading configuration of the loggers from the saved properties or file in bundle (`log4j.xml`).

Methods

checkListContainLogger

public boolean **checkListContainLogger** ([List](#)<[LogMapping](#)> *loggers*, [String](#) *log*)

checkLogXmlConfiguration

public boolean **checkLogXmlConfiguration** ([Document](#) *log4jDoc*)

createLoggerProperties

public [Properties](#) **createLoggerProperties** ([List](#)<[LogMapping](#)> *log*)

loadBundle

public void **loadBundle** ([Bundle](#) *bundle*)

loadLoggerDbConfiguration

public void **loadLoggerDbConfiguration** ()

setLog4jConf

public void **setLog4jConf** ([String](#) *log4jConf*)

11.53.11 ModuleRegistrationData

public class **ModuleRegistrationData**

Object used to registered a module withing the Motech UI system. Represents a module and is used for building the common user interface. All modules that wish to register within the UI system must either expose this class as a spring bean in their application context or manually register it through the `UIFrameworkService` OSGi service.

See also: `UIFrameworkService`

Constructors

ModuleRegistrationData

public **ModuleRegistrationData** ()

ModuleRegistrationData

public **ModuleRegistrationData** (*String moduleName, String url*)

ModuleRegistrationData

public **ModuleRegistrationData** (*String moduleName, Map<String, String> i18n*)

ModuleRegistrationData

public **ModuleRegistrationData** (*String moduleName, String url, List<String> angularModules, Map<String, String> i18n*)

Methods

addAngularModule

public void **addAngularModule** (*String moduleName*)

addI18N

public void **addI18N** (*String fileName, String fileLocation*)

addSubMenu

public void **addSubMenu** (*String url, String label*)

addSubMenu

public void **addSubMenu** (*String url, String label, String roleForAccess*)

equals

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

getAngularModules

public *List<String>* **getAngularModules** ()

getAngularModulesStr

public *String* **getAngularModulesStr** ()

getBundle

public *Bundle* **getBundle** ()

getCriticalMessage

public *String* **getCriticalMessage** ()

getDefaultURL

public *String* **getDefaultURL** ()

getI18n

public *Map<String, String>* **getI18n** ()

getModuleName

public *String* **getModuleName** ()

getResourcePath

public *String* **getResourcePath** ()

getRoleForAccess

public *List<String>* **getRoleForAccess** ()

getSettingsURL

public *String* **getSettingsURL** ()

getSubMenu

```
public Map<String, SubmenuInfo> getSubMenu ()
```

getUrl

```
public String getUrl ()
```

hashCode

```
public int hashCode ()
```

isNeedsAttention

```
public boolean isNeedsAttention ()
```

removeAngularModule

```
public void removeAngularModule (String moduleName)
```

setBundle

```
public void setBundle (Bundle bundle)
```

setCriticalMessage

```
public void setCriticalMessage (String criticalMessage)
```

setDefaultURL

```
public void setDefaultURL (String defaultURL)
```

setModuleName

```
public void setModuleName (String moduleName)
```

setNeedsAttention

```
public void setNeedsAttention (boolean needsAttention)
```

setResourcePath

```
public void setResourcePath (String resourcePath)
```

setRoleForAccess

public void **setRoleForAccess** (*String* *role*)

setRoleForAccess

public void **setRoleForAccess** (*List<String>* *roles*)

setSettingsURL

public void **setSettingsURL** (*String* *settingsURL*)

setSubMenu

public void **setSubMenu** (*Map<String, SubmenuInfo>* *subMenu*)

setUrl

public void **setUrl** (*String* *url*)

subMenuNeedsAttention

public void **subMenuNeedsAttention** (*String* *submenu*)

submenuBackToNormal

public void **submenuBackToNormal** (*String* *submenu*)

11.53.12 MotechOsgiWebApplicationContext

public class **MotechOsgiWebApplicationContext** extends *OsgiBundleXmlApplicationContext* implements *ConfigurableWebA*

Constructors

MotechOsgiWebApplicationContext

public **MotechOsgiWebApplicationContext** ()

Methods

getNamespace

public *String* **getNamespace** ()

getServletConfig

```
public ServletConfig getServletConfig ()
```

getServletContext

```
public ServletContext getServletContext ()
```

isInitialized

```
public boolean isInitialized ()
```

setConfigLocation

```
public void setConfigLocation (String configLocation)
```

setNamespace

```
public void setNamespace (String namespace)
```

setServletConfig

```
public void setServletConfig (ServletConfig servletConfig)
```

setServletContext

```
public void setServletContext (ServletContext servletContext)
```

waitForContext

```
public void waitForContext (int waitTimeInMillis)
```

11.53.13 OsgiDispatcherServlet

```
public class OsgiDispatcherServlet extends DispatcherServlet
```

Constructors**OsgiDispatcherServlet**

```
public OsgiDispatcherServlet (BundleContext bundleContext)
```

OsgiDispatcherServlet

```
public OsgiDispatcherServlet (BundleContext bundleContext, ConfigurableWebApplicationContext
                             configurableWebApplicationContext)
```

Methods

initFrameworkServlet

```
protected void initFrameworkServlet ()
```

postProcessWebApplicationContext

```
protected void postProcessWebApplicationContext (ConfigurableWebApplicationContext wac)
```

11.53.14 OsgiWebApplicationContext

```
public class OsgiWebApplicationContext implements WebApplicationContext
```

Constructors

OsgiWebApplicationContext

```
public OsgiWebApplicationContext (ApplicationContext applicationContext, ConfigurableWebAppli-
                                     cationContext configurableWebApplicationContext)
```

Methods

containsBean

```
public boolean containsBean (String name)
```

containsBeanDefinition

```
public boolean containsBeanDefinition (String beanName)
```

containsLocalBean

```
public boolean containsLocalBean (String name)
```

findAnnotationOnBean

```
public <A extends Annotation> A findAnnotationOnBean (String beanName, Class<A> annotation-
                                                         Type)
```

getAliases

```
public String[] getAliases (String name)
```

getAutowireCapableBeanFactory

```
public AutowireCapableBeanFactory getAutowireCapableBeanFactory ()
```

getBean

```
public Object getBean (String name)
```

getBean

```
public <T> T getBean (String name, Class<T> requiredType)
```

getBean

```
public <T> T getBean (Class<T> requiredType)
```

getBean

```
public Object getBean (String name, Object... args)
```

getBeanDefinitionCount

```
public int getBeanDefinitionCount ()
```

getBeanDefinitionNames

```
public String[] getBeanDefinitionNames ()
```

getBeanNamesForType

```
public String[] getBeanNamesForType (Class<?> type)
```

getBeanNamesForType

```
public String[] getBeanNamesForType (Class<?> type, boolean includeNonSingletons, boolean allowEagerInit)
```

getBeansOfType

```
public <T> Map<String, T> getBeansOfType (Class<T> type)
```

getBeansOfType

```
public <T> Map<String, T> getBeansOfType (Class<T> type, boolean includeNonSingletons, boolean allowEagerInit)
```

getBeansWithAnnotation

```
public Map<String, Object> getBeansWithAnnotation (Class<? extends Annotation> annotationType)
```

getClassLoader

```
public ClassLoader getClassLoader ()
```

getDisplayName

```
public String getDisplayName ()
```

getEnvironment

```
public ConfigurableEnvironment getEnvironment ()
```

getId

```
public String getId ()
```

getMessage

```
public String getMessage (String code, Object[] args, String defaultMessage, Locale locale)
```

getMessage

```
public String getMessage (String code, Object[] args, Locale locale)
```

getMessage

```
public String getMessage (MessageSourceResolvable resolvable, Locale locale)
```

getParent

```
public ApplicationContext getParent ()
```

getParentBeanFactory

```
public BeanFactory getParentBeanFactory ()
```

getResource

```
public Resource getResource (String location)
```

getResources

```
public Resource[] getResources (String locationPattern)
```

getServletContext

```
public ServletContext getServletContext ()
```

getStartupDate

```
public long getStartupDate ()
```

getType

```
public Class<?> getType (String name)
```

isPrototype

```
public boolean isPrototype (String name)
```

isSingleton

```
public boolean isSingleton (String name)
```

isTypeMatch

```
public boolean isTypeMatch (String name, Class<?> targetType)
```

publishEvent

```
public void publishEvent (ApplicationEvent event)
```

11.53.15 SubmenuInfo

```
public class SubmenuInfo
```

- Class to encapsulate information about submenu Links to be shown on UI

Constructors

SubmenuInfo

```
public SubmenuInfo (String url)
```

Methods

getCriticalMessage

```
public String getCriticalMessage ()
```

getRoleForAccess

```
public List<String> getRoleForAccess ()
```

getUrl

```
public String getUrl ()
```

isNeedsAttention

```
public boolean isNeedsAttention ()
```

setCriticalMessage

```
public void setCriticalMessage (String criticalMessage)
```

setNeedsAttention

```
public void setNeedsAttention (boolean needsAttention)
```

setRoleForAccess

```
public void setRoleForAccess (String roleForAccess)
```

setRoleForAccess

```
public void setRoleForAccess (List<String> roleForAccess)
```

setUrl

```
public void setUrl (String url)
```


11.53.16 UIFrameworkService

public interface **UIFrameworkService**

Service responsible for managing the interface. Provides methods for registering/un-registering modules. All modules are represented by `ModuleRegistrationData` objects, either registered directly through this service or automatically by exposing it in their spring context. This service also allows manipulation of module state, by marking given modules as requiring attention on the UI.

Fields

MODULES_WITHOUT_SUBMENU

`String` **MODULES_WITHOUT_SUBMENU**

MODULES_WITHOUT_UI

`String` **MODULES_WITHOUT_UI**

MODULES_WITH_SUBMENU

`String` **MODULES_WITH_SUBMENU**

Methods

getModuleData

`ModuleRegistrationData` **getModuleData** (`String` *moduleName*)

getModuleDataByAngular

`ModuleRegistrationData` **getModuleDataByAngular** (`String` *angularModule*)

getModuleDataByBundle

`ModuleRegistrationData` **getModuleDataByBundle** (`Bundle` *bundle*)

getRegisteredModules

`Map<String, Collection<ModuleRegistrationData>>` **getRegisteredModules** ()

isModuleRegistered

`boolean` **isModuleRegistered** (`String` *moduleName*)

moduleBackToNormal

void **moduleBackToNormal** (*String moduleName*)

moduleBackToNormal

void **moduleBackToNormal** (*String moduleName, String submenu*)

moduleNeedsAttention

void **moduleNeedsAttention** (*String moduleName, String message*)

moduleNeedsAttention

void **moduleNeedsAttention** (*String moduleName, String submenu, String message*)

registerModule

void **registerModule** (*ModuleRegistrationData module*)

unregisterModule

void **unregisterModule** (*String moduleName*)

11.53.17 UIServiceTracker

public class **UIServiceTracker** extends *ServiceTracker*

Constructors

UIServiceTracker

public **UIServiceTracker** (*BundleContext context, ModuleRegistrationData moduleRegistrationData*)

UIServiceTracker

public **UIServiceTracker** (*BundleContextWrapper wrapper, ModuleRegistrationData moduleRegistrationData*)

Methods

addingService

public *Object* **addingService** (*ServiceReference ref*)

removedService

```
public void removedService (ServiceReference ref, Object service)
```

start

```
public void start ()
```

11.53.18 UIServiceTrackers

```
public class UIServiceTrackers
```

Methods**addTrackerFor**

```
public UIServiceTracker addTrackerFor (Bundle bundle, ApplicationContext applicationContext)
```

isBeingTracked

```
public boolean isBeingTracked (Bundle bundle)
```

removeTrackerFor

```
public UIServiceTracker removeTrackerFor (Bundle bundle)
```

11.53.19 WebUIBundleActivator

```
public class WebUIBundleActivator extends org.motechproject.osgi.web.Activator
```

Methods**moduleId**

```
protected String moduleId ()
```

resourceFolder

```
protected String resourceFolder ()
```

resourceMappings

```
protected Map<String, String> resourceMappings ()
```

start

public void **start** (*BundleContext context*)

stop

public void **stop** (*BundleContext context*)

11.54 org.motechproject.osgi.web.domain

11.54.1 LogMapping

public class **LogMapping**

Constructors

LogMapping

public **LogMapping** ()

LogMapping

public **LogMapping** (*String logName*, *String logLevel*)

Methods

equals

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

getLogLevel

public *String* **getLogLevel** ()

getLogName

public *String* **getLogName** ()

hashCode

public int **hashCode** ()

setLogLevel

public void **setLogLevel** (*String logLevel*)

setLogName

public void **setLogName** (*String logName*)

toString

public *String* **toString** ()

11.55 org.motechproject.osgi.web.exception

11.55.1 RenderException

public class **RenderException** extends *Exception*

Constructors**RenderException**

public **RenderException** (*String message*)

RenderException

public **RenderException** (*String message*, *Throwable cause*)

RenderException

public **RenderException** (*Throwable cause*)

11.55.2 ServletRegistrationException

public class **ServletRegistrationException** extends *RuntimeException*

Constructors**ServletRegistrationException**

public **ServletRegistrationException** (*String message*)

ServletRegistrationException

```
public ServletRegistrationException (String message, Throwable cause)
```

ServletRegistrationException

```
public ServletRegistrationException (Throwable cause)
```

11.56 org.motechproject.osgi.web.ext

11.56.1 ApplicationEnvironment

```
public final class ApplicationEnvironment
```

Fields

DEVELOPMENT

```
public static final String DEVELOPMENT
```

ENVIRONMENT

```
public static final String ENVIRONMENT
```

Methods

getEnvironment

```
public static String getEnvironment ()
```

getModulePath

```
public static String getModulePath (BundleName bundleName)
```

isInDevelopmentMode

```
public static boolean isInDevelopmentMode ()
```

11.56.2 BundleName

```
public class BundleName
```

Constructors

BundleName

```
public BundleName (String bundleSymbolicName)
```

Methods

equals

```
public boolean equals (Object bundle)
```

hashCode

```
public int hashCode ()
```

underscore

```
public String underscore ()
```

11.56.3 FileSystemAwareUIHttpContext

```
public class FileSystemAwareUIHttpContext extends UiHttpContext
```

Constructors

FileSystemAwareUIHttpContext

```
public FileSystemAwareUIHttpContext (HttpContext context, String resourceRootDirectoryPath)
```

Methods

getResource

```
public URL getResource (String name)
```

getResourceRootDirectoryPath

```
public String getResourceRootDirectoryPath ()
```

11.56.4 HttpContextFactory

```
public final class HttpContextFactory
```

Methods

getHttpContext

public static [HttpContext](#) **getHttpContext** ([HttpContext](#) *httpContext*, [Bundle](#) *bundle*)

11.56.5 UiHttpContext

public class **UiHttpContext** implements [HttpContext](#)

Constructors

UiHttpContext

public **UiHttpContext** ([HttpContext](#) *context*)

Methods

getContext

protected [HttpContext](#) **getContext** ()

getMimeType

public [String](#) **getMimeType** ([String](#) *name*)

getResource

public [URL](#) **getResource** ([String](#) *name*)

handleSecurity

public boolean **handleSecurity** ([HttpServletRequest](#) *request*, [HttpServletResponse](#) *response*)

11.57 org.motechproject.osgi.web.service

11.57.1 ServerLogService

public interface **ServerLogService**

Interface for accessing Logger's configuration from the saved properties

Fields

ROOT_LOGGER_NAME

`String` **ROOT_LOGGER_NAME**

Methods

changeLogLevel

void **changeLogLevel** (`String` *name*, `String` *level*)
Changes the logging level for one, specified logger.

Parameters

- **name** – name of the logger you wish to change level for
- **level** – a new level for the logger

changeRootLogLevel

void **changeRootLogLevel** (`String` *level*)
Changes logging level for the root logger

Parameters

- **level** – a new logging level for root logger

getAllLogMappings

`List<LogMapping>` **getAllLogMappings** ()
Returns details for all loggers. Root logger included.

Returns all loggers details

getLogLevels

`List<LogMapping>` **getLogLevels** ()
Returns details for all loggers. Root logger is NOT included.

Returns all loggers details

getRootLogLevel

`LogMapping` **getRootLogLevel** ()
Returns the logger details for the root logger

Returns root logger details

reconfigure

void **reconfigure** ()
Sets the logger properties from scratch.

removeLogger

void **removeLogger** (*String name*)
Removes specified logger.

Parameters

- **name** – name of the logger you wish to remove

11.58 org.motechproject.osgi.web.settings

11.58.1 Loggers

public class **Loggers**
Loggers class that holds information about all loggers in our system.

Constructors

Loggers

public **Loggers** ()

Loggers

public **Loggers** (*List<LogMapping> loggers*, *LogMapping root*)

Methods

equals

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

getLoggers

public *List<LogMapping>* **getLoggers** ()

getRoot

public *LogMapping* **getRoot** ()

getTrash

```
public List<LogMapping> getTrash ()
```

hashCode

```
public int hashCode ()
```

setLoggers

```
public void setLoggers (List<LogMapping> loggers)
```

setRoot

```
public void setRoot (LogMapping root)
```

setTrash

```
public void setTrash (List<LogMapping> trash)
```

11.59 org.motechproject.osgi.web.util

11.59.1 BundleHeaders

```
public class BundleHeaders
```

Fields**BLUEPRINT_ENABLED**

```
public static final String BLUEPRINT_ENABLED
```

CONTEXT_PATH

```
public static final String CONTEXT_PATH
```

RESOURCE_PATH

```
public static final String RESOURCE_PATH
```

Constructors

BundleHeaders

public **BundleHeaders** (*BundleContext bundleContext*)

BundleHeaders

public **BundleHeaders** (*Bundle bundle*)

Methods

get

public *Object* **get** (*Object key*)

getContextPath

public *String* **getContextPath** ()

getName

public *String* **getName** ()

getResourcePath

public *String* **getResourcePath** ()

getStringValue

public *String* **getStringValue** (*String key*)

getSymbolicName

public *String* **getSymbolicName** ()

getVersion

public *String* **getVersion** ()

isBlueprintEnabled

public boolean **isBlueprintEnabled** ()

11.59.2 WebBundleUtil

public final class **WebBundleUtil**

Utility class that's purpose is easing bundle related operations/searches.

Methods

findBundleByName

public static **Bundle** **findBundleByName** (**BundleContext** *bundleContext*, **String** *name*)

findBundleBySymbolicName

public static **Bundle** **findBundleBySymbolicName** (**BundleContext** *bundleContext*, **String** *symbolic-Name*)

getContextLocation

public static **String** **getContextLocation** (**Bundle** *bundle*)

getContextPath

public static **String** **getContextPath** (**Bundle** *bundle*)

getModuleId

public static **String** **getModuleId** (**Bundle** *bundle*)

11.60 org.motechproject.scheduler.builder

11.60.1 CronJobExpressionBuilder

public class **CronJobExpressionBuilder**

Constructors

CronJobExpressionBuilder

public **CronJobExpressionBuilder** (**Time** *startTime*, **Integer** *repeatWindowInHours*, **Integer** *repeatIntervalInMinutes*)

Methods

build

public **String** **build** ()

11.60.2 CronJobSimpleExpressionBuilder

```
public class CronJobSimpleExpressionBuilder
```

Constructors

CronJobSimpleExpressionBuilder

```
public CronJobSimpleExpressionBuilder (Time startTime)
```

Methods

build

```
public String build ()
```

withRepeatIntervalInDays

```
public CronJobSimpleExpressionBuilder withRepeatIntervalInDays (int repeatIntervalInDays)
```

11.60.3 WeeklyCronJobExpressionBuilder

```
public class WeeklyCronJobExpressionBuilder
```

Constructors

WeeklyCronJobExpressionBuilder

```
public WeeklyCronJobExpressionBuilder (DayOfWeek dayOfWeek)
```

WeeklyCronJobExpressionBuilder

```
public WeeklyCronJobExpressionBuilder (int dayOfWeekNumber)
```

Methods

build

```
public String build ()
```

withTime

```
public WeeklyCronJobExpressionBuilder withTime (Time time)
```

11.61 org.motechproject.scheduler.contract

11.61.1 CronJobId

public class **CronJobId** extends [JobId](#)

Constructors

CronJobId

public **CronJobId** ([String](#) *subject*, [String](#) *id*)

CronJobId

public **CronJobId** ([MotechEvent](#) *event*)

11.61.2 CronSchedulableJob

public class **CronSchedulableJob** implements [Serializable](#)

Schedulable Job - a data carrier class for a scheduled job that can be fired unlimited number of times as specified with the cron expression

Author Igor (iopushnyev@2paths.com) Date: 16/02/11 Time: 1:43 PM

Constructors

CronSchedulableJob

public **CronSchedulableJob** ([MotechEvent](#) *motechEvent*, [String](#) *cronExpression*, [Date](#) *startTime*, [Date](#) *endTime*)

CronSchedulableJob

public **CronSchedulableJob** ([MotechEvent](#) *motechEvent*, [String](#) *cronExpression*)

CronSchedulableJob

public **CronSchedulableJob** ([MotechEvent](#) *motechEvent*, [String](#) *cronExpression*, [Date](#) *startTime*, [Date](#) *endTime*, boolean *ignorePastFiresAtStart*)

Methods

equals

public boolean **equals** ([Object](#) *obj*)

getCronExpression

```
public String getCronExpression ()
```

getEndTime

```
public Date getEndTime ()
```

getMotechEvent

```
public MotechEvent getMotechEvent ()
```

getStartTime

```
public Date getStartTime ()
```

hashCode

```
public int hashCode ()
```

isIgnorePastFiresAtStart

```
public boolean isIgnorePastFiresAtStart ()
```

toString

```
public String toString ()
```

11.61.3 DayOfWeekSchedulableJob

public final class **DayOfWeekSchedulableJob** implements `Serializable`
Job that is scheduled on particular days of week

Constructors**DayOfWeekSchedulableJob**

```
public DayOfWeekSchedulableJob (MotechEvent motechEvent, LocalDate start, LocalDate end,  
                                List<DayOfWeek> days, Time time, boolean ignorePastFiresAt-  
                                Start)
```

DayOfWeekSchedulableJob

```
public DayOfWeekSchedulableJob (MotechEvent motechEvent, LocalDate start, LocalDate end,  
                                List<DayOfWeek> days, Time time)
```


Methods

getCronDays

```
public List<Integer> getCronDays ()
```

getEndDate

```
public LocalDate getEndDate ()
```

getMotechEvent

```
public MotechEvent getMotechEvent ()
```

getStartDate

```
public LocalDate getStartDate ()
```

getTime

```
public Time getTime ()
```

isIgnorePastFiresAtStart

```
public boolean isIgnorePastFiresAtStart ()
```

11.61.4 EventInfo

```
public class EventInfo
```

EventInfo is the class which contains information about event associated with scheduled job.

Constructors

EventInfo

```
public EventInfo ()
```

Methods

getParameters

```
public Map<String, Object> getParameters ()
```

getSubject

```
public String getSubject ()
```

setParameters

```
public void setParameters (Map<String, Object> parameters)
```

setSubject

```
public void setSubject (String subject)
```

11.61.5 JobBasicInfo

```
public class JobBasicInfo
```

JobBasicInfo is the class which contains information about scheduled job and its current state.

Fields

ACTIVITY_ACTIVE

```
public static final String ACTIVITY_ACTIVE
```

ACTIVITY_FINISHED

```
public static final String ACTIVITY_FINISHED
```

ACTIVITY_NOTSTARTED

```
public static final String ACTIVITY_NOTSTARTED
```

JOBTYPE_CRON

```
public static final String JOBTYPE_CRON
```

JOBTYPE_REPEATING

```
public static final String JOBTYPE_REPEATING
```

JOBTYPE_RUNONCE

```
public static final String JOBTYPE_RUNONCE
```

STATUS_BLOCKED

```
public static final String STATUS_BLOCKED
```

STATUS_ERROR

```
public static final String STATUS_ERROR
```

STATUS_OK

```
public static final String STATUS_OK
```

STATUS_PAUSED

```
public static final String STATUS_PAUSED
```

Constructors**JobBasicInfo**

```
public JobBasicInfo ()
```

JobBasicInfo

```
public JobBasicInfo (String activity, String status, String name, String startDate, String nextFireDate,  
                    String endDate, String jobType, String info)
```

Methods**getActivity**

```
public String getActivity ()
```

getEndDate

```
public String getEndDate ()
```

getInfo

```
public String getInfo ()
```

getJobType

```
public String getJobType ()
```

getName

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

getNextFireDate

public **String** **getNextFireDate** ()

getStartDate

public **String** **getStartDate** ()

getStatus

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

setActivity

public void **setActivity** (**String** *activity*)

setEndDate

public void **setEndDate** (**String** *endDate*)

setInfo

public void **setInfo** (**String** *info*)

setJobType

public void **setJobType** (**String** *jobType*)

setName

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

setNextFireDate

public void **setNextFireDate** (**String** *nextFireDate*)

setStartDate

public void **setStartDate** (**String** *startDate*)

setStatus

public void **setStatus** (*String status*)

11.61.6 JobDetailedInfo

public class **JobDetailedInfo**

JobDetailedInfo is the class which wraps the EventInfo list.

See also: [EventInfo](#)

Constructors

JobDetailedInfo

public **JobDetailedInfo** ()

JobDetailedInfo

public **JobDetailedInfo** (*List<EventInfo> eventInfoList*)

Methods

getEventInfoList

public *List<EventInfo>* **getEventInfoList** ()

setEventInfoList

public void **setEventInfoList** (*List<EventInfo> eventInfoList*)

11.61.7 JobId

public abstract class **JobId** implements [Serializable](#)

Constructors

JobId

public **JobId** (*String subject, String id, String suffix*)

JobId

public **JobId** (*MotechEvent motechEvent, String suffix*)

Methods

toString

```
public String toString ()
```

value

```
public String value ()
```

11.61.8 RepeatingJobId

```
public class RepeatingJobId extends JobId
```

Fields

SUFFIX_REPEATJOBID

```
public static final String SUFFIX_REPEATJOBID
```

Constructors

RepeatingJobId

```
public RepeatingJobId (String subject, String id)
```

RepeatingJobId

```
public RepeatingJobId (MotechEvent repeatingEvent)
```

11.61.9 RepeatingSchedulableJob

```
public class RepeatingSchedulableJob implements Serializable
```

Schedulable Job - a data carrier class for a scheduled job that can be fired set number of times

Constructors

RepeatingSchedulableJob

```
public RepeatingSchedulableJob ()
```

RepeatingSchedulableJob

```
public RepeatingSchedulableJob (MotechEvent motechEvent, Date startTime, Date endTime, Integer repeatCount, Long repeatIntervalInMilliseconds, boolean ignorePastFiresAtStart)
```

RepeatingSchedulableJob

public **RepeatingSchedulableJob** (*MotechEvent* motechEvent, *Date* startTime, *Date* endTime, *Long* repeatIntervalInMilliseconds, boolean ignorePastFiresAtStart)

Methods

equals

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

getEndTime

public *Date* **getEndTime** ()

getMotechEvent

public *MotechEvent* **getMotechEvent** ()

getRepeatCount

public *Integer* **getRepeatCount** ()

getRepeatIntervalInMilliseconds

public *Long* **getRepeatIntervalInMilliseconds** ()

getStartTime

public *Date* **getStartTime** ()

hashCode

public int **hashCode** ()

isIgnorePastFiresAtStart

public boolean **isIgnorePastFiresAtStart** ()

isUseOriginalFireTimeAfterMisfire

public boolean **isUseOriginalFireTimeAfterMisfire** ()

setEndTime

```
public RepeatingSchedulableJob setEndTime (Date endTime)
```

setIgnorePastFiresAtStart

```
public RepeatingSchedulableJob setIgnorePastFiresAtStart (boolean ignorePastFiresAtStart)
```

Ignore past fires when start time of job is in past.

```
ex : repeating job with interval of 5 unit, and current time in between fire 2 and 3 will start
  1      2      3      4
  +-----+-----+-----+
  start      ^current time
```

Parameters

- **ignorePastFiresAtStart** –

setMotechEvent

```
public RepeatingSchedulableJob setMotechEvent (MotechEvent motechEvent)
```

setRepeatCount

```
public RepeatingSchedulableJob setRepeatCount (Integer repeatCount)
```

setRepeatIntervalInMilliseconds

```
public RepeatingSchedulableJob setRepeatIntervalInMilliseconds (Long repeatIntervalInMil-
                                                                    liSeconds)
```

setStartTime

```
public RepeatingSchedulableJob setStartTime (Date startTime)
```

setUseOriginalFireTimeAfterMisfire

```
public RepeatingSchedulableJob setUseOriginalFireTimeAfterMisfire (boolean useOriginalFire-
                                                                    TimeAfterMisfire)
```

toString

```
public String toString ()
```

11.61.10 RunOnceJobId

```
public class RunOnceJobId extends JobId
```


Fields

SUFFIX_RUNONCEJOBID

public static final [String](#) SUFFIX_RUNONCEJOBID

Constructors

RunOnceJobId

public **RunOnceJobId** ([String](#) subject, [String](#) id)

RunOnceJobId

public **RunOnceJobId** ([MotechEvent](#) runOnceEvent)

11.61.11 RunOnceSchedulableJob

public final class **RunOnceSchedulableJob** implements [Serializable](#)

Run Once Schedulable Job - a data carrier class for a job scheduled in the future that can be fired only once

This class is immutable

User: Igor (iopushnyev@2paths.com) Date: 16/02/11 Time: 1:43 PM

Constructors

RunOnceSchedulableJob

public **RunOnceSchedulableJob** ([MotechEvent](#) motechEvent, [Date](#) startDate)

Constructor

Parameters

- **motechEvent** –
 - event data message that will be send by Motech Scheduler when this job is fired
- **startDate** –
 - date and time when the job fill be fired

Throws

- **IllegalArgumentException** – if motechEvent or startDate is null or startDate is in past

Methods

equals

public boolean **equals** ([Object](#) o)

getMotechEvent

```
public MotechEvent getMotechEvent ()
```

getStartDate

```
public Date getStartDate ()
```

hashCode

```
public int hashCode ()
```

toString

```
public String toString ()
```

11.62 org.motechproject.scheduler.exception

11.62.1 MotechSchedulerException

```
public class MotechSchedulerException extends RuntimeException
```

User: Igor (iopushnyev@2paths.com) Date: 17/02/11 Time: 4:20 PM

Constructors**MotechSchedulerException**

```
public MotechSchedulerException ()
```

MotechSchedulerException

```
public MotechSchedulerException (String message)
```

MotechSchedulerException

```
public MotechSchedulerException (String message, Throwable cause)
```

MotechSchedulerException

```
public MotechSchedulerException (Throwable cause)
```

11.62.2 SchedulerInstantiationException

```
public class SchedulerInstantiationException extends RuntimeException
```

Constructors

`SchedulerInstantiationException`

```
public SchedulerInstantiationException (Throwable e)
```

11.62.3 `SchedulerShutdownException`

```
public class SchedulerShutdownException extends RuntimeException
```

The `SchedulerShutdownException` exception informs about that there were problems with shutdown scheduler.

Constructors

`SchedulerShutdownException`

```
public SchedulerShutdownException (Throwable e)
```

11.63 `org.motechproject.scheduler.factory`

11.63.1 `MotechSchedulerFactoryBean`

```
public class MotechSchedulerFactoryBean
```

The `MotechSchedulerFactoryBean` is used to created scheduler and start it.

Constructors

`MotechSchedulerFactoryBean`

```
public MotechSchedulerFactoryBean (ApplicationContext applicationContext, Properties scheduler-Properties)
```

Methods

`getQuartzScheduler`

```
public Scheduler getQuartzScheduler ()
```

`getQuartzSchedulerFactoryBean`

```
public SchedulerFactoryBean getQuartzSchedulerFactoryBean ()
```

`init`

```
public void init ()
```

shutdown

public void **shutdown** ()

11.64 org.motechproject.scheduler.service

11.64.1 MotechSchedulerActionProxyService

public interface **MotechSchedulerActionProxyService**

Methods

scheduleCronJob

void **scheduleCronJob** (String *subject*, Map<Object, Object> *parameters*, String *cronExpression*, DateTime *startTime*, DateTime *endTime*, Boolean *ignorePastFiresAtStart*)

scheduleDayOfWeekJob

void **scheduleDayOfWeekJob** (String *subject*, Map<Object, Object> *parameters*, DateTime *start*, DateTime *end*, List<Object> *days*, DateTime *time*, Boolean *ignorePastFiresAtStart*)

scheduleRepeatingJob

void **scheduleRepeatingJob** (String *subject*, Map<Object, Object> *parameters*, DateTime *startTime*, DateTime *endTime*, Integer *repeatCount*, Long *repeatIntervalInMilliseconds*, Boolean *ignorePastFiresAtStart*, Boolean *useOriginalFireTimeAfterMisfire*)

scheduleRunOnceJob

void **scheduleRunOnceJob** (String *subject*, Map<Object, Object> *parameters*, DateTime *startDate*)

unscheduleJobs

void **unscheduleJobs** (String *subject*)

11.64.2 MotechSchedulerService

public interface **MotechSchedulerService**

ingroup scheduler Motech Scheduler Service Interface provides methods to schedule reschedule and unschedule a job Set a global policy that determines trigger fire behaviour for misfired triggers. For details see quartz documentations for misfire policy do_nothing -> @see CronTrigger.MISFIRE_INSTRUCTION_DO_NOTHING

```
fire_once_now    -> @see CronTrigger.MISFIRE_INSTRUCTION_FIRE_ONCE_NOW ignore -
> @see CronTrigger.MISFIRE_INSTRUCTION_IGNORE_MISFIRE_POLICY fire_now -
> @see SimpleTrigger.MISFIRE_INSTRUCTION_FIRE_NOW ignore -> @see SimpleTrigger.MISFIRE_INSTRUCTION_IGNORE_MISFIRE_POLICY
reschedule_next_with_existing_count -> @see SimpleTrigger.MISFIRE_INSTRUCTION_RESCHEDULE_NEXT_WITH_EXISTING_COUNT
reschedule_next_with_remaining_count -> @see SimpleTrigger.MISFIRE_INSTRUCTION_RESCHEDULE_NEXT_WITH_REMAINING_COUNT
reschedule_now_with_existing_count -> @see SimpleTrigger.MISFIRE_INSTRUCTION_RESCHEDULE_NOW_WITH_EXISTING_COUNT
reschedule_now_with_remaining_count -> @see SimpleTrigger.MISFIRE_INSTRUCTION_RESCHEDULE_NOW_WITH_REMAINING_COUNT
```

Author Igor (iopushnyev@2paths.com) Date: 16/02/11

Fields

JOB_ID_KEY

String **JOB_ID_KEY**

Methods

getScheduledJobDetailedInfo

JobDetailedInfo **getScheduledJobDetailedInfo** (JobBasicInfo jobBasicInfo)

getScheduledJobTimings

List<Date> **getScheduledJobTimings** (String subject, String externalJobId, Date startDate, Date endDate)

getScheduledJobTimingsWithPrefix

List<Date> **getScheduledJobTimingsWithPrefix** (String subject, String externalJobIdPrefix, Date startDate, Date endDate)

getScheduledJobsBasicInfo

List<JobBasicInfo> **getScheduledJobsBasicInfo** ()

rescheduleJob

void **rescheduleJob** (String subject, String externalId, String cronExpression)

Reschedules a job with the given job ID to be fired according to the given Cron Expression Previous version of the configured Motech Scheduled Even that will be created when the job is fired remains us it was

Parameters

- **subject** –
- **externalId** –
- **cronExpression** –

safeScheduleJob

void **safeScheduleJob** ([CronSchedulableJob](#) *cronSchedulableJob*)

Same as `scheduleJob`, except that it would update existing job if one exists instead of creating a new one

Parameters

- `cronSchedulableJob` –

safeScheduleRepeatingJob

void **safeScheduleRepeatingJob** ([RepeatingSchedulableJob](#) *repeatingSchedulableJob*)

Same as `safeScheduleRepeatingJob` with `intervening = true`

Parameters

- `repeatingSchedulableJob` –

safeScheduleRunOnceJob

void **safeScheduleRunOnceJob** ([RunOnceSchedulableJob](#) *schedulableJob*)

Same as `scheduleRunOnceJob`, except that it would update existing job if one exists instead of creating a new one

Parameters

- `schedulableJob` –

safeUnscheduleAllJobs

void **safeUnscheduleAllJobs** ([String](#) *jobIdPrefix*)

safeUnscheduleJob

void **safeUnscheduleJob** ([String](#) *subject*, [String](#) *externalId*)

Same as `unscheduleJob` except that it would not throw an exception if the job doesn't exist

Parameters

- `subject` –
- `externalId` –

safeUnscheduleRepeatingJob

void **safeUnscheduleRepeatingJob** ([String](#) *subject*, [String](#) *externalId*)

Same as `unscheduleRepeatingJob` except that it would not throw an exception if the job doesn't exist

Parameters

- `subject` –
- `externalId` –

safeUnscheduleRunOnceJob

void **safeUnscheduleRunOnceJob** (*String subject, String externalId*)

Same as unscheduleRunOnceJob except that it would not throw an exception if the job doesn't exist

Parameters

- **subject** –
- **externalId** –

scheduleDayOfWeekJob

void **scheduleDayOfWeekJob** (*DayOfWeekSchedulableJob dayOfWeekSchedulableJob*)

Same as safeScheduleDayOfWeekJob with intervening = true

Parameters

- **dayOfWeekSchedulableJob** –

scheduleJob

void **scheduleJob** (*CronSchedulableJob cronSchedulableJob*)

Schedules the given schedulable job. The Job ID by which the job will be referencing in the future should be provided in an Instance of MotechEvent in SchedulableJob (see MotechEvent.jobId) If a job with the same job ID as the given exists, this job will be unscheduled and the given schedulable job will be scheduled

Parameters

- **cronSchedulableJob** –

scheduleRepeatingJob

void **scheduleRepeatingJob** (*RepeatingSchedulableJob repeatingSchedulableJob*)

Schedules the given schedulable job. The Job ID by which the job will be referencing in the future should be provided in an Instance of MotechEvent in SchedulableJob (see MotechEvent.jobId) If a job with the same job ID as the given exists, this job will be unscheduled and the given schedulable job will be scheduled

Parameters

- **repeatingSchedulableJob** –

scheduleRunOnceJob

void **scheduleRunOnceJob** (*RunOnceSchedulableJob schedulableJob*)

unscheduleAllJobs

void **unscheduleAllJobs** (*String jobIdPrefix*)

unscheduleJob

void **unscheduleJob** (*String subject*, *String externalId*)

Unschedule a job with the given job ID

Parameters

- **subject** – : String representing domain operation eg. “pill-reminder”, “outbox-call” or `motechEvent.getSubject()`
- **externalId** – : domain specific id as String.

unscheduleJob

void **unscheduleJob** (*JobId job*)

unscheduleRepeatingJob

void **unscheduleRepeatingJob** (*String subject*, *String externalId*)

unscheduleRunOnceJob

void **unscheduleRunOnceJob** (*String subject*, *String externalId*)

Unschedule a run once job with the given job ID

Parameters

- **subject** – : String representing domain operation eg. “pill-reminder”, “outbox-call” or `motechEvent.getSubject()`
- **externalId** – : domain specific id as String.

updateScheduledJob

void **updateScheduledJob** (*MotechEvent motechEvent*)

Updates MotechEvent data of the job defined by `jobId` in the given instance of that class

Parameters

- **motechEvent** –

11.65 org.motechproject.scheduler.service.impl

11.65.1 MotechScheduledJob

public class **MotechScheduledJob** implements `Job`

Methods

execute

public void **execute** (*JobExecutionContext jobExecutionContext*)

11.65.2 MotechScheduler

public final class **MotechScheduler**
ingroup scheduler

Main class that can bootstrap a Motech Scheduler

Author Igor (iopushnyev@2paths.com)

Methods

main

public static void **main** (*String[] args*)

11.65.3 MotechSchedulerActionProxyServiceImpl

public class **MotechSchedulerActionProxyServiceImpl** implements [MotechSchedulerActionProxyService](#)

Constructors

MotechSchedulerActionProxyServiceImpl

public **MotechSchedulerActionProxyServiceImpl** (*MotechSchedulerService schedulerService*)

Methods

scheduleCronJob

public void **scheduleCronJob** (*String subject*, *Map<Object, Object> parameters*, *String cronExpression*,
DateTime startTime, *DateTime endTime*, *Boolean ignorePastFiresAtStart*)

scheduleDayOfWeekJob

public void **scheduleDayOfWeekJob** (*String subject*, *Map<Object, Object> parameters*, *DateTime start*,
DateTime end, *List<Object> days*, *DateTime time*, *Boolean ignorePastFiresAtStart*)

scheduleRepeatingJob

```
public void scheduleRepeatingJob (String subject, Map<Object, Object> parameters, DateTime startTime, DateTime endTime, Integer repeatCount, Long repeatIntervalInMilliseconds, Boolean ignorePastFiresAtStart, Boolean useOriginalFireTimeAfterMisfire)
```

scheduleRunOnceJob

```
public void scheduleRunOnceJob (String subject, Map<Object, Object> parameters, DateTime startDate)
```

unscheduleJobs

```
public void unscheduleJobs (String subject)
```

11.65.4 MotechSchedulerServiceImpl

public class **MotechSchedulerServiceImpl** implements [MotechSchedulerService](#)
Motech Scheduler Service implementation

See also: [MotechSchedulerService](#)

Fields

JOB_GROUP_NAME

```
public static final String JOB_GROUP_NAME
```

Constructors

MotechSchedulerServiceImpl

```
public MotechSchedulerServiceImpl (MotechSchedulerFactoryBean motechSchedulerFactoryBean,  
SettingsFacade schedulerSettings)
```

Methods

assertArgumentNotNull

```
protected void assertArgumentNotNull (String objectName, Object object)
```

getScheduledJobDetailedInfo

```
public JobDetailedInfo getScheduledJobDetailedInfo (JobBasicInfo jobBasicInfo)
```

getScheduledJobTimings

```
public List<Date> getScheduledJobTimings (String subject, String externalJobId, Date startDate, Date endDate)
```

getScheduledJobTimingsWithPrefix

```
public List<Date> getScheduledJobTimingsWithPrefix (String subject, String externalJobIdPrefix, Date startDate, Date endDate)
```

getScheduledJobsBasicInfo

```
public List<JobBasicInfo> getScheduledJobsBasicInfo ()
```

logObjectIfNotNull

```
protected void logObjectIfNotNull (Object obj)
```

rescheduleJob

```
public void rescheduleJob (String subject, String externalId, String cronExpression)
```

safeScheduleJob

```
public void safeScheduleJob (CronSchedulableJob cronSchedulableJob)
```

safeScheduleRepeatingJob

```
public void safeScheduleRepeatingJob (RepeatingSchedulableJob repeatingSchedulableJob)
```

safeScheduleRunOnceJob

```
public void safeScheduleRunOnceJob (RunOnceSchedulableJob schedulableJob)
```

safeUnscheduleAllJobs

```
public void safeUnscheduleAllJobs (String jobIdPrefix)
```

safeUnscheduleJob

```
public void safeUnscheduleJob (String subject, String externalId)
```

safeUnscheduleRepeatingJob

```
public void safeUnscheduleRepeatingJob (String subject, String externalId)
```

safeUnscheduleRunOnceJob

public void **safeUnscheduleRunOnceJob** (*String subject*, *String externalId*)

scheduleDayOfWeekJob

public void **scheduleDayOfWeekJob** (*DayOfWeekSchedulableJob dayOfWeekSchedulableJob*)

scheduleJob

public void **scheduleJob** (*CronSchedulableJob cronSchedulableJob*)

scheduleRepeatingJob

public void **scheduleRepeatingJob** (*RepeatingSchedulableJob repeatingSchedulableJob*)

scheduleRunOnceJob

public void **scheduleRunOnceJob** (*RunOnceSchedulableJob schedulableJob*)

unscheduleAllJobs

public void **unscheduleAllJobs** (*String jobIdPrefix*)

unscheduleJob

public void **unscheduleJob** (*String subject*, *String externalId*)

unscheduleJob

public void **unscheduleJob** (*JobId job*)

unscheduleRepeatingJob

public void **unscheduleRepeatingJob** (*String subject*, *String externalId*)

unscheduleRunOnceJob

public void **unscheduleRunOnceJob** (*String subject*, *String externalId*)

updateScheduledJob

public void **updateScheduledJob** (*MotechEvent motechEvent*)

11.66 org.motechproject.security.annotations

11.66.1 SecurityAnnotationBeanPostProcessor

public class **SecurityAnnotationBeanPostProcessor** implements [BeanPostProcessor](#)

A [BeanPostProcessor](#) used by Motech to load permissions from modules. Given a module context, it looks for [PreAuthorize](#) and [PostAuthorize](#) annotations. These annotations are then parsed using an [ExpressionParser](#). The permission names are deduced from [hasRole](#) and [hasAnyRole](#) in the annotation value. The names of permissions are then saved using the [MotechPermissionService](#). The bundle name used to construct the permission is retrieved from the application context.

Constructors

SecurityAnnotationBeanPostProcessor

public **SecurityAnnotationBeanPostProcessor** ([MotechPermissionService](#) *permissionService*)

Methods

postProcessAfterInitialization

public [Object](#) **postProcessAfterInitialization** ([Object](#) *bean*, [String](#) *beanName*)

postProcessBeforeInitialization

public [Object](#) **postProcessBeforeInitialization** ([Object](#) *bean*, [String](#) *beanName*)

processAnnotations

public void **processAnnotations** ([ApplicationContext](#) *applicationContext*)

11.67 org.motechproject.security.authentication

11.67.1 MotechAccessVoter

public class **MotechAccessVoter** implements [AccessDecisionVoter](#)<[Object](#)>

A custom [AccessDecisionVoter](#) for voting on whether a specific user has access to a particular URL. For example, a security rule can specify that the users `motech` and `admin` have access to a particular URL. This loads the metadata source with attributes for `ACCESS_motech` and `ACCESS_admin`. When a user invokes that URL, an affirmative based voting system will check whether or not the user is `motech` or `admin`. If not, they are denied permission, otherwise they are granted access.

Methods

supports

public boolean **supports** (*ConfigAttribute attribute*)

supports

public boolean **supports** (*Class<?> clazz*)

vote

public int **vote** (*Authentication authentication, Object object, Collection<ConfigAttribute> attributes*)

11.67.2 MotechBasicAuthenticationEntryPoint

public class **MotechBasicAuthenticationEntryPoint** extends *BasicAuthenticationEntryPoint*
An entry point for BASIC authentications, sets the correct server realm key.

Fields

SECURITY_REALM_KEY

public static final *String* **SECURITY_REALM_KEY**

Constructors

MotechBasicAuthenticationEntryPoint

public **MotechBasicAuthenticationEntryPoint** (*SettingsFacade settingsFacade*)

11.67.3 MotechLoginUrlAuthenticationEntryPoint

public class **MotechLoginUrlAuthenticationEntryPoint** extends *LoginUrlAuthenticationEntryPoint*

Methods

commence

public void **commence** (*HttpServletRequest request, HttpServletResponse response, AuthenticationException authException*)

11.67.4 MotechLogoutSuccessHandler

public class **MotechLogoutSuccessHandler** implements [LogoutHandler](#)

A logout handler for removing Motech user sessions from Motech's internally kept session handler. This is invoked when a user logs out.

Methods

logout

public void **logout** ([HttpServletRequest](#) *request*, [HttpServletResponse](#) *response*, [Authentication](#) *authentication*)

11.67.5 MotechPasswordEncoder

public class **MotechPasswordEncoder** extends [BCryptPasswordEncoder](#)

Methods

encodePassword

public [String](#) **encodePassword** ([String](#) *rawPassword*)

isPasswordValid

public boolean **isPasswordValid** ([String](#) *encPassword*, [String](#) *rawPassword*)

11.67.6 MotechRestBasicAuthenticationEntryPoint

public class **MotechRestBasicAuthenticationEntryPoint** extends [BasicAuthenticationEntryPoint](#)

A custom entry point that is invoked when there is an authentication exception within the filter. This ensures that when a user does not have login privileges and are unable to authenticate, a 401 unauthorized response is returned.

Fields

SECURITY_REALM_KEY

public static final [String](#) **SECURITY_REALM_KEY**

Constructors

MotechRestBasicAuthenticationEntryPoint

public **MotechRestBasicAuthenticationEntryPoint** ([SettingsFacade](#) *settingsFacade*)

Methods

commence

```
public void commence (HttpServletRequest request, HttpServletResponse response, AuthenticationException
    authException)
```

11.67.7 MotechSuccessHandler

```
public class MotechSuccessHandler extends SavedRequestAwareAuthenticationSuccessHandler
```

Methods

onAuthenticationSuccess

```
public void onAuthenticationSuccess (HttpServletRequest request, HttpServletResponse response,
    Authentication authentication)
```

setSessionHandler

```
public void setSessionHandler (SessionHandler sessionHandler)
```

11.68 org.motechproject.security.builder

11.68.1 SecurityRuleBuilder

```
public class SecurityRuleBuilder
```

The security rule builder is responsible for building a `SecurityFilterChain`, which consists of a matcher pattern and a list of Spring security filters. The filters are created and configured base upon the security rule's settings.

Fields

NO_METHODS_REQUIRED_EXCEPTION_MESSAGE

```
public static final String NO_METHODS_REQUIRED_EXCEPTION_MESSAGE
```

NO_PATTERN_EXCEPTION_MESSAGE

```
public static final String NO_PATTERN_EXCEPTION_MESSAGE
```

NO_PROTOCOL_EXCEPTION_MESSAGE

```
public static final String NO_PROTOCOL_EXCEPTION_MESSAGE
```


NO_SUPPORTED_SCHEMES_EXCEPTION_MESSAGE

```
public static final String NO_SUPPORTED_SCHEMES_EXCEPTION_MESSAGE
```

Methods

buildSecurityChain

```
public synchronized SecurityFilterChain buildSecurityChain (MotechURLSecurityRule securityRule,  
                                                           HTTPMethod method)
```

setAuthenticationManager

```
public void setAuthenticationManager (AuthenticationManager authenticationManager)
```

setBasicAuthenticationEntryPoint

```
public void setBasicAuthenticationEntryPoint (AuthenticationEntryPoint basicAuthenticationEn-  
                                              tryPoint)
```

setChannelDecisionManager

```
public void setChannelDecisionManager (ChannelDecisionManager channelDecisionManager)
```

setLoginAuthenticationEntryPoint

```
public void setLoginAuthenticationEntryPoint (AuthenticationEntryPoint loginAuthenticationEn-  
                                              tryPoint)
```

setMotechLogoutHandler

```
public void setMotechLogoutHandler (MotechLogoutSuccessHandler motechLogoutHandler)
```

setOpenIDAuthenticationFilter

```
public void setOpenIDAuthenticationFilter (OpenIDAuthenticationFilter openIDAuthentication-  
                                           Filter)
```

setSettingsFacade

```
public void setSettingsFacade (SettingsFacade settingsFacade)
```

`setUsernamePasswordAuthenticationFilter`

```
public void setUsernamePasswordAuthenticationFilter (UsernamePasswordAuthenticationFilter  
                                                    usernamePasswordAuthenticationFilter)
```

11.69 org.motechproject.security.constants

11.69.1 HTTPMethod

```
public enum HTTPMethod  
    Enumeration of HTTP request method.
```

Enum Constants

ANY

```
public static final HTTPMethod ANY
```

DELETE

```
public static final HTTPMethod DELETE
```

GET

```
public static final HTTPMethod GET
```

HEAD

```
public static final HTTPMethod HEAD
```

OPTIONS

```
public static final HTTPMethod OPTIONS
```

POST

```
public static final HTTPMethod POST
```

PUT

```
public static final HTTPMethod PUT
```

TRACE

public static final [HTTPMethod](#) **TRACE**

11.69.2 PermissionNames

public final class **PermissionNames**

Fields

ACTIVATE_USER_PERMISSION

public static final [String](#) **ACTIVATE_USER_PERMISSION**

ADD_USER_PERMISSION

public static final [String](#) **ADD_USER_PERMISSION**

DELETE_USER_PERMISSION

public static final [String](#) **DELETE_USER_PERMISSION**

EDIT_USER_PERMISSION

public static final [String](#) **EDIT_USER_PERMISSION**

MANAGE_PERMISSION_PERMISSION

public static final [String](#) **MANAGE_PERMISSION_PERMISSION**

MANAGE_ROLE_PERMISSION

public static final [String](#) **MANAGE_ROLE_PERMISSION**

MANAGE_USER_PERMISSION

public static final [String](#) **MANAGE_USER_PERMISSION**

VIEW_BASIC_EMAIL_LOGS_PERMISSION

public static final [String](#) **VIEW_BASIC_EMAIL_LOGS_PERMISSION**

VIEW_DETAILED_EMAIL_LOGS_PERMISSION

public static final [String](#) **VIEW_DETAILED_EMAIL_LOGS_PERMISSION**

VIEW_SECURITY

public static final [String](#) **VIEW_SECURITY**

11.69.3 Protocol

public enum **Protocol**
Enumeration of Protocols supported by the module

Enum Constants

HTTP

public static final [Protocol](#) **HTTP**

HTTPS

public static final [Protocol](#) **HTTPS**

11.69.4 Scheme

public enum **Scheme**
Enumeration of scheme supported by the module

Enum Constants

BASIC

public static final [Scheme](#) **BASIC**

NO_SECURITY

public static final [Scheme](#) **NO_SECURITY**

OATH

public static final [Scheme](#) **OATH**
not supported yet

OPEN_ID

public static final [Scheme](#) **OPEN_ID**

USERNAME_PASSWORD

public static final [Scheme](#) **USERNAME_PASSWORD**

11.69.5 SecurityConfigConstants

public final class **SecurityConfigConstants**

A class for holding constants related to the options available for dynamic security rules. MotechURLSecurityRule is where these options are used. Prefixes related to security voting are also stored in this class.

Fields

ANY_PATTERN

public static final [String](#) **ANY_PATTERN**

ROLE_ACCESS_PREFIX

public static final [String](#) **ROLE_ACCESS_PREFIX**

USER_ACCESS_PREFIX

public static final [String](#) **USER_ACCESS_PREFIX**

11.69.6 UserRoleNames

public final class **UserRoleNames**

Fields

BUNDLE_ADMIN_ROLE

public static final [String](#) **BUNDLE_ADMIN_ROLE**

EMAIL_ADMIN_ROLE

public static final [String](#) **EMAIL_ADMIN_ROLE**

MDS_ADMIN

public static final [String](#) **MDS_ADMIN**

ROLES_ADMIN

public static final [String](#) **ROLES_ADMIN**

SECURITY_ADMIN_ROLE

```
public static final String SECURITY_ADMIN_ROLE
```

USER_ADMIN_ROLE

```
public static final String USER_ADMIN_ROLE
```

11.70 org.motechproject.security.domain

11.70.1 MotechPermission

```
public class MotechPermission
```

Constructors

MotechPermission

```
public MotechPermission ()
```

MotechPermission

```
public MotechPermission (String permissionName, String bundleName)
```

Methods

getBundleName

```
public String getBundleName ()
```

getPermissionName

```
public String getPermissionName ()
```

setBundleName

```
public void setBundleName (String bundleName)
```

setPermissionName

```
public void setPermissionName (String permissionName)
```

toString

```
public String toString ()
```

11.70.2 MotechRole

```
public class MotechRole
```

Constructors

MotechRole

```
public MotechRole ()
```

MotechRole

```
public MotechRole (String roleName, List<String> permissionNames, boolean deletable)
```

Methods

getPermissionNames

```
public List<String> getPermissionNames ()
```

getRoleName

```
public String getRoleName ()
```

hasPermission

```
public boolean hasPermission (String permissionName)
```

isDeletable

```
public boolean isDeletable ()
```

removePermission

```
public void removePermission (String permissionName)
```

setDeletable

```
public void setDeletable (boolean deletable)
```

setPermissionNames

```
public void setPermissionNames (List<String> permissionNames)
```

setRoleName

```
public void setRoleName (String roleName)
```

11.70.3 MotechSecurityConfiguration

```
public class MotechSecurityConfiguration
```

The MotechSecurityConfiguration is a single document that contains all of the URL security rule configuration. The configuration was designed as one document because the entire filter chain must be reconstructed each time it is updated, therefore managing many references is unnecessary.

Constructors

MotechSecurityConfiguration

```
public MotechSecurityConfiguration ()
```

MotechSecurityConfiguration

```
public MotechSecurityConfiguration (List<MotechURLSecurityRule> securityRules)
```

Methods

getSecurityRules

```
public List<MotechURLSecurityRule> getSecurityRules ()
```

setSecurityRules

```
public void setSecurityRules (List<MotechURLSecurityRule> securityRules)
```

11.70.4 MotechURLSecurityRule

```
public class MotechURLSecurityRule
```

The MotechURLSecurityRule specifies the configuration for setting up a Spring SecurityFilterChain.

Details regarding configuration:

- **pattern** - URL pattern the security rule applies to
- **supportedSchemes** - Security rules that should apply to the URL, such as BASIC or OPEN_ID
- **protocol** - Protocol the security rule applies to, such as HTTP or HTTPS
- **permissionAccess** - Requires user has at least one of the listed permission to access the URL

- userAccess** - User specific access for a URL, such as motech or admin, when combined with permission access they act as an either or (one must be true)
- priority** - For future use in determining the ordering of filter chains, may be deprecated depending on UI implementation
- rest** - Whether the endpoint is meant for a form login process or as an REST end-point that does not create a session for the
- origin** - The module or user the rule originated from
- version** - The version of the module or platform the rule was created
- methodsRequired** - HTTP methods the rule applies to, if ANY is used then any method is matched, if a set is used, such as GET, POST, etc, then each will have its own corresponding filter chain with the same security found in that rule

Methods

getId

```
public Long getId ()
```

getMethodsRequired

```
public List<HTTPMethod> getMethodsRequired ()
```

getOrigin

```
public String getOrigin ()
```

getPattern

```
public String getPattern ()
```

getPermissionAccess

```
public List<String> getPermissionAccess ()
```

getPriority

```
public int getPriority ()
```

getProtocol

```
public Protocol getProtocol ()
```

getSupportedSchemes

public [List](#)<[Scheme](#)> **getSupportedSchemes** ()

getUserAccess

public [List](#)<[String](#)> **getUserAccess** ()

getVersion

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

isActive

public boolean **isActive** ()

isDeleted

public boolean **isDeleted** ()

isRest

public boolean **isRest** ()

setActive

public void **setActive** (boolean *active*)

setDeleted

public void **setDeleted** (boolean *deleted*)

setId

public void **setId** ([Long](#) *id*)

setMethodsRequired

public void **setMethodsRequired** ([List](#)<[HTTPMethod](#)> *methodsRequired*)

setOrigin

public void **setOrigin** ([String](#) *origin*)

setPattern

public void **setPattern** (*String pattern*)

setPermissionAccess

public void **setPermissionAccess** (*List<String> permissionAccess*)

setPriority

public void **setPriority** (*int priority*)

setProtocol

public void **setProtocol** (*Protocol protocol*)

setRest

public void **setRest** (*boolean rest*)

setSupportedSchemes

public void **setSupportedSchemes** (*List<Scheme> supportedSchemes*)

setUserAccess

public void **setUserAccess** (*List<String> userAccess*)

setVersion

public void **setVersion** (*String version*)

11.70.5 MotechUser

public class **MotechUser**

Constructors**MotechUser**

public **MotechUser** ()

MotechUser

```
public MotechUser (String userName, String password, String email, String externalId, List<String> roles,  
                  String openId, Locale locale)
```

Methods

equals

```
public boolean equals (Object o)
```

getEmail

```
public String getEmail ()
```

getExternalId

```
public String getExternalId ()
```

getLocale

```
public Locale getLocale ()
```

getOpenId

```
public String getOpenId ()
```

getPassword

```
public String getPassword ()
```

getRoles

```
public List<String> getRoles ()
```

getUserName

```
public String getUserName ()
```

hasRole

```
public boolean hasRole (String role)
```

hashCode

```
public int hashCode ()
```

isActive

```
public boolean isActive ()
```

setActive

```
public void setActive (boolean active)
```

setEmail

```
public void setEmail (String email)
```

setExternalId

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

setLocale

```
public void setLocale (Locale locale)
```

setOpenId

```
public void setOpenId (String openId)
```

setPassword

```
public void setPassword (String password)
```

setRoles

```
public void setRoles (List<String> roles)
```

setUserName

```
public void setUserName (String userName)
```

toString

```
public String toString ()
```

11.70.6 MotechUserProfile

public class **MotechUserProfile** implements [Serializable](#)

Constructors

MotechUserProfile

public **MotechUserProfile** ([MotechUser](#) *user*)

Methods

getExternalId

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

getLocale

public [Locale](#) **getLocale** ()

getRoles

public [List](#)<[String](#)> **getRoles** ()

getUserName

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

hasRole

public boolean **hasRole** ([String](#) *role*)

isActive

public boolean **isActive** ()

11.70.7 PasswordRecovery

public class **PasswordRecovery**

Methods

getEmail

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

getExpirationDate

```
public DateTime getExpirationDate ()
```

getLocale

```
public Locale getLocale ()
```

getToken

```
public String getToken ()
```

getUsername

```
public String getUsername ()
```

setEmail

```
public void setEmail (String email)
```

setExpirationDate

```
public void setExpirationDate (DateTime expirationDate)
```

setLocale

```
public void setLocale (Locale locale)
```

setToken

```
public void setToken (String token)
```

setUsername

```
public void setUsername (String username)
```

11.71 org.motechproject.security.ex

11.71.1 EmailExistsException

```
public class EmailExistsException extends RuntimeException
```

Constructors

EmailExistsException

public **EmailExistsException** (*String message*)

11.71.2 InvalidTokenException

public class **InvalidTokenException** extends [Exception](#)

Constructors

InvalidTokenException

public **InvalidTokenException** ()

InvalidTokenException

public **InvalidTokenException** (*String message*)

11.71.3 NonAdminUserException

public class **NonAdminUserException** extends [Exception](#)

Constructors

NonAdminUserException

public **NonAdminUserException** (*String message*)

11.71.4 RoleHasUserException

public class **RoleHasUserException** extends [RuntimeException](#)

Represents a failed attempt to delete a role currently assigned to a user.

Constructors

RoleHasUserException

public **RoleHasUserException** (*String message*)

11.71.5 SecurityConfigException

public class **SecurityConfigException** extends [RuntimeException](#)

A runtime exception thrown when the security config does not pass validation constraints required in order to construct a new security chain. Ideally should not be thrown as the UI should not allow invalid data to be submitted.

Constructors

SecurityConfigException

public **SecurityConfigException** ([String message](#))

11.71.6 UserNotFoundException

public class **UserNotFoundException** extends [Exception](#)

Constructors

UserNotFoundException

public **UserNotFoundException** ()

UserNotFoundException

public **UserNotFoundException** ([String message](#))

11.72 org.motechproject.security.model

11.72.1 PermissionDto

public class **PermissionDto** implements [Serializable](#)

The `PermissionDto` contains information about permission.

Constructors

PermissionDto

public **PermissionDto** ()

PermissionDto

public **PermissionDto** ([MotechPermission motechPermission](#))

PermissionDto

```
public PermissionDto (String permissionName, String bundleName)
```

Methods

equals

```
public boolean equals (Object obj)
```

getBundleName

```
public String getBundleName ()
```

getPermissionName

```
public String getPermissionName ()
```

hashCode

```
public int hashCode ()
```

setBundleName

```
public void setBundleName (String bundleName)
```

setPermissionName

```
public void setPermissionName (String permissionName)
```

toString

```
public String toString ()
```

11.72.2 RoleDto

```
public class RoleDto
```

Transfer Motech role data between representations.

Role data transfer object facilitates exchange of role data among services, repository, and client user interface.

Constructors

RoleDto

```
public RoleDto ()
```

RoleDto

```
public RoleDto (MotechRole motechRole)
```

RoleDto

```
public RoleDto (String roleName, List<String> permissionNames)
```

RoleDto

```
public RoleDto (String roleName, List<String> permissionNames, boolean deletable)
```

Methods**equals**

```
public boolean equals (Object obj)
```

getOriginalRoleName

```
public String getOriginalRoleName ()
```

getPermissionNames

```
public List<String> getPermissionNames ()
```

getRoleName

```
public String getRoleName ()
```

hashCode

```
public int hashCode ()
```

isDeletable

```
public boolean isDeletable ()
```

setDeletable

```
public void setDeletable (boolean deletable)
```

setOriginalRoleName

```
public void setOriginalRoleName (String originalRoleName)
```

setPermissionNames

```
public void setPermissionNames (List<String> permissionNames)
```

setRoleName

```
public void setRoleName (String roleName)
```

toString

```
public String toString ()
```

11.72.3 SecurityConfigDto

```
public class SecurityConfigDto
```

Used to transfer security configuration to and from a web request and UI

Methods**getSecurityRules**

```
public List<SecurityRuleDto> getSecurityRules ()
```

setSecurityRules

```
public void setSecurityRules (List<SecurityRuleDto> securityRules)
```

11.72.4 SecurityRuleDto

```
public class SecurityRuleDto
```

Transfer Motech security rule data between representations.

Methods**getId**

```
public Long getId ()
```

getMethodsRequired

```
public List<String> getMethodsRequired ()
```

getOrigin

```
public String getOrigin ()
```

getPattern

```
public String getPattern ()
```

getPermissionAccess

```
public List<String> getPermissionAccess ()
```

getPriority

```
public int getPriority ()
```

getProtocol

```
public String getProtocol ()
```

getSupportedSchemes

```
public List<String> getSupportedSchemes ()
```

getUserAccess

```
public List<String> getUserAccess ()
```

getVersion

```
public String getVersion ()
```

isActive

```
public boolean isActive ()
```

isDeleted

```
public boolean isDeleted ()
```

isRest

```
public boolean isRest ()
```

setActive

```
public void setActive (boolean active)
```

setDeleted

public void **setDeleted** (boolean *deleted*)

setId

public void **setId** (Long *id*)

setMethodsRequired

public void **setMethodsRequired** (List<String> *methodsRequired*)

setOrigin

public void **setOrigin** (String *origin*)

setPattern

public void **setPattern** (String *pattern*)

setPermissionAccess

public void **setPermissionAccess** (List<String> *permissionAccess*)

setPriority

public void **setPriority** (int *priority*)

setProtocol

public void **setProtocol** (String *protocol*)

setRest

public void **setRest** (boolean *rest*)

setSupportedSchemes

public void **setSupportedSchemes** (List<String> *supportedSchemes*)

setUserAccess

public void **setUserAccess** (List<String> *userAccess*)

setVersion

```
public void setVersion (String version)
```

11.72.5 UserDto

```
public class UserDto
```

Constructors

UserDto

```
public UserDto ()
```

UserDto

```
public UserDto (MotechUser motechUser)
```

Methods

getEmail

```
public String getEmail ()
```

getExternalId

```
public String getExternalId ()
```

getLocale

```
public Locale getLocale ()
```

getOpenId

```
public String getOpenId ()
```

getPassword

```
public String getPassword ()
```

getRoles

```
public List<String> getRoles ()
```

getUserName

public **String** **getUserName** ()

isActive

public boolean **isActive** ()

isGeneratePassword

public boolean **isGeneratePassword** ()

setActive

public void **setActive** (boolean *active*)

setEmail

public void **setEmail** (**String** *email*)

setExternalId

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

setGeneratePassword

public void **setGeneratePassword** (boolean *generatePassword*)

setLocale

public void **setLocale** (**Locale** *locale*)

setOpenId

public void **setOpenId** (**String** *openId*)

setPassword

public void **setPassword** (**String** *password*)

setRoles

public void **setRoles** (**List**<**String**> *roles*)

setUserName

```
public void setUserName (String userName)
```

11.73 org.motechproject.security.repository

11.73.1 AllMotechPermissions

```
public class AllMotechPermissions
```

Methods**add**

```
public void add (MotechPermission permission)
```

delete

```
public void delete (MotechPermission permission)
```

findByPermissionName

```
public MotechPermission findByPermissionName (String permissionName)
```

getPermissions

```
public List<MotechPermission> getPermissions ()
```

setDataService

```
public void setDataService (MotechPermissionsDataService dataService)
```

11.73.2 AllMotechRoles

```
public class AllMotechRoles
```

Methods**add**

```
public void add (MotechRole role)
```

findByRoleName

```
public MotechRole findByRoleName (String roleName)
```

getRoles

```
public List<MotechRole> getRoles ()
```

remove

```
public void remove (MotechRole motechRole)
```

setDataService

```
public void setDataService (MotechRolesDataService dataService)
```

update

```
public void update (MotechRole motechRole)
```

11.73.3 AllMotechSecurityRules

```
public class AllMotechSecurityRules
```

Implementation of DAO interface that utilizes a MDS back-end for storage. Only one MotechSecurityConfiguration file should be saved at a time, so adding the document looks for the old document in order to update it if it already exists. Rather than updating the object reference, the old configuration's ID and revision are used for the new document.

Methods**addOrUpdate**

```
public void addOrUpdate (MotechSecurityConfiguration config)
```

getMotechSecurityConfiguration

```
public MotechSecurityConfiguration getMotechSecurityConfiguration ()
```

getRuleById

```
public MotechURLSecurityRule getRuleById (Long id)
```

getRules

```
public List<MotechURLSecurityRule> getRules ()
```

getRulesByOrigin

```
public List<MotechURLSecurityRule> getRulesByOrigin (String origin)
```

remove

```
public void remove (MotechSecurityConfiguration config)
```

setDataService

```
public void setDataService (MotechURLSecurityRuleDataService dataService)
```

11.73.4 AllMotechUsers

```
public class AllMotechUsers
```

Methods**add**

```
public void add (MotechUser user)
```

addOpenIdUser

```
public void addOpenIdUser (MotechUser user)
```

findByRole

```
public List<MotechUser> findByRole (String role)
```

findByUsername

```
public MotechUser findByUsername (String userName)
```

findUserByEmail

```
public MotechUser findUserByEmail (String email)
```

findUserByOpenId

```
public MotechUser findUserByOpenId (String openId)
```

getOpenIdUsers

```
public List<MotechUser> getOpenIdUsers ()
```

getUsers

```
public List<MotechUser> getUsers ()
```

remove

```
public void remove (MotechUser motechUser)
```

setDataService

```
public void setDataService (MotechUsersDataService dataService)
```

update

```
public void update (MotechUser motechUser)
```

11.73.5 AllPasswordRecoveries

```
public class AllPasswordRecoveries
```

Methods

add

```
public void add (PasswordRecovery passwordRecovery)
```

allRecoveries

```
public List<PasswordRecovery> allRecoveries ()
```

createRecovery

```
public PasswordRecovery createRecovery (String username, String email, String token, DateTime expirationDate, Locale locale)
```

findForToken

```
public PasswordRecovery findForToken (String token)
```

findForUser

```
public PasswordRecovery findForUser (String username)
```

getExpired

```
public List<PasswordRecovery> getExpired ()
```

remove

```
public void remove (PasswordRecovery passwordRecovery)
```

setDataService

```
public void setDataService (PasswordRecoveriesDataService dataService)
```

update

```
public void update (PasswordRecovery passwordRecovery)
```

11.73.6 MotechPermissionsDataService

```
public interface MotechPermissionsDataService extends MotechDataService<MotechPermission>
```

Methods**findByPermissionName**

```
MotechPermission findByPermissionName (String permissionName)
```

11.73.7 MotechRolesDataService

```
public interface MotechRolesDataService extends MotechDataService<MotechRole>
```

Methods**findByRoleName**

```
MotechRole findByRoleName (String roleName)
```

11.73.8 MotechURLSecurityRuleDataService

```
public interface MotechURLSecurityRuleDataService extends MotechDataService<MotechURLSecurityRule>
```

Methods**findByOrigin**

```
List<MotechURLSecurityRule> findByOrigin (String origin)
```

11.73.9 MotechUsersDataService

public interface **MotechUsersDataService** extends **MotechDataService**<**MotechUser**>

Methods

findByEmail

MotechUser **findByEmail** (*String email*)

findByOpenId

MotechUser **findByOpenId** (*String openId*)

findByRole

List<**MotechUser**> **findByRole** (*String role*)

findByUserName

MotechUser **findByUserName** (*String userName*)

11.73.10 PasswordRecoveriesDataService

public interface **PasswordRecoveriesDataService** extends **MotechDataService**<**PasswordRecovery**>

Methods

findByExpirationDate

List<**PasswordRecovery**> **findByExpirationDate** (*Range*<**DateTime**> *range*)

findForToken

PasswordRecovery **findForToken** (*String token*)

findForUser

PasswordRecovery **findForUser** (*String username*)

11.74 org.motechproject.security.service

11.74.1 AuthoritiesService

public interface **AuthoritiesService**

Service interface to retrieve authorities(permissions) for a given MotechUser

Methods

authoritiesFor

List<GrantedAuthority> **authoritiesFor** (MotechUser user)

11.74.2 MotechPermissionService

public interface **MotechPermissionService**

Service for managing Motech permissions.

Methods

addPermission

void **addPermission** (PermissionDto permission)

deletePermission

void **deletePermission** (String permissionName)

getPermissions

List<PermissionDto> **getPermissions** ()

11.74.3 MotechProxyManager

public class **MotechProxyManager**

The MotechProxyManager acts as a wrapper around Spring's FilterChainProxy. The FilterChainProxy contains a list of immutable SecurityFilterChain objects which Spring's security consults for filters when handling requests. In order to dynamically define new secure, a new FilterChainProxy is constructed and the reference is updated. The MotechProxyManager acts as a customized delegate in MotechDelegatingFilterProxy.

Methods

getDefaultSecurityConfiguration

public MotechSecurityConfiguration **getDefaultSecurityConfiguration** ()

This method reads default security configuration from the file containing security rules and returns it.

Returns MotechSecurityConfiguration default security rules

getFilterChainProxy

public [FilterChainProxy](#) **getFilterChainProxy** ()

initializeProxyChain

public void **initializeProxyChain** ()

This method serves the same purpose of `rebuildProxyChain`, but does not require any kind of security authentication so it should only ever be used by the activator, which does not have an authentication object.

rebuildProxyChain

public synchronized void **rebuildProxyChain** ()

Method to invoke to dynamically re-define the Spring security. All rules converted into security filter chains in order to create a new `FilterChainProxy`. The order of the rules in the list matters for filtering purposes.

setProxy

public void **setProxy** ([FilterChainProxy](#) *proxy*)

setSecurityRuleBuilder

public void **setSecurityRuleBuilder** ([SecurityRuleBuilder](#) *securityRuleBuilder*)

setSecurityRulesDAO

public void **setSecurityRulesDAO** ([AllMotechSecurityRules](#) *securityRulesDAO*)

11.74.4 MotechRoleService

public interface **MotechRoleService**

Methods

createRole

void **createRole** ([RoleDto](#) *role*)

deleteRole

void **deleteRole** ([RoleDto](#) *role*)

getRole

`RoleDto` **getRole** (`String` *roleName*)

getRoles

`List<RoleDto>` **getRoles** ()

updateRole

`void` **updateRole** (`RoleDto` *role*)

11.74.5 MotechURLSecurityService

public interface **MotechURLSecurityService**

Service to access and update security configuration details from the platform. Permission based, method level security is defined to prevent unauthorized users from updating security.

Methods**findAllSecurityRules**

`List<SecurityRuleDto>` **findAllSecurityRules** ()

A protected method for viewing security rule information for the platform.

Returns All URL security rules found in the database

updateSecurityConfiguration

`void` **updateSecurityConfiguration** (`SecurityConfigDto` *configuration*)

A protected method for updating security configuration for the platform.

Parameters

- **configuration** – The updated security information, which will cause an updating of the motech proxy manager

11.74.6 MotechUserService

public interface **MotechUserService**

Service interface that defines APIs to retrieve and manage user details

Methods**activateUser**

`void` **activateUser** (`String` *username*)

changePassword

`MotechUserProfile` **changePassword** (`String` *username*, `String` *oldPassword*, `String` *newPassword*)

deleteUser

`void` **deleteUser** (`UserDto` *user*)

getCurrentUser

`UserDto` **getCurrentUser** ()

getLocale

`Locale` **getLocale** (`String` *userName*)

getOpenIdUsers

`List<MotechUserProfile>` **getOpenIdUsers** ()

getRoles

`List<String>` **getRoles** (`String` *userName*)

getUser

`UserDto` **getUser** (`String` *userName*)

getUserByEmail

`UserDto` **getUserByEmail** (`String` *email*)

getUsers

`List<MotechUserProfile>` **getUsers** ()

hasActiveAdminUser

`boolean` **hasActiveAdminUser** ()

hasUser

`boolean` **hasUser** (`String` *username*)

register

void **register** (*String username, String password, String email, String externalId, List<String> roles, Locale locale*)

register

void **register** (*String username, String password, String email, String externalId, List<String> roles, Locale locale, boolean isActive, String openId*)

retrieveUserByCredentials

MotechUserProfile **retrieveUserByCredentials** (*String username, String password*)

sendLoginInformation

void **sendLoginInformation** (*String userName, String password*)

setLocale

void **setLocale** (*String userName, Locale locale*)

updateUserDetailsWithPassword

void **updateUserDetailsWithPassword** (*UserDto user*)

updateUserDetailsWithoutPassword

void **updateUserDetailsWithoutPassword** (*UserDto user*)

11.74.7 PasswordRecoveryService

public interface **PasswordRecoveryService**

Methods**cleanUpExpiredRecoveries**

void **cleanUpExpiredRecoveries** ()

oneTimeTokenOpenId

void **oneTimeTokenOpenId** (*String email*)

passwordRecoveryRequest

void **passwordRecoveryRequest** (*String email*)

resetPassword

void **resetPassword** (*String token*, *String password*, *String passwordConfirmation*)

validateToken

boolean **validateToken** (*String token*)

validateTokenAndLoginUser

void **validateTokenAndLoginUser** (*String token*, *HttpServletRequest request*, *HttpServletResponse response*)

11.74.8 SecurityRoleLoader

public class **SecurityRoleLoader**

Constructors

SecurityRoleLoader

public **SecurityRoleLoader** (*MotechRoleService roleService*)

Methods

loadRoles

public void **loadRoles** (*ApplicationContext applicationContext*)

11.74.9 SecurityRuleLoader

public class **SecurityRuleLoader**

Helper class that scans an application context for security rules and re-initializes the MotechProxyManager security chain.

Methods

loadRules

public synchronized void **loadRules** (*ApplicationContext applicationContext*)

Attempts to load rules from the application context, if rules are found, the security configuration is updated. Synchronized so there are not race conditions on the data.

setAllSecurityRules

```
public void setAllSecurityRules (AllMotechSecurityRules allSecurityRules)
```

setProxyManager

```
public void setProxyManager (MotechProxyManager proxyManager)
```

11.74.10 UserContextService

```
public interface UserContextService  
    Interface to refresh user context (all or specified username)
```

Methods**refreshAllUsersContextIfActive**

```
void refreshAllUsersContextIfActive ()
```

refreshUserContextIfActive

```
void refreshUserContextIfActive (String userName)
```

11.75 org.motechproject.server.api

11.75.1 BundleIcon

```
public class BundleIcon
```

Fields**ICON_LOCATIONS**

```
public static final String[] ICON_LOCATIONS
```

Constructors**BundleIcon**

```
public BundleIcon (byte[] icon, String mime)
```

Methods

getContentLength

```
public int getContentLength ()
```

getIcon

```
public byte[] getIcon ()
```

getMimeType

```
public String getMimeType ()
```

11.75.2 BundleInformation

```
public class BundleInformation
```

Class acting as a DTO for a *Bundle* in the system.

Fields

BUNDLE_NAME

```
protected static final String BUNDLE_NAME
```

Constructors

BundleInformation

```
public BundleInformation (Bundle bundle)
```

Methods

equals

```
public boolean equals (Object arg0)
```

getAngularModule

```
public String getAngularModule ()
```

getBundleId

```
public long getBundleId ()
```

getLocation

```
public String getLocation ()
```

getModuleName

```
public String getModuleName ()
```

getName

```
public String getName ()
```

getSettingsURL

```
public String getSettingsURL ()
```

getState

```
public State getState ()
```

getSymbolicName

```
public String getSymbolicName ()
```

getVersion

```
public Version getVersion ()
```

hasStatus

```
public boolean hasStatus (int status)
```

hashCode

```
public int hashCode ()
```

setAngularModule

```
public void setAngularModule (String angularModule)
```

setModuleName

```
public void setModuleName (String moduleName)
```

`setSettingsURL`

public void **setSettingsURL** (*String settingsURL*)

11.75.3 BundleInformation.State

public enum **State**

Represents the bundle state.

Enum Constants

ACTIVE

public static final *BundleInformation.State* **ACTIVE**

INSTALLED

public static final *BundleInformation.State* **INSTALLED**

RESOLVED

public static final *BundleInformation.State* **RESOLVED**

STARTING

public static final *BundleInformation.State* **STARTING**

STOPPING

public static final *BundleInformation.State* **STOPPING**

UNINSTALLED

public static final *BundleInformation.State* **UNINSTALLED**

UNKNOWN

public static final *BundleInformation.State* **UNKNOWN**

11.75.4 BundleLoader

public interface **BundleLoader**

Interface for custom bundle loading processes

Author Ricky Wang

Methods

loadBundle

void **loadBundle** (*Bundle bundle*)

Parameters

- **bundle** –

Throws

- **BundleLoadingException** –

11.75.5 BundleLoadingException

public class **BundleLoadingException** extends *Exception*

Constructors

BundleLoadingException

public **BundleLoadingException** (*String message*)

BundleLoadingException

public **BundleLoadingException** (*String message, Throwable cause*)

BundleLoadingException

public **BundleLoadingException** (*Throwable cause*)

11.75.6 JarInformation

public class **JarInformation**

Fields

BUNDLE_SYMBOLIC_NAME

public static final *String* **BUNDLE_SYMBOLIC_NAME**

BUNDLE_VERSION

public static final *String* **BUNDLE_VERSION**

EXTRACTION_FAILED

public static final `String` **EXTRACTION_FAILED**

IMPLEMENTATION_TITLE

public static final `String` **IMPLEMENTATION_TITLE**

IMPLEMENTATION_VENDOR_ID

public static final `String` **IMPLEMENTATION_VENDOR_ID**

IMPLEMENTATION_VERSION

public static final `String` **IMPLEMENTATION_VERSION**

Constructors

JarInformation

public **JarInformation** (`File` *file*)

Methods

getBundleSymbolicName

public `String` **getBundleSymbolicName** ()

getBundleVersion

public `String` **getBundleVersion** ()

getFilename

public `String` **getFilename** ()

getImplementationTitle

public `String` **getImplementationTitle** ()

getImplementationVendorID

public `String` **getImplementationVendorID** ()

getImplementationVersion

```
public String getImplementationVersion ()
```

getPath

```
public String getPath ()
```

11.75.7 JarInformationHandler

```
public class JarInformationHandler
```

Fields**JAR_FILE_EXTENSION**

```
public static final String JAR_FILE_EXTENSION
```

Constructors**JarInformationHandler**

```
public JarInformationHandler (String path)
```

Methods**extractJarInformationFromPath**

```
public void extractJarInformationFromPath ()
```

getJarList

```
public List<JarInformation> getJarList ()
```

getPath

```
public String getPath ()
```

initHandler

```
public void initHandler ()
```

11.76 org.motechproject.server.config

11.76.1 SettingsFacade

public class **SettingsFacade**

SettingsFacade provides an interface to access application configuration present in files or database

Methods

afterPropertiesSet

public void **afterPropertiesSet** ()

areConfigurationSettingsRegistered

public boolean **areConfigurationSettingsRegistered** ()

asProperties

public [Properties](#) **asProperties** ()

constructSymbolicName

protected [String](#) **constructSymbolicName** ()

findFilename

protected [String](#) **findFilename** ([String](#) *key*)

getActivemqConfig

public [Properties](#) **getActivemqConfig** ()

getModuleName

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

getPlatformSettings

public [MotechSettings](#) **getPlatformSettings** ()

getProperties

public [Properties](#) **getProperties** ([String](#) *filename*)

getProperty

```
public String getProperty (String key)
```

getProperty

```
public String getProperty (String key, String filename)
```

getRawConfig

```
public InputStream getRawConfig (String filename)
```

getResourceFileName

```
protected static String getResourceFileName (Resource resource)
```

getSymbolicName

```
public String getSymbolicName ()
```

registerAllProperties

```
protected void registerAllProperties ()
```

registerAllRawConfig

```
protected void registerAllRawConfig ()
```

registerProperties

```
protected void registerProperties (String filename, Properties properties)
```

saveConfigProperties

```
public void saveConfigProperties (String filename, Properties properties)
```

savePlatformSettings

```
public void savePlatformSettings (MotechSettings settings)
```

saveRawConfig

```
public void saveRawConfig (String filename, Resource resource)
```

setConfigFiles

public void **setConfigFiles** ([List<Resource>](#) *resources*)

setConfigurationService

public void **setConfigurationService** ([ConfigurationService](#) *configurationService*)

setModuleName

public void **setModuleName** ([String](#) *moduleName*)

setProperty

public void **setProperty** ([String](#) *key*, [String](#) *value*)

setRawConfigFiles

public void **setRawConfigFiles** ([List<Resource>](#) *resources*)

unregisterProperties

public void **unregisterProperties** ([String](#) *symbolicName*)

11.77 org.motechproject.server.config.domain

11.77.1 LoginMode

public final class **LoginMode**
Encapsulates the operations on login mode

Fields

OPEN_ID

public static final [LoginMode](#) **OPEN_ID**

REPOSITORY

public static final [LoginMode](#) **REPOSITORY**

Methods

getName

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

isOpenId

public boolean **isOpenId** ()

isRepository

public boolean **isRepository** ()

valueOf

public static [LoginMode](#) **valueOf** ([String](#) *loginMode*)

11.77.2 MotechSettings

public interface **MotechSettings**

Interface for main motech settings managment

Methods

asProperties

[Properties](#) **asProperties** ()

getActivemqProperties

[Properties](#) **getActivemqProperties** ()

getConfigFileChecksum

[String](#) **getConfigFileChecksum** ()

getFilePath

[String](#) **getFilePath** ()

getLanguage

[String](#) **getLanguage** ()

getLastRun

`DateTime` **getLastRun** ()

getLoginMode

`LoginMode` **getLoginMode** ()

getProviderName

`String` **getProviderName** ()

getProviderUrl

`String` **getProviderUrl** ()

getServerHost

`String` **getServerHost** ()

getServerUrl

`String` **getServerUrl** ()

getStatusMsgTimeout

`String` **getStatusMsgTimeout** ()

getUploadSize

`String` **getUploadSize** ()

isPlatformInitialized

`boolean` **isPlatformInitialized** ()

load

`void` **load** (`DigestInputStream` *dis*)

savePlatformSetting

`void` **savePlatformSetting** (`String` *key*, `String` *value*)

setConfigFileChecksum

void **setConfigFileChecksum** (*String configFileChecksum*)

setFilePath

void **setFilePath** (*String filePath*)

setLanguage

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

setLastRun

void **setLastRun** (*DateTime lastRun*)

setLoginModeValue

void **setLoginModeValue** (*String loginMode*)

setPlatformInitialized

void **setPlatformInitialized** (*boolean platformInitialized*)

setProviderName

void **setProviderName** (*String providerName*)

setProviderUrl

void **setProviderUrl** (*String providerUrl*)

setServerUrl

void **setServerUrl** (*String serverUrl*)

setStatusMsgTimeout

void **setStatusMsgTimeout** (*String statusMsgTimeout*)

setUploadSize

void **setUploadSize** (*String uploadSize*)

updateFromProperties

void **updateFromProperties** (*Properties props*)

updateSettings

void **updateSettings** (*String configFileChecksum, String filePath, Properties platformSettings*)

11.77.3 MotechURL

public class **MotechURL**

Fields

URL_PATTERN

public static final *String* **URL_PATTERN**

Constructors

MotechURL

public **MotechURL** (*String url*)

Methods

getHost

public *String* **getHost** ()

toString

public *String* **toString** ()

11.77.4 SettingsRecord

public class **SettingsRecord** implements *MotechSettings*
Class for storing settings values

Constructors

SettingsRecord

public **SettingsRecord** ()

Methods

asProperties

```
public Properties asProperties ()
```

getActivemqProperties

```
public Properties getActivemqProperties ()
```

getConfigFileChecksum

```
public String getConfigFileChecksum ()
```

getFilePath

```
public String getFilePath ()
```

getLanguage

```
public String getLanguage ()
```

getLastRun

```
public DateTime getLastRun ()
```

getLoginMode

```
public LoginMode getLoginMode ()
```

getLoginModeValue

```
public String getLoginModeValue ()
```

getPlatformSettings

```
public Map<String, String> getPlatformSettings ()
```

getProviderName

```
public String getProviderName ()
```

getProviderUrl

```
public String getProviderUrl ()
```

getServerHost

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

getServerUrl

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

getStatusMsgTimeout

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

getUploadSize

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

isPlatformInitialized

public boolean **isPlatformInitialized** ()

load

public synchronized void **load** ([DigestInputStream](#) *dis*)

mergeWithDefaults

public void **mergeWithDefaults** ([Properties](#) *defaultConfig*)

removeDefaults

public void **removeDefaults** ([Properties](#) *defaultConfig*)

savePlatformSetting

public void **savePlatformSetting** ([String](#) *key*, [String](#) *value*)

setConfigFileChecksum

public void **setConfigFileChecksum** ([String](#) *configFileChecksum*)

setFilePath

public void **setFilePath** ([String](#) *filePath*)

setLanguage

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

setLastRun

public void **setLastRun** (*DateTime lastRun*)

setLoginModeValue

public void **setLoginModeValue** (*String loginMode*)

setPlatformInitialized

public void **setPlatformInitialized** (*boolean platformInitialized*)

setPlatformSettings

public void **setPlatformSettings** (*Map<String, String> platformSettings*)

setProviderName

public void **setProviderName** (*String providerName*)

setProviderUrl

public void **setProviderUrl** (*String providerUrl*)

setServerUrl

public void **setServerUrl** (*String serverUrl*)

setStatusMsgTimeout

public void **setStatusMsgTimeout** (*String statusMsgTimeout*)

setUploadSize

public void **setUploadSize** (*String uploadSize*)

updateFromProperties

public void **updateFromProperties** (*Properties props*)

updateSettings

public void **updateSettings** (*String* configFileChecksum, *String* filePath, *Properties* platformSettings)

11.78 org.motechproject.server.config.service

11.78.1 ConfigLoader

public class **ConfigLoader**

Config loader used to load the platform core settings.

Methods

findExistingConfigs

public *List*<*File*> **findExistingConfigs** ()

Finds all configs from the config location

Throws

- **IOException** – If there is any error while handling the files.

loadDefaultConfig

public *SettingsRecord* **loadDefaultConfig** ()

loadMotechSettings

public *SettingsRecord* **loadMotechSettings** ()

setCoreConfigurationService

public void **setCoreConfigurationService** (*CoreConfigurationService* coreConfigurationService)

setEventAdmin

public void **setEventAdmin** (*EventAdmin* eventAdmin)

setResourceLoader

public void **setResourceLoader** (*ResourceLoader* resourceLoader)

11.78.2 SettingService

public interface **SettingService** extends *MotechDataService*<*SettingsRecord*>

11.79 org.motechproject.server.startup

11.79.1 MotechPlatformState

public enum **MotechPlatformState**

Defines the different states of the MOTECH system.

Enum Constants

DB_ERROR

public static final [MotechPlatformState](#) **DB_ERROR**

FIRST_RUN

public static final [MotechPlatformState](#) **FIRST_RUN**

NEED_BOOTSTRAP_CONFIG

public static final [MotechPlatformState](#) **NEED_BOOTSTRAP_CONFIG**

NEED_CONFIG

public static final [MotechPlatformState](#) **NEED_CONFIG**

NORMAL_RUN

public static final [MotechPlatformState](#) **NORMAL_RUN**

NO_DB

public static final [MotechPlatformState](#) **NO_DB**

STARTUP

public static final [MotechPlatformState](#) **STARTUP**

11.79.2 StartupManager

public class **StartupManager**

StartupManager controlling and managing the application loading

Methods

canLaunchBundles

public boolean **canLaunchBundles** ()

getDefaultSettings

public [SettingsRecord](#) **getDefaultSettings** ()

This function is only called when the default configuration is loaded and is no config in the database or external files

isBootstrapConfigRequired

public boolean **isBootstrapConfigRequired** ()

isConfigRequired

public boolean **isConfigRequired** ()

startup

public void **startup** ()

11.80 org.motechproject.server.ui

11.80.1 LocaleService

public interface **LocaleService**

A service responsible for localization. Allows retrieval/settings of user language as well as retrieving localized messages for a user request. Can also be used to retrieve a list of usable languages.

Methods

getMessages

[Map](#)<[String](#), [String](#)> **getMessages** ([HttpServletRequest](#) *request*)

getSupportedLanguages

[NavigableMap](#)<[String](#), [String](#)> **getSupportedLanguages** ()

getUserLocale

[Locale](#) **getUserLocale** ([HttpServletRequest](#) *request*)

setUserLocale

void **setUserLocale** ([HttpServletRequest](#) *request*, [HttpServletResponse](#) *response*, [Locale](#) *locale*)

11.81 org.motechproject.server.ui.ex

11.81.1 AlreadyRegisteredException

public class **AlreadyRegisteredException** extends [RuntimeException](#)

Constructors

AlreadyRegisteredException

public **AlreadyRegisteredException** ([String](#) *msg*)

11.82 org.motechproject.server.web.controller

11.82.1 Constants

public final class **Constants**
Class that has all the common UI constants.

Fields

REDIRECT_BOOTSTRAP

public static final [String](#) **REDIRECT_BOOTSTRAP**

REDIRECT_HOME

public static final [String](#) **REDIRECT_HOME**

REDIRECT_STARTUP

public static final [String](#) **REDIRECT_STARTUP**

11.82.2 DashboardController

public class **DashboardController**
Main application controller. Responsible for retrieving information shared across the UI of different modules.
The view returned by this controller will embed the UI of the currently requested module.

Methods

accessdenied

public ModelAndView **accessdenied** (HttpServletRequest *request*)

getTime

public String **getTime** (HttpServletRequest *request*)

getUptime

public DateTime **getUptime** ()

getUser

public UserInfo **getUser** (HttpServletRequest *request*)

index

public ModelAndView **index** (HttpServletRequest *request*)

setBundleContext

public void **setBundleContext** (BundleContext *bundleContext*)

setLocaleService

public void **setLocaleService** (LocaleService *localeService*)

setStartupManager

public void **setStartupManager** (StartupManager *startupManager*)

11.82.3 ForgotController

public class **ForgotController**
Forgot Controller for reset password.

Methods

forgotPost

public String **forgotPost** (String *email*)

getForgotViewData

```
public ForgotViewData getForgotViewData (HttpServletRequest request)
```

login

```
public ModelAndView login (HttpServletRequest request)
```

11.82.4 LocaleController

```
public class LocaleController
```

The `LocaleController` class is responsible for handling requests connected with internationalization

Methods**getAvailableLocales**

```
public Map<String, String> getAvailableLocales (HttpServletRequest request)
```

getLangLocalisation

```
public Map<String, String> getLangLocalisation (HttpServletRequest request)
```

getSupportedLanguages

```
public NavigableMap<String, String> getSupportedLanguages ()
```

getUserLang

```
public String getUserLang (HttpServletRequest request)
```

setUserLang

```
public void setUserLang (HttpServletRequest request, HttpServletResponse response, String language,  
                        String country, String variant)
```

11.82.5 LoginController

```
public class LoginController
```

Login Controller for user authentication.

Methods**getLoginViewData**

```
public LoginViewData getLoginViewData (HttpServletRequest request)
```

login

public ModelAndView **login** (HttpServletRequest *request*)

11.82.6 ModuleController

public class **ModuleController**

Methods

getConfig

public List<ModuleConfig> **getConfig** ()

getCriticalMessage

public String **getCriticalMessage** (String *moduleName*)

getMenu

public ModuleMenu **getMenu** (HttpServletRequest *request*)

getUser

public UserInfo **getUser** (HttpServletRequest *request*)

setBundleContext

public void **setBundleContext** (BundleContext *bundleContext*)

setLocaleService

public void **setLocaleService** (LocaleService *localeService*)

setMenuBuilder

public void **setMenuBuilder** (MenuBuilder *menuBuilder*)

setUiFrameworkService

public void **setUiFrameworkService** (UiFrameworkService *uiFrameworkService*)

11.82.7 ResetController

public class **ResetController**

Methods

getResetViewData

```
public ResetViewData getResetViewData (HttpServletRequest request)
```

reset

```
public ResetViewData reset (ResetForm form, HttpServletRequest request)
```

resetView

```
public ModelAndView resetView (HttpServletRequest request)
```

11.82.8 StartupController

```
public class StartupController
```

```
    StartupController that manages the platform system start up and captures the platform core settings and user information.
```

Methods

getStartupViewData

```
public StartupViewData getStartupViewData (HttpServletRequest request)
```

setStartupFormValidatorFactory

```
public void setStartupFormValidatorFactory (StartupFormValidatorFactory validatorFactory)
```

startup

```
public ModelAndView startup ()
```

submitForm

```
public List<String> submitForm (StartupForm startupSettings)
```

11.82.9 StatusController

```
public class StatusController
```

Methods

status

```
public String status ()
```

11.83 org.motechproject.server.web.form

11.83.1 LoginForm

```
public class LoginForm
```

Methods

getPassword

```
public String getPassword ()
```

getUserName

```
public String getUserName ()
```

setPassword

```
public void setPassword (String password)
```

setUserName

```
public void setUsername (String userName)
```

11.83.2 ResetForm

```
public class ResetForm
```

Fields

PASSWORD

```
public static final String PASSWORD
```

PASSWORD_CONFIRMATION

```
public static final String PASSWORD_CONFIRMATION
```

Methods

getPassword

```
public String getPassword ()
```

getPasswordConfirmation

```
public String getPasswordConfirmation ()
```

getToken

```
public String getToken ()
```

setPassword

```
public void setPassword (String password)
```

setPasswordConfirmation

```
public void setPasswordConfirmation (String passwordConfirmation)
```

setToken

```
public void setToken (String token)
```

11.83.3 StartupForm

```
public class StartupForm
```

Fields

ADMIN_CONFIRM_PASSWORD

```
public static final String ADMIN_CONFIRM_PASSWORD
```

ADMIN_LOGIN

```
public static final String ADMIN_LOGIN
```

ADMIN_PASSWORD

```
public static final String ADMIN_PASSWORD
```

LANGUAGE

public static final [String](#) **LANGUAGE**

LOGIN_MODE

public static final [String](#) **LOGIN_MODE**

PROVIDER_NAME

public static final [String](#) **PROVIDER_NAME**

PROVIDER_URL

public static final [String](#) **PROVIDER_URL**

QUEUE_URL

public static final [String](#) **QUEUE_URL**

Methods

getAdminConfirmPassword

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

getAdminEmail

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

getAdminLogin

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

getAdminPassword

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

getLanguage

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

getLoginMode

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

getProviderName

```
public String getProviderName ()
```

getProviderUrl

```
public String getProviderUrl ()
```

getQueueUrl

```
public String getQueueUrl ()
```

getSchedulerUrl

```
public String getSchedulerUrl ()
```

setAdminConfirmPassword

```
public void setAdminConfirmPassword (String adminConfirmPassword)
```

setAdminEmail

```
public void setAdminEmail (String adminEmail)
```

setAdminLogin

```
public void setAdminLogin (String adminLogin)
```

setAdminPassword

```
public void setAdminPassword (String adminPassword)
```

setLanguage

```
public void setLanguage (String language)
```

setLoginMode

```
public void setLoginMode (String loginMode)
```

setProviderName

```
public void setProviderName (String providerName)
```

setProviderUrl

public void **setProviderUrl** (*String providerUrl*)

setQueueUrl

public void **setQueueUrl** (*String queueUrl*)

setSchedulerUrl

public void **setSchedulerUrl** (*String schedulerUrl*)

11.83.4 StartupSuggestionsForm

public class **StartupSuggestionsForm**

Constructors

StartupSuggestionsForm

public **StartupSuggestionsForm** ()

Methods

addDatabaseSuggestion

public void **addDatabaseSuggestion** (*String suggestion*)

addQueueSuggestion

public void **addQueueSuggestion** (*String suggestion*)

addSchedulerSuggestion

public void **addSchedulerSuggestion** (*String suggestion*)

getDatabaseUrls

public *List<String>* **getDatabaseUrls** ()

getQueueUrls

public *List<String>* **getQueueUrls** ()

`getSchedulerUrls`

```
public List<String> getSchedulerUrls ()
```

11.83.5 UserInfo

```
public class UserInfo
```

Constructors

`UserInfo`

```
public UserInfo (String userName, boolean securityLaunch, String lang)
```

Methods

`equals`

```
public boolean equals (Object obj)
```

`getLang`

```
public String getLang ()
```

`getUserName`

```
public String getUserName ()
```

`hashCode`

```
public int hashCode ()
```

`isSecurityLaunch`

```
public boolean isSecurityLaunch ()
```

`toString`

```
public String toString ()
```

11.84 org.motechproject.server.web.helper

11.84.1 Header

public final class **Header**

Methods

generateHeader

public static *String* **generateHeader** (*Bundle bundle*)

11.84.2 Header.ElementOrder

public static class **ElementOrder**

Methods

getAfter

public *String* **getAfter** ()

getBefore

public *String* **getBefore** ()

getOrder

public *String* **getOrder** ()

getPath

public *String* **getPath** ()

setAfter

public void **setAfter** (*String after*)

setBefore

public void **setBefore** (*String before*)

setOrder

public void **setOrder** (*String order*)

setPath

```
public void setPath (String path)
```

11.84.3 Header.HeaderOrder

```
public static class HeaderOrder
```

Methods**getCss**

```
public List<ElementOrder> getCss ()
```

getJs

```
public List<ElementOrder> getJs ()
```

getLib

```
public List<ElementOrder> getLib ()
```

setCss

```
public void setCss (List<ElementOrder> css)
```

setJs

```
public void setJs (List<ElementOrder> js)
```

setLib

```
public void setLib (List<ElementOrder> lib)
```

11.84.4 MenuBuilder

```
public class MenuBuilder
```

Helper class for building the modules menu view(left-hand side nav). Modules to display are retrieved from the `UIFrameworkService`. Filters entries based on user permissions.

See also: `UIFrameworkService`

Methods

buildMenu

public ModuleMenu **buildMenu** (*String username*)

Builds the menu for the given user. Modules are retrieved from UIFrameworkService and filtered based on permissions.

Parameters

- **username** – username of the user for which the menu should be built.

Returns the built menu object.

11.84.5 SuggestionHelper

public class **SuggestionHelper**

Helper class for creating UI suggestions for the user. Checks default locations for available services.

Fields

DEFAULT_ACTIVEMQ_URL

public static final *String* **DEFAULT_ACTIVEMQ_URL**

Methods

suggestActivemqUrl

public *String* **suggestActivemqUrl** ()

Suggests the ActiveMQ url.

Returns The suggested url, or an empty string if the instance is not found.

11.85 org.motechproject.server.web.validator

11.85.1 AbstractValidator

public interface **AbstractValidator**

Basic interface which startup settings validators implement

Methods

validate

void **validate** (*StartupForm target*, *List<String> errors*, *ConfigSource configSource*)

11.85.2 OpenIdUserValidator

public class **OpenIdUserValidator** implements [AbstractValidator](#)
Validates presence of OpenId related field values Also validates provider URL

Constructors

OpenIdUserValidator

public **OpenIdUserValidator** ([UrlValidator](#) *urlValidator*)

Methods

validate

public void **validate** ([StartupForm](#) *target*, [List<String>](#) *errors*, [ConfigSource](#) *configSource*)

11.85.3 PersistedUserValidator

public class **PersistedUserValidator** implements [AbstractValidator](#)
Validates presence of admin user registration fields. Checks existence of user with identical name Checks existence of user with identical email Checks that password and confirmed password field are same.

Constructors

PersistedUserValidator

public **PersistedUserValidator** ([MotechUserService](#) *userService*)

Methods

validate

public void **validate** ([StartupForm](#) *target*, [List<String>](#) *errors*, [ConfigSource](#) *configSource*)

11.85.4 QueueURLValidator

public class **QueueURLValidator** implements [AbstractValidator](#)
Validates presence of Queue URL and if present whether it is in expected format or not

Constructors

QueueURLValidator

public **QueueURLValidator** ()

QueueURLValidator

```
public QueueURLValidator (UrlValidator urlValidator)
```

Methods

validate

```
public void validate (StartupForm target, List<String> errors, ConfigSource configSource)
```

11.85.5 RequiredFieldValidator

```
public class RequiredFieldValidator implements AbstractValidator  
    Generic validator class that validates presence of a given field
```

Fields

ERROR_REQUIRED

```
public static final String ERROR_REQUIRED
```

Constructors

RequiredFieldValidator

```
public RequiredFieldValidator (String fieldName, String fieldValue)
```

Methods

equals

```
public boolean equals (Object o)
```

hashCode

```
public int hashCode ()
```

validate

```
public void validate (StartupForm target, List<String> errors, ConfigSource configSource)
```

11.85.6 ResetFormValidator

```
public class ResetFormValidator
```


Methods

validate

```
public List<String> validate (ResetForm target)
```

11.85.7 StartupFormValidator

```
public class StartupFormValidator  
    StartupFormValidator validates user information during registration process
```

Constructors

StartupFormValidator

```
public StartupFormValidator ()
```

Methods

add

```
public void add (AbstractValidator validator)
```

getValidators

```
public List<AbstractValidator> getValidators ()
```

validate

```
public List<String> validate (StartupForm target, ConfigSource configSource)
```

11.85.8 StartupFormValidatorFactory

```
public class StartupFormValidatorFactory  
    Factory to create startUpFormValidator with requisite validators. If Admin User exists,the admin user is not  
    created so the relevant validators are not added.
```

Methods

getStartupFormValidator

```
public StartupFormValidator getStartupFormValidator (StartupForm startupSettings, MotechUserService userService)
```

11.85.9 UserRegistrationValidator

public class **UserRegistrationValidator** implements [AbstractValidator](#)

Validator to validate user registration details. Delegates to either [@OpenIdUserValidator](#) or [@UserRegistrationValidator](#) depending on login mode preference.

Constructors

UserRegistrationValidator

public **UserRegistrationValidator** ([PersistedUserValidator](#) *persistedUserValidator*, [OpenIdUserValidator](#) *openIdUserValidator*)

Methods

validate

public void **validate** ([StartupForm](#) *target*, [List<String>](#) *errors*, [ConfigSource](#) *configSource*)

11.85.10 ValidationUtils

public final class **ValidationUtils**

Validation utils that consists of common validations that can be used across multiple controllers.

Methods

validateEmptyOrWhitespace

public static void **validateEmptyOrWhitespace** ([Errors](#) *errors*, [String](#) *errorMessageFormat*, [String...](#) *fields*)

11.86 org.motechproject.tasks.annotations

11.86.1 TaskAction

public @interface **TaskAction**

11.86.2 TaskActionParam

public @interface **TaskActionParam**

Marks method parameter to be treated as action parameter.

Each parameter in the given method has to have this annotation otherwise it will be a problem with the proper execution of the channel action.

See also: [TaskAction](#), [TaskChannel](#), [TaskAnnotationBeanPostProcessor](#)

11.86.3 TaskAnnotationBeanPostProcessor

public class **TaskAnnotationBeanPostProcessor** implements [BeanPostProcessor](#)

Factory class which is looking for classes with [TaskChannel](#) annotation to add them to the channel definition as channel action.

See also: [TaskAction](#), [TaskActionParam](#), [TaskChannel](#)

Constructors

TaskAnnotationBeanPostProcessor

public **TaskAnnotationBeanPostProcessor** ([BundleContext](#) *bundleContext*, [ChannelService](#) *channelService*)

Methods

postProcessAfterInitialization

public [Object](#) **postProcessAfterInitialization** ([Object](#) *bean*, [String](#) *beanName*)

postProcessBeforeInitialization

public [Object](#) **postProcessBeforeInitialization** ([Object](#) *bean*, [String](#) *beanName*)

processAnnotations

public void **processAnnotations** ([ApplicationContext](#) *applicationContext*)

11.86.4 TaskChannel

public @interface **TaskChannel**

11.87 org.motechproject.tasks.contract

11.87.1 ActionEventRequest

public class **ActionEventRequest**

Constructors

ActionEventRequest

public **ActionEventRequest** ([String](#) *displayName*, [String](#) *subject*, [String](#) *description*, [String](#) *serviceInterface*, [String](#) *serviceMethod*, [SortedSet](#)<[ActionParameterRequest](#)> *actionParameters*)

ActionEventRequest

```
public ActionEventRequest (String displayName, String subject, String description, String serviceInterface, String serviceMethod)
```

Methods

addParameter

```
public void addParameter (ActionParameterRequest parameter, boolean changeOrder)
```

equals

```
public boolean equals (Object obj)
```

getActionParameters

```
public SortedSet<ActionParameterRequest> getActionParameters ()
```

getDescription

```
public String getDescription ()
```

getDisplayName

```
public String getDisplayName ()
```

getServiceInterface

```
public String getServiceInterface ()
```

getServiceMethod

```
public String getServiceMethod ()
```

getSubject

```
public String getSubject ()
```

hasService

```
public boolean hasService ()
```

hasSubject

```
public boolean hasSubject ()
```

hashCode

```
public int hashCode ()
```

isValid

```
public boolean isValid ()
```

toString

```
public String toString ()
```

11.87.2 ActionParameterRequest

```
public class ActionParameterRequest implements Comparable<ActionParameterRequest>
```

Object representation of a parameter in the channel action request definition.

See also: [ActionEventRequest](#)

Constructors**ActionParameterRequest**

```
public ActionParameterRequest (String key, String displayName, Integer order, String type, boolean  
    required)
```

ActionParameterRequest

```
public ActionParameterRequest (String key, String displayName, Integer order, String type)
```

ActionParameterRequest

```
public ActionParameterRequest (String key, String displayName, Integer order)
```

ActionParameterRequest

```
public ActionParameterRequest (String key, String displayName)
```

Methods

compareTo

public int **compareTo** ([ActionParameterRequest](#) *o*)

equals

public boolean **equals** ([Object](#) *obj*)

getDisplayName

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

getKey

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

getOrder

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

getType

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

hashCode

public int **hashCode** ()

isRequired

public boolean **isRequired** ()

setOrder

public void **setOrder** (int *order*)

toString

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

11.87.3 ChannelRequest

public class **ChannelRequest**

Service layer object denoting a `org.motechproject.tasks.domain.Channel`. Used by `ChannelService`

Constructors

ChannelRequest

public **ChannelRequest** (*String displayName*, *String moduleName*, *String moduleVersion*, *String description*, *List<TriggerEventRequest> triggerTaskEvents*, *List<ActionEventRequest> actionTaskEvents*)

Methods

equals

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

getActionTaskEvents

public *List<ActionEventRequest>* **getActionTaskEvents** ()

getDescription

public *String* **getDescription** ()

getDisplayName

public *String* **getDisplayName** ()

getModuleName

public *String* **getModuleName** ()

getModuleVersion

public *String* **getModuleVersion** ()

getTriggerTaskEvents

public *List<TriggerEventRequest>* **getTriggerTaskEvents** ()

hashCode

```
public int hashCode ()
```

setModuleName

```
public void setModuleName (String moduleName)
```

setModuleVersion

```
public void setModuleVersion (String moduleVersion)
```

toString

```
public String toString ()
```

11.87.4 EventParameterRequest

```
public class EventParameterRequest
```

Constructors

EventParameterRequest

```
public EventParameterRequest (String displayName, String eventKey, String type)
```

EventParameterRequest

```
public EventParameterRequest (String displayName, String eventKey)
```

Methods

equals

```
public boolean equals (Object obj)
```

getDisplayName

```
public String getDisplayName ()
```

getEventKey

```
public String getEventKey ()
```


getType

```
public String getType ()
```

hashCode

```
public int hashCode ()
```

toString

```
public String toString ()
```

11.87.5 TriggerEventRequest

```
public class TriggerEventRequest
```

Constructors**TriggerEventRequest**

```
public TriggerEventRequest (String displayName, String subject, String description,  
                           List<EventParameterRequest> eventParameters)
```

Methods**equals**

```
public boolean equals (Object obj)
```

getDescription

```
public String getDescription ()
```

getDisplayName

```
public String getDisplayName ()
```

getEventParameters

```
public List<EventParameterRequest> getEventParameters ()
```

getSubject

```
public String getSubject ()
```

hashCode

```
public int hashCode ()
```

toString

```
public String toString ()
```

11.88 org.motechproject.tasks.domain

11.88.1 ActionEvent

```
public class ActionEvent extends TaskEvent
```

Constructors

ActionEvent

```
public ActionEvent ()
```

ActionEvent

```
public ActionEvent (String displayName, String subject, String description, SortedSet<ActionParameter>  
                    actionParameters)
```

ActionEvent

```
public ActionEvent (String displayName, String description, String serviceInterface, String serviceMethod,  
                    SortedSet<ActionParameter> actionParameters)
```

ActionEvent

```
public ActionEvent (String displayName, String subject, String description, String serviceInterface, String  
                    serviceMethod, SortedSet<ActionParameter> actionParameters)
```

ActionEvent

```
public ActionEvent (ActionEventRequest actionEventRequest)
```

Methods

accept

```
public boolean accept (TaskActionInformation info)
```

addParameter

public void **addParameter** ([ActionParameter](#) *parameter*, boolean *changeOrder*)

containsParameter

public boolean **containsParameter** ([String](#) *key*)

equals

public boolean **equals** ([Object](#) *obj*)

getActionParameters

public [SortedSet](#)<[ActionParameter](#)> **getActionParameters** ()

getServiceInterface

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

getServiceMethod

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

hasService

public boolean **hasService** ()

hashCode

public int **hashCode** ()

setActionParameters

public void **setActionParameters** ([SortedSet](#)<[ActionParameter](#)> *actionParameters*)

setServiceInterface

public void **setServiceInterface** ([String](#) *serviceInterface*)

setServiceMethod

public void **setServiceMethod** ([String](#) *serviceMethod*)

toString

```
public String toString()
```

11.88.2 ActionParameter

public class **ActionParameter** extends [Parameter](#) implements [Comparable<ActionParameter>](#)
Object representation of a parameter in the channel action definition.

See also: [ActionEvent](#)

Constructors

ActionParameter

```
public ActionParameter()
```

ActionParameter

```
public ActionParameter(String displayName, String key)
```

ActionParameter

```
public ActionParameter(String displayName, String key, boolean required)
```

ActionParameter

```
public ActionParameter(String displayName, String key, Integer order)
```

ActionParameter

```
public ActionParameter(String displayName, String key, ParameterType type)
```

ActionParameter

```
public ActionParameter(String displayName, String key, ParameterType type, boolean required)
```

ActionParameter

```
public ActionParameter(String displayName, String key, ParameterType type, Integer order)
```

ActionParameter

```
public ActionParameter(ActionParameterRequest actionParameterRequest)
```

ActionParameter

```
public ActionParameter (String displayName, String key, ParameterType type, Integer order, boolean required)
```

Methods

compareTo

```
public int compareTo (ActionParameter o)
```

equals

```
public boolean equals (Object obj)
```

getKey

```
public String getKey ()
```

getOrder

```
public Integer getOrder ()
```

hashCode

```
public int hashCode ()
```

isRequired

```
public boolean isRequired ()
```

setKey

```
public void setKey (String key)
```

setOrder

```
public void setOrder (Integer order)
```

setRequired

```
public void setRequired (boolean required)
```

`toString`

```
public String toString ()
```

11.88.3 Channel

```
public class Channel
```

Constructors

Channel

```
public Channel ()
```

Channel

```
public Channel (String displayName, String moduleName, String moduleVersion)
```

Channel

```
public Channel (String displayName, String moduleName, String moduleVersion, String description,  
                List<TriggerEvent> triggerTaskEvents, List<ActionEvent> actionTaskEvents)
```

Channel

```
public Channel (ChannelRequest channelRequest)
```

Methods

addActionTaskEvent

```
public void addActionTaskEvent (ActionEvent actionEvent)
```

containsAction

```
public boolean containsAction (TaskActionInformation actionInformation)
```

containsTrigger

```
public boolean containsTrigger (TaskTriggerInformation triggerInformation)
```

equals

```
public boolean equals (Object obj)
```

getAction

```
public ActionEvent getAction (TaskActionInformation actionInformation)
```

getActionTaskEvents

```
public List<ActionEvent> getActionTaskEvents ()
```

getDescription

```
public String getDescription ()
```

getDisplayName

```
public String getDisplayName ()
```

getModuleName

```
public String getModuleName ()
```

getModuleVersion

```
public String getModuleVersion ()
```

getTrigger

```
public TriggerEvent getTrigger (TaskTriggerInformation triggerInformation)
```

getTriggerTaskEvents

```
public List<TriggerEvent> getTriggerTaskEvents ()
```

hashCode

```
public int hashCode ()
```

setActionTaskEvents

```
public void setActionTaskEvents (List<ActionEvent> actionTaskEvents)
```

setDescription

```
public void setDescription (String description)
```

setDisplayNames

```
public void setDisplayNames (String displayName)
```

setModuleName

```
public void setModuleName (String moduleName)
```

setModuleVersion

```
public void setModuleVersion (String moduleVersion)
```

setTriggerTaskEvents

```
public void setTriggerTaskEvents (List<TriggerEvent> triggerTaskEvents)
```

toString

```
public String toString ()
```

11.88.4 ChannelRegisterEvent

```
public class ChannelRegisterEvent
```

A wrapper over a `MotechEvent` with subject { `@value` EventSubjects#CHANNEL_REGISTER_SUBJECT }.
Raised when a channel is registered with the tasks module

Constructors

ChannelRegisterEvent

```
public ChannelRegisterEvent (String moduleName)
```

ChannelRegisterEvent

```
public ChannelRegisterEvent (MotechEvent motechEvent)
```

Methods

getChannelModuleName

```
public String getChannelModuleName ()
```

toMotechEvent

```
public MotechEvent toMotechEvent ()
```


11.88.5 DataSource

public class **DataSource** extends `TaskConfigStep`

Constructors

DataSource

public **DataSource** ()

DataSource

public **DataSource** (`Long providerId`, `Long objectId`, `String type`, `String name`, `List<Lookup> lookup`,
`boolean failIfDataNotFound`)

DataSource

public **DataSource** (`String providerName`, `Long providerId`, `Long objectId`, `String type`, `String name`,
`List<Lookup> lookup`, `boolean failIfDataNotFound`)

Methods

equals

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

getLookup

public `List<Lookup>` **getLookup** ()

getName

public `String` **getName** ()

getObjectId

public `Long` **getObjectId** ()

getProviderId

public `Long` **getProviderId** ()

getProviderName

public `String` **getProviderName** ()

getType

public **String** **getType** ()

hashCode

public int **hashCode** ()

isFailIfDataNotFound

public boolean **isFailIfDataNotFound** ()

objectEquals

public boolean **objectEquals** (*Long providerId*, *Long objectId*, *String type*)

setFailIfDataNotFound

public void **setFailIfDataNotFound** (boolean *failIfDataNotFound*)

setLookup

public void **setLookup** (*Object lookup*)

setName

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

setObjectId

public void **setObjectId** (*Long objectId*)

setProviderId

public void **setProviderId** (*Long providerId*)

setProviderName

public void **setProviderName** (*String providerName*)

setType

public void **setType** (*String type*)

toString

```
public String toString ()
```

11.88.6 EventParameter

```
public class EventParameter extends Parameter
```

Constructors

EventParameter

```
public EventParameter ()
```

EventParameter

```
public EventParameter (String displayName, String eventKey)
```

EventParameter

```
public EventParameter (String displayName, String eventKey, ParameterType type)
```

EventParameter

```
public EventParameter (EventParameterRequest eventParameterRequest)
```

Methods

equals

```
public boolean equals (Object obj)
```

getEventKey

```
public String getEventKey ()
```

hashCode

```
public int hashCode ()
```

setEventKey

```
public void setEventKey (String eventKey)
```

toString

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

11.88.7 FieldParameter

public class **FieldParameter** extends **Parameter**

Constructors

FieldParameter

public **FieldParameter** ()

FieldParameter

public **FieldParameter** (*String displayName*, *String fieldKey*)

FieldParameter

public **FieldParameter** (*String displayName*, *String fieldKey*, *ParameterType type*)

Methods

equals

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

getFieldKey

public **String** **getFieldKey** ()

hashCode

public int **hashCode** ()

setFieldKey

public void **setFieldKey** (*String fieldKey*)

toString

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

11.88.8 Filter

public class **Filter** implements [Serializable](#)

Constructors

Filter

public **Filter** ()

Filter

public **Filter** ([EventParameter](#) *eventParameter*, boolean *negationOperator*, [String](#) *operator*, [String](#) *expression*)

Filter

public **Filter** ([String](#) *displayName*, [String](#) *key*, [ParameterType](#) *type*, boolean *negationOperator*, [String](#) *operator*, [String](#) *expression*)

Methods

equals

public boolean **equals** ([Object](#) *obj*)

getDisplayName

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

getExpression

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

getKey

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

getOperator

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

getType

public [ParameterType](#) **getType** ()

hashCode

public int **hashCode** ()

isNegationOperator

public boolean **isNegationOperator** ()

setDisplayName

public void **setDisplayName** (*String displayName*)

setExpression

public void **setExpression** (*String expression*)

setKey

public void **setKey** (*String key*)

setNegationOperator

public void **setNegationOperator** (boolean *negationOperator*)

setOperator

public void **setOperator** (*String operator*)

setType

public void **setType** (*ParameterType type*)

toString

public *String* **toString** ()

11.88.9 FilterSet

public class **FilterSet** extends *TaskConfigStep*

Constructors

FilterSet

public **FilterSet** ()

FilterSet

```
public FilterSet (List<Filter> filters)
```

Methods

addFilter

```
public void addFilter (Filter filter)
```

equals

```
public boolean equals (Object obj)
```

getFilters

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

hashCode

```
public int hashCode ()
```

setFilters

```
public void setFilters (List<Filter> filters)
```

toString

```
public String toString ()
```

11.88.10 KeyInformation

```
public final class KeyInformation
```

Object representation of dragged field from trigger or data source.

This class represents a single dragged field from trigger or data source. This class does not expose a public constructor. You have to use `parse(String)` method if you want to parse single field or `parseAll(String)` if you want to get all fields from a given string.

Fields

ADDITIONAL_DATA_PREFIX

```
public static final String ADDITIONAL_DATA_PREFIX
```

Prefix which is used for data source fields.

TRIGGER_PREFIX

public static final `String` **TRIGGER_PREFIX**
Prefix which is used for trigger fields.

Methods

equals

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

fromAdditionalData

public boolean **fromAdditionalData** ()
Check if the field is from the data source.
Returns true if the field is from the data source otherwise false

fromTrigger

public boolean **fromTrigger** ()
Check if the field is from the trigger.
Returns true if the field is from the trigger otherwise false

getDataProviderId

public `Long` **getDataProviderId** ()

getKey

public `String` **getKey** ()

getManipulations

public `List<String>` **getManipulations** ()
Get manipulations assigned to the field.
Returns list of manipulations

getObjectId

public `Long` **getObjectId** ()

getObjectType

public `String` **getObjectType** ()

getOriginalKey

```
public String getOriginalKey ()
```

Get original representation of the dragged field.

Returns string representation of the field

getPrefix

```
public String getPrefix ()
```

hasManipulations

```
public boolean hasManipulations ()
```

Check if the field has any manipulations.

Returns true if the field has manipulations otherwise false

hashCode

```
public int hashCode ()
```

parse

```
public static KeyInformation parse (String input)
```

Parse given string to instance of `KeyInformation`.

This method should be used to convert string representation of dragged field to instance of `KeyInformation`.

Argument has adhere to one of the following format:

- trigger field format: `trigger.eventKey`
- data source format: `ad.dataProviderId.objectType#objectId.fieldKey`

Argument can also contain list of manipulation which should be executed on field before it will be used by `org.motechproject.tasks.service.TaskTriggerHandler` class. Manipulations should be connected together by the `?` character.

Example of input argument:

- ad.279f5fdf60700d9717270b1ae3011eb1.CaseInfo#0.fieldValues.phu_id
- trigger.message?format(Ala,cat)?capitalize

Parameters

- **input** – string representation of a dragged field from trigger or data source

Throws

- **IllegalArgumentException** – exception is thrown if format for data source field is incorrect or if the dragged field is not from trigger or data source.

Returns Object representation of a dragged field

parseAll

public static [List<KeyInformation>](#) **parseAll** ([String](#) input)

Find all fields from given input and convert them to the instance of [KeyInformation](#).

This method should be used to find and convert all of string representation of the field from trigger and/or data sources. Fields in input have to adhere to one of the following formats:

- trigger field format: `{{trigger.eventKey}}`
- data source format: `{{ad.dataProviderId.objectType#objectId.fieldKey}}`

To find fields in the input argument this method uses regular expression. When field is found it is converted to an instance of [KeyInformation](#) by using the [parse\(String\)](#) method.

Fields are found by the following regular expression: `{{((.*?))}}(?![^(]*))`. The expression searches for strings that start with `{{` and end with `}}` and are not within `(` and `)`. Because of manipulations which contain additional data in `(...)` needed to execute manipulation on the field (e.g.: join needs to have the join character) and the text in `(...)` can be another string representation of the dragged field, the expression has to check if the field has this kind of manipulation.

Example of input argument:

- `{{trigger.message?format(Ala,cat)?capitalize}}`
- You get the following message: `{{trigger.message}}`

Parameters

- **input** – string with one or more string representation of dragged fields from trigger and/or data sources

Throws

- **IllegalArgumentException** – in the same situations as the [parse\(String\)](#) method.

Returns list of object representation of dragged fields.

11.88.11 Lookup

public class **Lookup** implements [Serializable](#)

Constructors

Lookup

public **Lookup** ()

Lookup

public **Lookup** ([String](#) field, [String](#) value)

Methods

`equals`

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

`getField`

public *String* **getField** ()

`getValue`

public *String* **getValue** ()

`hashCode`

public int **hashCode** ()

`setField`

public void **setField** (*String field*)

`setValue`

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

11.88.12 LookupFieldsParameter

public class **LookupFieldsParameter**

Constructors

LookupFieldsParameter

public **LookupFieldsParameter** ()

LookupFieldsParameter

public **LookupFieldsParameter** (*String displayName*, *List<String> fields*)

Methods

`equals`

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

getDisplayName

```
public String getDisplayName ()
```

getFields

```
public List<String> getFields ()
```

hashCode

```
public int hashCode ()
```

setDisplayName

```
public void setDisplayName (String displayName)
```

setFields

```
public void setFields (List<String> fields)
```

toString

```
public String toString ()
```

11.88.13 ManipulationTarget

```
public enum ManipulationTarget
```

Defines the target of various manipulations used in a task for both triggers and data sources.

Enum Constants**ALL**

```
public static final ManipulationTarget ALL
```

DATE

```
public static final ManipulationTarget DATE
```

STRING

```
public static final ManipulationTarget STRING
```

11.88.14 ManipulationType

public enum **ManipulationType**

Defines the type of various manipulations used in a task for both triggers and data sources.

Enum Constants

CAPITALIZE

public static final ManipulationType **CAPITALIZE**

DATETIME

public static final ManipulationType **DATETIME**

FORMAT

public static final ManipulationType **FORMAT**

JOIN

public static final ManipulationType **JOIN**

PLUSDAYS

public static final ManipulationType **PLUSDAYS**

SPLIT

public static final ManipulationType **SPLIT**

SUBSTRING

public static final ManipulationType **SUBSTRING**

TOLOWER

public static final ManipulationType **TOLOWER**

TOUPPER

public static final ManipulationType **TOUPPER**

UNKNOWN

public static final [ManipulationType](#) UNKNOWN

URLENCODE

public static final [ManipulationType](#) URLENCODE

11.88.15 OperatorType

public enum **OperatorType**

Object representation of available operators in filter definition.

Enum Constants

AFTER

public static final [OperatorType](#) AFTER

AFTER_NOW

public static final [OperatorType](#) AFTER_NOW

BEFORE

public static final [OperatorType](#) BEFORE

BEFORE_NOW

public static final [OperatorType](#) BEFORE_NOW

CONTAINS

public static final [OperatorType](#) CONTAINS

ENDSWITH

public static final [OperatorType](#) ENDSWITH

EQUALS

public static final [OperatorType](#) EQUALS

EQ_NUMBER

public static final [OperatorType](#) **EQ_NUMBER**

EXIST

public static final [OperatorType](#) **EXIST**

GT

public static final [OperatorType](#) **GT**

LESS_DAYS_FROM_NOW

public static final [OperatorType](#) **LESS_DAYS_FROM_NOW**

LESS_MONTHS_FROM_NOW

public static final [OperatorType](#) **LESS_MONTHS_FROM_NOW**

LT

public static final [OperatorType](#) **LT**

MORE_DAYS_FROM_NOW

public static final [OperatorType](#) **MORE_DAYS_FROM_NOW**

MORE_MONTHS_FROM_NOW

public static final [OperatorType](#) **MORE_MONTHS_FROM_NOW**

STARTSWITH

public static final [OperatorType](#) **STARTSWITH**

11.88.16 Parameter

public abstract class **Parameter** implements [Serializable](#)

Constructors**Parameter**

protected **Parameter** ()

Parameter

protected **Parameter** (*String displayName*, *ParameterType type*)

Methods

equals

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

getDisplayName

public *String* **getDisplayName** ()

getType

public *ParameterType* **getType** ()

hashCode

public int **hashCode** ()

setDisplayName

public void **setDisplayName** (*String displayName*)

setType

public void **setType** (*ParameterType type*)

toString

public *String* **toString** ()

11.88.17 ParameterType

public enum **ParameterType**

Defines the type of various values used in a task including trigger parameters, action parameters and data source object fields.

Enum Constants

BOOLEAN

public static final *ParameterType* **BOOLEAN**

DATE

public static final [ParameterType](#) **DATE**

DOUBLE

public static final [ParameterType](#) **DOUBLE**

INTEGER

public static final [ParameterType](#) **INTEGER**

LIST

public static final [ParameterType](#) **LIST**

LONG

public static final [ParameterType](#) **LONG**

MAP

public static final [ParameterType](#) **MAP**

TEXTAREA

public static final [ParameterType](#) **TEXTAREA**

TIME

public static final [ParameterType](#) **TIME**

UNICODE

public static final [ParameterType](#) **UNICODE**

UNKNOWN

public static final [ParameterType](#) **UNKNOWN**

11.88.18 SettingsDto

public class **SettingsDto**

Methods

getTaskPossibleErrors

```
public String getTaskPossibleErrors ()
```

isValid

```
public boolean isValid ()
```

setTaskPossibleErrors

```
public void setTaskPossibleErrors (String taskPossibleErrors)
```

11.88.19 Task

```
public class Task
```

A task is set of actions that are executed in response to a trigger. The actions and the trigger are defined by their respective [Channels](#).

Constructors

Task

```
public Task ()
```

Task

```
public Task (String name, TaskTriggerInformation trigger, List<TaskActionInformation> actions)
```

Task

```
public Task (String name, TaskTriggerInformation trigger, List<TaskActionInformation> actions, TaskConfig taskConfig, boolean enabled, boolean hasRegisteredChannel)
```

Methods

addAction

```
public void addAction (TaskActionInformation action)
```

addValidationErrors

```
public void addValidationErrors (Set<TaskError> validationErrors)
```

equals

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

getActions

public *List*<*TaskActionInformation*> **getActions** ()

getDescription

public *String* **getDescription** ()

getId

public *Long* **getId** ()

getName

public *String* **getName** ()

getTaskConfig

public *TaskConfig* **getTaskConfig** ()

getTrigger

public *TaskTriggerInformation* **getTrigger** ()

getValidationErrors

public *Set*<*TaskError*> **getValidationErrors** ()

hasRegisteredChannel

public boolean **hasRegisteredChannel** ()

hasValidationErrors

public boolean **hasValidationErrors** ()

hashCode

public int **hashCode** ()

isEnabled

public boolean **isEnabled**()

removeValidationError

public void **removeValidationError**(String *message*)

setActions

public void **setActions**(List<TaskActionInformation> *actions*)

setDescription

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

setEnabled

public void **setEnabled**(boolean *enabled*)

setHasRegisteredChannel

public void **setHasRegisteredChannel**(boolean *hasRegisteredChannel*)

setId

public void **setId**(Long *id*)

setName

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

setTaskConfig

public void **setTaskConfig**(TaskConfig *taskConfig*)

setTrigger

public void **setTrigger**(TaskTriggerInformation *trigger*)

setValidationErrors

public void **setValidationErrors**(Set<TaskError> *validationErrors*)

`toString`

```
public String toString()
```

11.88.20 TaskActionInformation

public class **TaskActionInformation** extends `TaskEventInformation`

Represents an action event configured with a `Task`

Constructors

TaskActionInformation

```
public TaskActionInformation()
```

TaskActionInformation

```
public TaskActionInformation(String displayName, String channelName, String moduleName, String  
                             moduleVersion, String subject)
```

TaskActionInformation

```
public TaskActionInformation(String displayName, String channelName, String moduleName, String  
                             moduleVersion, String subject, Map<String, String> values)
```

TaskActionInformation

```
public TaskActionInformation(String displayName, String channelName, String moduleName, String  
                             moduleVersion, String serviceInterface, String serviceMethod)
```

TaskActionInformation

```
public TaskActionInformation(String displayName, String channelName, String moduleName, String  
                             moduleVersion, String subject, String serviceInterface, String ser-  
viceMethod, Map<String, String> values)
```

Methods

equals

```
public boolean equals(Object obj)
```

getServiceInterface

```
public String getServiceInterface()
```

getServiceMethod

```
public String getServiceMethod ()
```

getValues

```
public Map<String, String> getValues ()
```

hasService

```
public boolean hasService ()
```

hashCode

```
public int hashCode ()
```

setServiceInterface

```
public void setServiceInterface (String serviceInterface)
```

setServiceMethod

```
public void setServiceMethod (String serviceMethod)
```

setValues

```
public void setValues (Map<String, String> values)
```

toString

```
public String toString ()
```

11.88.21 TaskActivity

```
public class TaskActivity implements Comparable<TaskActivity>
```

Constructors

TaskActivity

```
public TaskActivity ()
```

TaskActivity

```
public TaskActivity (String message, Long task, TaskActivityType activityType)
```

TaskActivity

```
public TaskActivity (String message, String field, Long task, TaskActivityType activityType)
```

TaskActivity

```
public TaskActivity (String message, List<String> fields, Long task, TaskActivityType activityType)
```

TaskActivity

```
public TaskActivity (String message, List<String> fields, Long task, TaskActivityType activityType, String  
stackTraceElement)
```

Methods**compareTo**

```
public int compareTo (TaskActivity o)
```

equals

```
public boolean equals (Object obj)
```

getActivityType

```
public TaskActivityType getActivityType ()
```

getDate

```
public DateTime getDate ()
```

getFields

```
public List<String> getFields ()
```

getMessage

```
public String getMessage ()
```

getStackTraceElement

```
public String getStackTraceElement ()
```

getTask

public Long **getTask** ()

hashCode

public int **hashCode** ()

setActivityType

public void **setActivityType** (TaskActivityType *activityType*)

setDate

public void **setDate** (DateTime *date*)

setField

public void **setField** (String *field*)

setFields

public void **setFields** (List<String> *fields*)

setMessage

public void **setMessage** (String *message*)

setStackTraceElement

public void **setStackTraceElement** (String *stackTraceElement*)

setTask

public void **setTask** (Long *task*)

toString

public String **toString** ()

11.88.22 TaskActivityType

public enum **TaskActivityType**

Enum Constants

ERROR

public static final [TaskActivityType](#) **ERROR**

SUCCESS

public static final [TaskActivityType](#) **SUCCESS**

WARNING

public static final [TaskActivityType](#) **WARNING**

11.88.23 TaskBuilder

public class **TaskBuilder**

Constructors

TaskBuilder

public **TaskBuilder** ()

Methods

addAction

public [TaskBuilder](#) **addAction** ([TaskActionInformation](#) *action*)

addDataSource

public [TaskBuilder](#) **addDataSource** ([DataSource](#) *dataSource*)

addFilterSet

public [TaskBuilder](#) **addFilterSet** ([FilterSet](#) *filterSet*)

build

public [Task](#) **build** ()

clear

public [TaskBuilder](#) **clear** ()

isEnabled

public [TaskBuilder](#) **isEnabled** (boolean *enabled*)

withDescription

public [TaskBuilder](#) **withDescription** ([String](#) *description*)

withId

public [TaskBuilder](#) **withId** ([Long](#) *id*)

withName

public [TaskBuilder](#) **withName** ([String](#) *name*)

withTaskConfig

public [TaskBuilder](#) **withTaskConfig** ([TaskConfig](#) *taskConfig*)

withTrigger

public [TaskBuilder](#) **withTrigger** ([TaskTriggerInformation](#) *trigger*)

11.88.24 TaskConfig

public class **TaskConfig** implements [Serializable](#)

Methods

add

public [TaskConfig](#) **add** ([TaskConfigStep](#)... *configSteps*)

addAll

public [TaskConfig](#) **addAll** ([SortedSet](#)<[TaskConfigStep](#)> *set*)

equals

public boolean **equals** ([Object](#) *obj*)

getDataSource

public [DataSource](#) **getDataSource** ([Long](#) *providerId*, [Long](#) *objectId*, [String](#) *objectType*)

getDataSources

```
public List<DataSource> getDataSources ()
```

getDataSources

```
public SortedSet<DataSource> getDataSources (Long providerId)
```

getFilters

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

getSteps

```
public SortedSet<TaskConfigStep> getSteps ()
```

hashCode

```
public int hashCode ()
```

removeAll

```
public TaskConfig removeAll ()
```

removeDataSources

```
public TaskConfig removeDataSources ()
```

removeFilterSets

```
public TaskConfig removeFilterSets ()
```

setDataSources

```
public void setDataSources (List<DataSource> dataSources)
```

setFilters

```
public void setFilters (List<FilterSet> filters)
```

toString

```
public String toString ()
```

11.88.25 TaskConfigStep

public abstract class **TaskConfigStep** implements [Comparable<TaskConfigStep>](#), [Serializable](#)

Methods

compareTo

public int **compareTo** ([TaskConfigStep o](#))

equals

public boolean **equals** ([Object obj](#))

getOrder

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

hashCode

public int **hashCode** ()

setOrder

public void **setOrder** ([Integer order](#))

toString

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

11.88.26 TaskDataProvider

public class **TaskDataProvider**

The `TaskDataProvider` class contains all informations about data providers used in tasks

Constructors

TaskDataProvider

public **TaskDataProvider** ()

TaskDataProvider

public **TaskDataProvider** ([String name](#), [List<TaskDataProviderObject> objects](#))

Methods

containsProviderObject

public boolean **containsProviderObject** (*String type*)

containsProviderObjectField

public boolean **containsProviderObjectField** (*String type*, *String fieldKey*)

containsProviderObjectLookup

public boolean **containsProviderObjectLookup** (*String type*, *String lookupField*)

equals

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

getId

public *Long* **getId** ()

getKeyType

public *String* **getKeyType** (*String key*)

getName

public *String* **getName** ()

getObjects

public *List*<*TaskDataProviderObject*> **getObjects** ()

getProviderObject

public *TaskDataProviderObject* **getProviderObject** (*String type*)

hashCode

public int **hashCode** ()

setId

public void **setId** (*Long id*)

setName

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

setObjects

public void **setObjects** (*List<TaskDataProviderObject> objects*)

toString

public *String* **toString** ()

11.88.27 TaskDataProviderObject

public class **TaskDataProviderObject** implements *Serializable*

Constructors

TaskDataProviderObject

public **TaskDataProviderObject** ()

TaskDataProviderObject

public **TaskDataProviderObject** (*String displayName*, *String type*, *List<LookupFieldsParameter> lookupFields*, *List<FieldParameter> fields*)

Methods

containsField

public boolean **containsField** (*String fieldKey*)

equals

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

getDisplayName

public *String* **getDisplayName** ()

getFields

public *List<FieldParameter>* **getFields** ()

getLookupFields

```
public List<LookupFieldsParameter> getLookupFields ()
```

getType

```
public String getType ()
```

hashCode

```
public int hashCode ()
```

setDisplayName

```
public void setDisplayName (String displayName)
```

setFields

```
public void setFields (List<FieldParameter> fields)
```

setLookupFields

```
public void setLookupFields (List<Object> lookupFields)
```

setType

```
public void setType (String type)
```

toString

```
public String toString ()
```

11.88.28 TaskError

```
public class TaskError implements Serializable
```

Constructors**TaskError**

```
public TaskError ()
```

TaskError

```
public TaskError (TaskErrorType type, String... args)
```

TaskError

public **TaskError** (*String message*, *String... args*)

Methods

equals

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

getArgs

public *List<String>* **getArgs** ()

getMessage

public *String* **getMessage** ()

hashCode

public int **hashCode** ()

setArgs

public void **setArgs** (*List<String> args*)

setMessage

public void **setMessage** (*String message*)

toString

public *String* **toString** ()

11.88.29 TaskErrorType

public enum **TaskErrorType**

Enum Constants

BLANK

public static final *TaskErrorType* **BLANK**

EMPTY_COLLECTION

public static final [TaskErrorType](#) **EMPTY_COLLECTION**

NULL

public static final [TaskErrorType](#) **NULL**

VERSION

public static final [TaskErrorType](#) **VERSION**

11.88.30 TaskEvent

public abstract class **TaskEvent** implements [Serializable](#)

Constructors

TaskEvent

protected **TaskEvent** ()

TaskEvent

protected **TaskEvent** (*String description*, *String displayName*, *String subject*)

Methods

containsParameter

public boolean **containsParameter** (*String key*)

equals

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

equalsSubject

protected boolean **equalsSubject** (*String subject*)

getDescription

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

getDisplayName

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

getSubject

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

hasSubject

public boolean **hasSubject** ()

hashCode

public int **hashCode** ()

setDescription

public void **setDescription** ([String](#) *description*)

setDisplayName

public void **setDisplayName** ([String](#) *displayName*)

setSubject

public void **setSubject** ([String](#) *subject*)

toString

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

11.88.31 TaskEventInformation

public abstract class **TaskEventInformation** implements [Serializable](#)

Constructors

TaskEventInformation

public **TaskEventInformation** ()

TaskEventInformation

```
public TaskEventInformation (String displayName, String channelName, String moduleName, String  
                             moduleVersion, String subject)
```

Methods

equals

```
public boolean equals (Object obj)
```

getChannelName

```
public String getChannelName ()
```

getDisplayName

```
public String getDisplayName ()
```

getModuleName

```
public String getModuleName ()
```

getModuleVersion

```
public String getModuleVersion ()
```

getSubject

```
public String getSubject ()
```

hasSubject

```
public boolean hasSubject ()
```

hashCode

```
public int hashCode ()
```

setChannelName

```
public void setChannelName (String channelName)
```

setDisplayNames

public void **setDisplayNames** (*String displayName*)

setModuleName

public void **setModuleName** (*String moduleName*)

setModuleVersion

public void **setModuleVersion** (*String moduleVersion*)

setSubject

public void **setSubject** (*String subject*)

toString

public *String* **toString** ()

11.88.32 TaskTriggerInformation

public class **TaskTriggerInformation** extends *TaskEventInformation*

Constructors

TaskTriggerInformation

public **TaskTriggerInformation** ()

TaskTriggerInformation

public **TaskTriggerInformation** (*String displayName*, *String channelName*, *String moduleName*, *String moduleVersion*, *String subject*)

TaskTriggerInformation

public **TaskTriggerInformation** (*TaskTriggerInformation other*)

11.88.33 TriggerEvent

public class **TriggerEvent** extends *TaskEvent*

The *TriggerEvent* class is responsible for storing data about event

Constructors

TriggerEvent

```
public TriggerEvent ()
```

TriggerEvent

```
public TriggerEvent (String displayName, String subject, String description, List<EventParameter> event-  
Parameters)
```

TriggerEvent

```
public TriggerEvent (TriggerEventRequest triggerEventRequest)
```

Methods

containsParameter

```
public boolean containsParameter (String key)
```

equals

```
public boolean equals (Object obj)
```

getEventParameters

```
public List<EventParameter> getEventParameters ()
```

getKeyType

```
public String getKeyType (String key)
```

hashCode

```
public int hashCode ()
```

setEventParameters

```
public void setEventParameters (List<EventParameter> eventParameters)
```

toString

```
public String toString ()
```

11.89 org.motechproject.tasks.ex

11.89.1 ActionNotFoundException

public class **ActionNotFoundException** extends [Exception](#)

Constructors

ActionNotFoundException

public **ActionNotFoundException** ([String](#) *message*)

11.89.2 TaskHandlerException

public class **TaskHandlerException** extends [Exception](#)

Constructors

TaskHandlerException

public **TaskHandlerException** ([TaskFailureCause](#) *failureCause*, [String](#) *messageKey*)

TaskHandlerException

public **TaskHandlerException** ([TaskFailureCause](#) *failureCause*, [String](#) *messageKey*, [String](#)... *args*)

TaskHandlerException

public **TaskHandlerException** ([TaskFailureCause](#) *failureCause*, [String](#) *messageKey*, [Throwable](#) *cause*,
[String](#)... *args*)

Methods

getArgs

public [List](#)<[String](#)> **getArgs** ()

getFailureCause

public [TaskFailureCause](#) **getFailureCause** ()

getMessage

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

11.89.3 TaskNotFoundException

public class **TaskNotFoundException** extends [IllegalArgumentException](#)

Constructors

TaskNotFoundException

public **TaskNotFoundException** ([Long](#) *taskId*)

11.89.4 TriggerNotFoundException

public class **TriggerNotFoundException** extends [Exception](#)

Constructors

TriggerNotFoundException

public **TriggerNotFoundException** ([String](#) *message*)

11.89.5 ValidationException

public class **ValidationException** extends [IllegalArgumentException](#)

Constructors

ValidationException

public **ValidationException** ([String](#) *objectType*, [Set](#)<[TaskError](#)> *taskErrors*)

Methods

getMessage

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

getTaskErrors

public [Set](#)<[TaskError](#)> **getTaskErrors** ()

11.90 org.motechproject.tasks.json

11.90.1 ActionEventRequestDeserializer

public class **ActionEventRequestDeserializer** implements [JsonDeserializer](#)<[ActionEventRequest](#)>

Fields

ACTION_PARAMETERS_FIELD

```
public static final String ACTION_PARAMETERS_FIELD
```

DESCRIPTION_FIELD

```
public static final String DESCRIPTION_FIELD
```

DISPLAY_NAME_FIELD

```
public static final String DISPLAY_NAME_FIELD
```

SERVICE_INTERFACE_FIELD

```
public static final String SERVICE_INTERFACE_FIELD
```

SERVICE_METHOD_FIELD

```
public static final String SERVICE_METHOD_FIELD
```

SUBJECT_FIELD

```
public static final String SUBJECT_FIELD
```

Methods

deserialize

```
public ActionEventRequest deserialize(JsonElement element, Type type, JsonDeserializationContext context)
```

11.90.2 TaskConfigDeserializer

```
public class TaskConfigDeserializer extends JsonSerializer<TaskConfig>
```

Methods

deserialize

```
public TaskConfig deserialize(JsonParser parser, DeserializationContext context)
```


11.90.3 TaskDeserializer

public class **TaskDeserializer** extends JsonSerializer<Task>

Methods

deserialize

public Task **deserialize** (JsonParser *jsonParser*, DeserializationContext *deserializationContext*)

11.91 org.motechproject.tasks.repository

11.91.1 ChannelsDataService

public interface **ChannelsDataService** extends MotechDataService<Channel>

Methods

findByModuleName

Channel **findByModuleName** (String *moduleName*)

11.91.2 TaskActivitiesDataService

public interface **TaskActivitiesDataService** extends MotechDataService<TaskActivity>

Methods

byTask

List<TaskActivity> **byTask** (Long *task*)

11.91.3 TasksDataService

public interface **TasksDataService** extends MotechDataService<Task>

11.92 org.motechproject.tasks.service

11.92.1 ChannelService

public interface **ChannelService**
Manages CRUD operations for a Channel.

Methods

addOrUpdate

void **addOrUpdate** ([Channel](#) *channel*)

getAllChannels

[List<Channel>](#) **getAllChannels** ()

getChannel

[Channel](#) **getChannel** ([String](#) *moduleName*)

getChannelIcon

[BundleIcon](#) **getChannelIcon** ([String](#) *moduleName*)

registerChannel

void **registerChannel** ([ChannelRequest](#) *channelRequest*)

registerChannel

void **registerChannel** ([InputStream](#) *stream*, [String](#) *moduleName*, [String](#) *moduleVersion*)

11.92.2 DataProviderDataService

public interface **DataProviderDataService** extends [MotechDataService<TaskDataProvider>](#)

Methods

findByName

[TaskDataProvider](#) **findByName** ([String](#) *name*)

11.92.3 DataSourceObject

public class **DataSourceObject**

DataSourceObject is the result of a `org.motechproject.commons.api.DataProvider` lookup.

Constructors

DataSourceObject

```
public DataSourceObject (String objectId, Object objectValue, boolean failIfNotFound)
```

Methods

equals

```
public boolean equals (Object o)
```

getObjectId

```
public String getObjectId ()
```

getObjectValue

```
public Object getObjectValue ()
```

hashCode

```
public int hashCode ()
```

isFailIfNotFound

```
public boolean isFailIfNotFound ()
```

11.92.4 HandlerPredicates

```
final class HandlerPredicates
```

Utility class defining filters over some collections.

Methods

activeTasks

```
public static Predicate activeTasks ()
```

withServiceName

```
public static Predicate withServiceName (String serviceName)
```

11.92.5 KeyEvaluator

public class **KeyEvaluator**

KeyEvaluator evaluates the value of a key in the context of a task which is used to execute filters and actions.

Constructors

KeyEvaluator

public **KeyEvaluator** ([TaskContext](#) *taskContext*)

Methods

evaluateTemplateString

public [String](#) **evaluateTemplateString** ([String](#) *template*)

getValue

public [Object](#) **getValue** ([KeyInformation](#) *keyInformation*)

manipulate

[String](#) **manipulate** ([String](#) *manipulation*, [String](#) *value*)

11.92.6 MethodHandler

class **MethodHandler**

Utility class used by [TaskTriggerHandler](#) to construct a list of parameter types of the method in the correct order.

See also: [TaskTriggerHandler](#)

Constructors

MethodHandler

public **MethodHandler** ([ActionEvent](#) *action*, [Map](#)<[String](#), [Object](#)> *parameters*)

Methods

getClasses

public [Class](#)[] **getClasses** ()

getObjects

```
public Object[] getObjects ()
```

isParametrized

```
public boolean isParametrized ()
```

11.92.7 TaskActionExecutor

```
class TaskActionExecutor
```

Builds action parameters from `TaskContext` and executes the action by invoking its service or raising its event.

Constructors

TaskActionExecutor

```
TaskActionExecutor (TaskService taskService, TaskActivityService activityService, EventRelay eventRelay)
```

Methods

createParameters

```
Map<String, Object> createParameters (TaskActionInformation info, ActionEvent action)
```

execute

```
void execute (Task task, TaskActionInformation actionInformation, TaskContext taskContext)
```

setBundleContext

```
void setBundleContext (BundleContext bundleContext)
```

11.92.8 TaskActivityService

```
public interface TaskActivityService
```

Methods

addError

```
void addError (Task task, TaskHandlerException e)
```

addSuccess

void **addSuccess** ([Task](#) *task*)

addWarning

void **addWarning** ([Task](#) *task*)

addWarning

void **addWarning** ([Task](#) *task*, [String](#) *key*, [String](#) *value*)

addWarning

void **addWarning** ([Task](#) *task*, [String](#) *key*, [String](#) *field*, [Exception](#) *e*)

deleteActivitiesForTask

void **deleteActivitiesForTask** ([Long](#) *taskId*)

errorsFromLastRun

[List](#)<[TaskActivity](#)> **errorsFromLastRun** ([Task](#) *task*)

getAllActivities

[List](#)<[TaskActivity](#)> **getAllActivities** ()

getTaskActivities

[List](#)<[TaskActivity](#)> **getTaskActivities** ([Long](#) *taskId*)

11.92.9 TaskContext

public class **TaskContext**

TaskContext holds task trigger event and data provider lookup objects that are used while executing filters/actions.

Constructors

TaskContext

public **TaskContext** ([Task](#) *task*, [MotechEvent](#) *event*, [TaskActivityService](#) *activityService*)

Methods

addDataSourceObject

```
public void addDataSourceObject (String objectId, Object dataSourceObject, boolean failIfDataNot-Found)
```

getDataSourceObjectValue

```
public Object getDataSourceObjectValue (String objectId, String field, String objectType)
```

getTask

```
public Task getTask ()
```

getTriggerParameters

```
public Map<String, Object> getTriggerParameters ()
```

getTriggerValue

```
public Object getTriggerValue (String key)
```

publishWarningActivity

```
public void publishWarningActivity (String message, String field)
```

11.92.10 TaskDataProviderService

```
public interface TaskDataProviderService
```

Methods

getProvider

```
TaskDataProvider getProvider (String name)
```

getProviderById

```
TaskDataProvider getProviderById (Long providerId)
```

getProviders

```
List<TaskDataProvider> getProviders ()
```

registerProvider

void **registerProvider** ([String](#) *json*)

registerProvider

void **registerProvider** ([InputStream](#) *stream*)

11.92.11 TaskFilterExecutor

public class **TaskFilterExecutor**

The `TaskFilterExecutor` applies a list of filters in a `#TaskContext`.

- convertTo** - convert a given value to a correct type,
- getFieldValue** - get value of a field defined in the key from the given object,
- getTriggerKey** - get value of a trigger event parameter,
- checkFilters** - executed defined filters for a task,
- manipulate** - executed the given manipulation on the given string value.

Methods

checkFilters

public boolean **checkFilters** ([List](#)<[Filter](#)> *filters*, [TaskContext](#) *taskContext*)

11.92.12 TaskInitializer

class **TaskInitializer**

The `TaskInitializer` class prepares an action in the task definition to execution.

- evalConfigSteps** - executes all config steps (load data sources, check filters) defined in the task,

See also: [TaskTriggerHandler](#), [TaskActionExecutor](#)

Constructors

TaskInitializer

TaskInitializer ([TaskContext](#) *taskContext*)

Methods

evalConfigSteps

public boolean **evalConfigSteps** ([Map](#)<[String](#), [DataProvider](#)> *dataProviders*)

11.92.13 TaskService

public interface **TaskService**

Methods

deleteTask

void **deleteTask** (Long *taskId*)

exportTask

String **exportTask** (Long *taskId*)

findTasksDependentOnModule

List<Task> **findTasksDependentOnModule** (String *moduleName*)

findTasksForTrigger

List<Task> **findTasksForTrigger** (TriggerEvent *trigger*)

findTasksForTriggerSubject

List<Task> **findTasksForTriggerSubject** (String *subject*)

findTrigger

TriggerEvent **findTrigger** (String *subject*)

getActionEventFor

ActionEvent **getActionEventFor** (TaskActionInformation *taskActionInformation*)

getAllTasks

List<Task> **getAllTasks** ()

getTask

Task **getTask** (Long *taskId*)

importTask

void **importTask** (String *json*)

save

void **save** ([Task](#) *task*)

11.92.14 TaskTriggerHandler

public class **TaskTriggerHandler** implements [TriggerHandler](#)

The `TaskTriggerHandler` receives events and executes tasks for which the trigger event subject is the same as the received event subject.

Constructors

TaskTriggerHandler

public **TaskTriggerHandler** ([TaskService](#) *taskService*, [TaskActivityService](#) *activityService*, [EventListenerRegistryService](#) *registryService*, [EventRelay](#) *eventRelay*, [SettingsFacade](#) *settings*)

Methods

addDataProvider

public void **addDataProvider** ([DataProvider](#) *provider*)

handle

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

registerHandlerFor

public final void **registerHandlerFor** ([String](#) *subject*)

removeDataProvider

public void **removeDataProvider** ([String](#) *taskDataProviderId*)

setBundleContext

public void **setBundleContext** ([BundleContext](#) *bundleContext*)

setDataProviders

void **setDataProviders** ([Map](#)<[String](#), [DataProvider](#)> *dataProviders*)

11.92.15 TriggerHandler

public interface **TriggerHandler**

Standard interface for tasks which executes tasks.

Methods

handle

void **handle** (*MotechEvent event*)

registerHandlerFor

void **registerHandlerFor** (*String subject*)

11.93 org.motechproject.tasks.util

11.93.1 BundleContextUtil

public final class **BundleContextUtil**

Methods

getSymbolicNames

public static *List<String>* **getSymbolicNames** (*BundleContext context*)

11.93.2 ManagementDataProvider

public class **ManagementDataProvider** implements *OsgiServiceLifecycleListener*

Constructors

ManagementDataProvider

public **ManagementDataProvider** (*TaskDataProviderService taskDataProviderService*)

ManagementDataProvider

public **ManagementDataProvider** (*TaskTriggerHandler handler*, *TaskDataProviderService taskDataProviderService*)

Methods

bind

public void **bind** (*Object service*, *Map serviceProperties*)

unbind

public void **unbind** (*Object service*, *Map serviceProperties*)

A

- abandonChanges(Long) (Java method), 302, 315
- AbstractDataProvider (Java class), 88
- AbstractDBConfig (Java class), 111
- AbstractDBConfig(String, String, String, String) (Java constructor), 112
- AbstractObjectValueGenerator (Java class), 268
- AbstractValidator (Java interface), 480
- accept(File) (Java method), 119
- accept(TaskActionInformation) (Java method), 492
- accessdenied(HttpServletRequest) (Java method), 468
- ACTION_PARAMETERS_FIELD (Java field), 538
- ActionEvent (Java class), 492
- ActionEvent() (Java constructor), 492
- ActionEvent(ActionEventRequest) (Java constructor), 492
- ActionEvent(String, String, String, SortedSet) (Java constructor), 492
- ActionEvent(String, String, String, String, SortedSet) (Java constructor), 492
- ActionEvent(String, String, String, String, String, SortedSet) (Java constructor), 492
- ActionEventRequest (Java class), 485
- ActionEventRequest(String, String, String, String, String) (Java constructor), 486
- ActionEventRequest(String, String, String, String, String, SortedSet) (Java constructor), 485
- ActionEventRequestDeserializer (Java class), 537
- ActionNotFoundException (Java class), 536
- ActionNotFoundException(String) (Java constructor), 536
- ActionParameter (Java class), 494
- ActionParameter() (Java constructor), 494
- ActionParameter(ActionParameterRequest) (Java constructor), 494
- ActionParameter(String, String) (Java constructor), 494
- ActionParameter(String, String, boolean) (Java constructor), 494
- ActionParameter(String, String, Integer) (Java constructor), 494
- ActionParameter(String, String, ParameterType) (Java constructor), 494
- ActionParameter(String, String, ParameterType, boolean) (Java constructor), 494
- ActionParameter(String, String, ParameterType, Integer) (Java constructor), 494
- ActionParameter(String, String, ParameterType, Integer, boolean) (Java constructor), 495
- ActionParameterRequest (Java class), 487
- ActionParameterRequest(String, String) (Java constructor), 487
- ActionParameterRequest(String, String, Integer) (Java constructor), 487
- ActionParameterRequest(String, String, Integer, String) (Java constructor), 487
- ActionParameterRequest(String, String, Integer, String, boolean) (Java constructor), 487
- ActionType (Java enum), 82
- ACTIVATE_USER_PERMISSION (Java field), 413
- activateUser(String) (Java method), 443
- Activator (Java class), 355
- ACTIVE (Java field), 450
- activeTasks() (Java method), 541
- ACTIVITY_ACTIVE (Java field), 388
- ACTIVITY_FINISHED (Java field), 388
- ACTIVITY_NOTSTARTED (Java field), 388
- add(AbstractValidator) (Java method), 483
- add(MotechPermission) (Java method), 435
- add(MotechRole) (Java method), 435
- add(MotechUser) (Java method), 437
- add>PasswordRecovery) (Java method), 438
- add(String) (Java method), 119
- add(TaskConfigStep) (Java method), 524
- ADD_NEW_INDEX (Java field), 214
- ADD_USER_PERMISSION (Java field), 413
- addAction(String) (Java method), 240
- addAction(TaskActionInformation) (Java method), 516, 523
- addActionTaskEvent(ActionEvent) (Java method), 496
- addAll(SortedSet) (Java method), 524
- addAngularModule(String) (Java method), 361

addBaseMetadata(JDOMetadata, ClassData, EntityType) (Java method), 159

addBundle(Bundle) (Java method), 358

addClass(ClassData) (Java method), 247

addConfigLocation(String) (Java method), 120

addCriterion(ValidationCriterionDto) (Java method), 227

addData(Map) (Java method), 97

addData(String, Object) (Java method), 97

addDatabaseSuggestion(String) (Java method), 476

addDataProvider(DataProvider) (Java method), 548

addDataSource(DataSource) (Java method), 523

addDataSourceObject(String, Object, boolean) (Java method), 545

addDisplayedField(Number) (Java method), 212

addDisplayedFields(EntityDto, Map) (Java method), 302, 316

addEmptyMetadata() (Java method), 223

addEntityMetadata(JDOMetadata, Entity) (Java method), 159

addError(Task, TaskHandlerException) (Java method), 543

addField(Field) (Java method), 172

addField(Integer) (Java method), 229

addField(Long) (Java method), 229

addField(Number) (Java method), 235, 240

addFields(EntityDto, Collection) (Java method), 302, 316

addFilter(Filter) (Java method), 505

addFilterableField(Number) (Java method), 212

addFilterableFields(EntityDto, Collection) (Java method), 302, 316

addFilterSet(FilterSet) (Java method), 523

addHelperClassMetadata(JDOMetadata, ClassData, Entity, EntityType) (Java method), 159

addI18N(String, String) (Java method), 361

addingService(ServiceReference) (Java method), 357, 359, 372

ADDITIONAL_DATA_PREFIX (Java field), 505

addLookup(Lookup) (Java method), 172

addLookup(Number) (Java method), 235

addLookups(Long, Collection) (Java method), 303, 316

addMetadata(FieldMetadata) (Java method), 182

addMetadata(MetadataDto) (Java method), 223

addNewIndex(String) (Java method), 210

addOpenIdUser(MotechUser) (Java method), 437

addOrUpdate(Channel) (Java method), 540

addOrUpdate(ConfigSettings) (Java method), 289

addOrUpdate(File) (Java method), 124

addOrUpdate(MotechSecurityConfiguration) (Java method), 436

addOrUpdateModuleRecord(ModulePropertiesRecord) (Java method), 124

addOrUpdateModuleRecords(List) (Java method), 125

addOrUpdateProperties(String, String, String, String, Properties, Properties) (Java method), 125

addParameter(ActionParameter, boolean) (Java method), 493

addParameter(ActionParameterRequest, boolean) (Java method), 486

addPermission(PermissionDto) (Java method), 441

addQueueSuggestion(String) (Java method), 476

addSchedulerSuggestion(String) (Java method), 476

addSetting(FieldSetting) (Java method), 183

addSubMenu(String, String) (Java method), 361

addSubMenu(String, String, String) (Java method), 361

addSuccess(Task) (Java method), 544

addTrackerFor(Bundle) (Java method), 360

addTrackerFor(Bundle, ApplicationContext) (Java method), 373

addValidation(FieldValidation) (Java method), 183

addValidationErrors(Set) (Java method), 516

addWarning(Task) (Java method), 544

addWarning(Task, String, String) (Java method), 544

addWarning(Task, String, String, Exception) (Java method), 544

ADMIN_CONFIRM_PASSWORD (Java field), 473

ADMIN_LOGIN (Java field), 473

ADMIN_PASSWORD (Java field), 473

AdminSettings (Java class), 77

AdminSettings(List, boolean) (Java constructor), 77

ADVANCED (Java field), 214

AdvancedSettingsDto (Java class), 210

advancedSettingsDto() (Java method), 172

AFTER (Java field), 512

AFTER_NOW (Java field), 512

afterPropertiesSet() (Java method), 273, 454

ALL (Java field), 200, 261, 510

AllConfigSettings (Java class), 289

AllConfigSettings() (Java constructor), 289

AllEntities (Java class), 289

AllEntities() (Java constructor), 289

AllEntityAudits (Java class), 290

AllEntityAudits() (Java constructor), 290

AllEntityDrafts (Java class), 290

AllEntityDrafts() (Java constructor), 291

AllMotechPermissions (Java class), 435

AllMotechRoles (Java class), 435

AllMotechSecurityRules (Java class), 436

AllMotechUsers (Java class), 437

ALLOW_MULTIPLE_SELECTIONS (Java field), 340

ALLOW_USER_SUPPLIED (Java field), 340

AllPasswordRecoveries (Java class), 438

allRecoveries() (Java method), 438

AllTypes (Java class), 292

AllTypes() (Java constructor), 292

AllTypeSettings (Java class), 291

AllTypeSettings() (Java constructor), 292

AllTypeValidations (Java class), 292

AllTypeValidations() (Java constructor), 292

- AlreadyRegisteredException (Java class), 467
 AlreadyRegisteredException(String) (Java constructor), 467
 AMQ_BROKER_URL (Java field), 108
 AMQ_CONCURRENT_CONSUMERS (Java field), 109
 AMQ_MAX_CONCURRENT_CONSUMERS (Java field), 109
 AMQ_MAX_REDELIVERIES (Java field), 109
 AMQ_QUEUE_EVENTS (Java field), 109
 AMQ_QUEUE_SCHEDULER (Java field), 109
 AMQ_REDELIVERY_DELAY_IN_MILLIS (Java field), 109
 AnnotationFields (Java class), 331
 ANY (Java field), 412
 ANY_PATTERN (Java field), 415
 ApplicationContextServiceReferenceUtils (Java class), 89
 ApplicationContextTracker (Java class), 356
 ApplicationContextTracker(BundleContext) (Java constructor), 356
 ApplicationEnvironment (Java class), 376
 areConfigurationSettingsRegistered() (Java method), 454
 ASC (Java field), 350
 ascOrder(String) (Java method), 285
 asDeclareParameter(int) (Java method), 282
 asFieldMapById(Collection) (Java method), 217
 asFieldMapByName(Collection) (Java method), 217
 asFilter(int) (Java method), 282
 asMatchesPattern(String) (Java method), 286
 asPeriod() (Java method), 106
 asProperties() (Java method), 162, 454, 457, 461
 asProperties(Map) (Java method), 92
 assertArgumentNotEmpty(String, String) (Java method), 93
 assertArgumentNotNull(String, Object) (Java method), 93, 404
 authoritiesFor(MotechUser) (Java method), 441
 AuthoritiesService (Java interface), 441
 AUTO_GENERATED (Java field), 340
 AUTO_GENERATED_EDITABLE (Java field), 340
- ## B
- BASE (Java field), 337
 BasePersistenceService (Java class), 324
 BASIC (Java field), 414
 BEFORE (Java field), 512
 BEFORE_NOW (Java field), 512
 bind(Object, Map) (Java method), 550
 BLANK (Java field), 530
 BLOB (Java field), 241
 BLUEPRINT_ENABLED (Java field), 381
 BLUEPRINT_TEMPLATE (Java field), 307
 BLUEPRINT_XML (Java field), 307
 BlueprintActivator (Java class), 356
 BlueprintApplicationContextTracker (Java class), 357
 BlueprintApplicationContextTracker(BundleContext) (Java constructor), 357
 BOOLEAN (Java field), 241, 514
 BOOLEAN_GETTER_PREFIX (Java field), 346
 BootstrapConfig (Java class), 113
 BootstrapConfig(SQLDBConfig, String, ConfigSource, String) (Java constructor), 114
 breakString(String) (Java method), 353
 breakString(String, String[], String[], String[], String) (Java method), 353
 breakStringForList(String) (Java method), 353
 BrowsingSettings (Java class), 166
 BrowsingSettings(Entity) (Java constructor), 167
 BrowsingSettingsDto (Java class), 212
 build() (Java method), 383, 384, 523
 build(ComboBoxHolder) (Java method), 160
 build(Entity) (Java method), 157
 buildDDE(Entity, Bundle) (Java method), 157
 buildFrom(File) (Java method), 121
 buildHistory(Entity) (Java method), 157
 buildHistoryInfrastructure(String) (Java method), 158
 buildInfrastructure(Entity) (Java method), 158
 buildMenu(String) (Java method), 480
 buildSecurityChain(MotechURLSecurityRule, HTTP-Method) (Java method), 411
 buildTrash(Entity) (Java method), 157
 BUNDLE_ADMIN_ROLE (Java field), 415
 BUNDLE_ID (Java field), 109
 BUNDLE_IMPORTS (Java field), 307
 BUNDLE_MANIFESTVERSION (Java field), 335
 BUNDLE_NAME (Java field), 448
 BUNDLE_NAME_SUFFIX (Java field), 335
 BUNDLE_SECTION (Java field), 109
 BUNDLE_SETTINGS_CHANGED_EVENT_SUBJECT (Java field), 109
 BUNDLE_SYMBOLIC_NAME (Java field), 109, 451
 BUNDLE_VERSION (Java field), 451
 BundleContextUtil (Java class), 549
 BundleContextWrapper (Java class), 357
 BundleContextWrapper() (Java constructor), 357
 BundleContextWrapper(BundleContext) (Java constructor), 357
 BundledJspView (Java class), 358
 BundleHeaders (Java class), 381
 BundleHeaders(Bundle) (Java constructor), 382
 BundleHeaders(BundleContext) (Java constructor), 382
 BundleIcon (Java class), 447
 BundleIcon(byte[], String) (Java constructor), 447
 BundleInformation (Java class), 448
 BundleInformation(Bundle) (Java constructor), 448
 BundleLoader (Java interface), 450
 BundleLoadingException (Java class), 451
 BundleLoadingException(String) (Java constructor), 451

BundleLoadingException(String, Throwable) (Java constructor), 451
BundleLoadingException(Throwable) (Java constructor), 451
BundleName (Java class), 376
BundleName(String) (Java constructor), 377
BundleNames (Java class), 332
BundleRegister (Java class), 358
byTask(Long) (Java method), 539

C

callHandler(MotechEvent) (Java method), 150, 151
canHaveQueue(String) (Java method), 95
canLaunchBundles() (Java method), 466
CAPITALIZE (Java field), 511
Cascade (Java annotation), 154
cast(Class, Enumeration) (Java method), 90
CastUtils (Java class), 90
changeLogLevel(String, String) (Java method), 379
changePassword(String, String, String) (Java method), 444
changeRootLogLevel(String) (Java method), 379
Channel (Java class), 496
Channel() (Java constructor), 496
Channel(ChannelRequest) (Java constructor), 496
Channel(String, String, String) (Java constructor), 496
Channel(String, String, String, String, List, List) (Java constructor), 496
ChannelRegisterEvent (Java class), 498
ChannelRegisterEvent(MotechEvent) (Java constructor), 498
ChannelRegisterEvent(String) (Java constructor), 498
ChannelRequest (Java class), 489
ChannelRequest(String, String, String, String, List, List) (Java constructor), 489
ChannelsDataService (Java interface), 539
ChannelService (Java interface), 539
checkFilters(List, TaskContext) (Java method), 546
checkListContainLogger(List, String) (Java method), 360
checkLogXmlConfiguration(Document) (Java method), 360
ClassData (Java class), 167
ClassData(Entity, byte[]) (Java constructor), 168
ClassData(Entity, byte[], boolean) (Java constructor), 168
ClassData(String, byte[]) (Java constructor), 167
ClassData(String, byte[], boolean) (Java constructor), 167
ClassData(String, String, String, byte[]) (Java constructor), 168
ClassData(String, String, String, byte[], boolean) (Java constructor), 168
ClassData(String, String, String, byte[], boolean, EntityType) (Java constructor), 168
ClassData(String, String, String, byte[], EntityType) (Java constructor), 168

classForName(String) (Java method), 270
classForName(String, boolean) (Java method), 270
classForName(String, ClassLoader) (Java method), 270, 271
classForName(String, ClassLoader, boolean) (Java method), 270
ClassName (Java class), 329
cleanupExpiredRecoveries() (Java method), 445
clear() (Java method), 523
clearEnhancedData() (Java method), 266
clearEntities() (Java method), 275, 279
clearListenersForBean(String) (Java method), 148
CollectionProperty (Java class), 280
CollectionProperty(String, Object) (Java constructor), 280
CollectionResultFromLookupExpectedException (Java class), 247
CollectionResultFromLookupExpectedException(String) (Java constructor), 247
COMBOBOX_VALUES (Java field), 340
ComboboxHolder (Java class), 169
ComboboxHolder(Class, FieldDto) (Java constructor), 169
ComboboxHolder(Entity, Field) (Java constructor), 169
ComboboxHolder(EntityDto, FieldDto) (Java constructor), 169
ComboboxHolder(Field) (Java constructor), 169
ComboboxHolder(List, List, String) (Java constructor), 169
commence(HttpServletRequest, HttpServletResponse, AuthenticationException) (Java method), 408, 410
commitChanges(Long) (Java method), 303, 316
commitChanges(Long, String) (Java method), 303, 316
compare(EmailRecord, EmailRecord) (Java method), 140
compareTo(ActionParameter) (Java method), 495
compareTo(ActionParameterRequest) (Java method), 488
compareTo(TaskActivity) (Java method), 521
compareTo(TaskConfigStep) (Java method), 526
compareTo(Time) (Java method), 99
Config (Java class), 333
CONFIG_LOCATION_PROPERTY_KEY (Java field), 118
CONFIG_MODULE_DIR_PREFIX (Java field), 109
CONFIG_SOURCE (Java field), 113
ConfigFileFilter (Java class), 119
ConfigFileMonitor (Java class), 123
ConfigLoader (Java class), 464
ConfigLocation (Java class), 115
ConfigLocation(String) (Java constructor), 115
ConfigLocationFileStore (Java class), 118
ConfigLocationFileStore(PropertiesConfiguration) (Java constructor), 118
ConfigSettings (Java class), 170

- ConfigSettings() (Java constructor), 170
- ConfigSettings(DeleteMode, boolean, int, TimeUnit) (Java constructor), 171
- ConfigSource (Java class), 116
- ConfigurationConstants (Java class), 108
- ConfigurationService (Java interface), 124
- Constants (Java class), 330, 467
- constructEntities(boolean) (Java method), 160
- constructSymbolicName() (Java method), 454
- CONTAINS (Java field), 512
- contains(String) (Java method), 290
- containsAction(TaskActionInformation) (Java method), 496
- containsBean(String) (Java method), 366
- containsBeanDefinition(String) (Java method), 366
- containsDeclaredField(CtClass, String) (Java method), 263
- containsDeclaredMethod(CtClass, String) (Java method), 263
- containsDisplayedField(Long) (Java method), 212
- containsField(CtClass, String) (Java method), 263
- containsField(String) (Java method), 528
- containsFieldId(Number) (Java method), 235
- containsFilterableField(Number) (Java method), 212
- containsLocalBean(String) (Java method), 366
- containsLookupId(Number) (Java method), 235
- containsMethod(CtClass, String) (Java method), 263
- containsOnlyNullValues(Collection) (Java method), 282
- containsParameter(String) (Java method), 493, 531, 535
- containsProviderObject(String) (Java method), 527
- containsProviderObjectField(String, String) (Java method), 527
- containsProviderObjectLookup(String, String) (Java method), 527
- containsTrigger(TaskTriggerInformation) (Java method), 496
- CONTEXT_PATH (Java field), 381
- CONTEXT_SERVICE_NAME (Java field), 357
- contextInvalidOrProcessed(ServiceReference, ApplicationContext) (Java method), 356
- copy() (Java method), 183, 188, 190, 191, 201, 203, 238
- copy(List) (Java method), 193
- copy(String, Map) (Java method), 146
- copyPropertiesFromTransient(Object, Object) (Java method), 351
- CORE_SETTINGS_CACHE_NAME (Java field), 120
- CoreConfigurationService (Java interface), 120
- count() (Java method), 299, 311
- count(Class, String, Map) (Java method), 309, 320
- count(InstanceSecurityRestriction) (Java method), 293
- count(List) (Java method), 299, 313
- count(List, InstanceSecurityRestriction) (Java method), 293
- count(String, String, Map) (Java method), 309, 321
- count(String[], Object[], InstanceSecurityRestriction) (Java method), 293
- countAll(Class) (Java method), 309, 321
- countAll(String) (Java method), 309, 321
- countEmailRecords(EmailRecordSearchCriteria) (Java method), 142
- countFind(String, String, String, String, Range, Set) (Java method), 143
- countForFilter(Filter) (Java method), 293, 299, 311
- countForFilter(Filter, InstanceSecurityRestriction) (Java method), 294
- countHistoryRecords(Object) (Java method), 305, 325
- countTrashRecords(String) (Java method), 313, 327
- CREATE (Java field), 200
- create(Class, Object, EntityType, ValueGetter) (Java method), 324
- create(Class, Object, EntityType, ValueGetter, ObjectReference) (Java method), 324
- create(Entity, String) (Java method), 291
- create(EntityDto) (Java method), 290
- create(Field, Object) (Java method), 283
- create(InputStream) (Java method), 297, 298
- create(String, Object) (Java method), 283
- create(String, Object, String) (Java method), 283
- create(T) (Java method), 294, 299, 311
- createApplicationContext(BundleContext) (Java method), 87
- createAudit(Entity, String) (Java method), 290
- createEntity(EntityDto) (Java method), 303, 316
- createEnumInitializer(String, String) (Java method), 262
- createField(CtClass, CtClass, String, String) (Java method), 262
- createGetter(String, CtClass, CtField) (Java method), 262
- createInitializer(String, String) (Java method), 262
- createListInitializer(String, Object) (Java method), 262
- createLocaleInitializer(String) (Java method), 262
- createLoggerProperties(List) (Java method), 360
- createParameters(TaskActionInformation, ActionEvent) (Java method), 543
- createRecovery(String, String, String, DateTime, Locale) (Java method), 438
- createRole(RoleDto) (Java method), 442
- createSetter(String, CtField) (Java method), 262
- createSimpleInitializer(String, Object) (Java method), 262
- createSimpleInitializer(String, String) (Java method), 262
- createZipWithConfigFiles(String, String) (Java method), 125
- CREATION_DATE_DISPLAY_FIELD_NAME (Java field), 341
- CREATION_DATE_FIELD_NAME (Java field), 341
- CreationDateValueGenerator (Java class), 268
- CREATOR (Java field), 352
- CREATOR_DISPLAY_FIELD_NAME (Java field), 341

CREATOR_FIELD_NAME (Java field), 341
CreatorValueGenerator (Java class), 269
CRITICAL (Java field), 83
critical(String, String) (Java method), 84
critical(String, String, DateTime) (Java method), 84
CronJobExpressionBuilder (Java class), 383
CronJobExpressionBuilder(Time, Integer, Integer) (Java constructor), 383
CronJobId (Java class), 385
CronJobId(MotechEvent) (Java constructor), 385
CronJobId(String, String) (Java constructor), 385
CronJobSimpleExpressionBuilder (Java class), 384
CronJobSimpleExpressionBuilder(Time) (Java constructor), 384
CronSchedulableJob (Java class), 385
CronSchedulableJob(MotechEvent, String) (Java constructor), 385
CronSchedulableJob(MotechEvent, String, Date, Date) (Java constructor), 385
CronSchedulableJob(MotechEvent, String, Date, Date, boolean) (Java constructor), 385
current() (Java method), 95
currentVersion(Class) (Java method), 326
CustomOperatorProperty (Java class), 281
CustomOperatorProperty(String, T, String) (Java constructor), 281

D

DashboardController (Java class), 467
DATA_ACCESS (Java field), 339
DATABASE_COLUMN_NAME (Java field), 335
DATANUCLEUS (Java field), 341
DATANUCLEUS_FILE (Java field), 333
DATANUCLEUS_PROPERTIES (Java field), 307
DataProvider (Java interface), 90
DataProviderDataService (Java interface), 540
DataSource (Java class), 499
DataSource() (Java constructor), 499
DataSource(Long, Long, String, String, List, boolean) (Java constructor), 499
DataSource(String, Long, Long, String, String, List, boolean) (Java constructor), 499
DataSourceObject (Java class), 540
DataSourceObject(String, Object, boolean) (Java constructor), 541
DATE (Java field), 241, 510, 515
DATETIME (Java field), 242, 511
DateTimeSource (Java interface), 105
DateTimeSourceUtil (Java class), 100
DateTimeValueGenerator (Java class), 269
DateUtil (Java class), 101
Day (Java field), 107
DayOfWeek (Java enum), 97
DayOfWeekSchedulableJob (Java class), 386

DayOfWeekSchedulableJob(MotechEvent, LocalDate, LocalDate, List, Time) (Java constructor), 386
DayOfWeekSchedulableJob(MotechEvent, LocalDate, LocalDate, List, Time, boolean) (Java constructor), 386
DAYS (Java field), 166
daysPast(LocalDate, DayOfWeek) (Java method), 101
daysStarting(DayOfWeek, int) (Java method), 101
daysToCalendarWeekEnd(LocalDate, int) (Java method), 101
DB_ERROR (Java field), 465
DBConfig (Java class), 117
DBConfig(String, String, String) (Java constructor), 117
DEBUG (Java field), 83
debug(String, String) (Java method), 84
debug(String, String, DateTime) (Java method), 84
DEFAULT_ACTIVEMQ_URL (Java field), 480
DEFAULT_DELETE_MODE (Java field), 162
DEFAULT_EMPTY_TRASH (Java field), 163
DEFAULT_TENANT_ID (Java field), 113
DEFAULT_TIME_UNIT (Java field), 163
DEFAULT_TIME_VALUE (Java field), 163
DEFAULT_WAIT_TIME (Java field), 322
DefaultDateTimeSource (Java class), 105
DefaultDateTimeSource() (Java constructor), 105
DefaultMotechDataService (Java class), 299
defineClass(String, byte[]) (Java method), 345
DELETE (Java field), 161, 201, 331, 412
delete(EmailRecord) (Java method), 142
delete(Long) (Java method), 290, 297, 298
delete(MotechPermission) (Java method), 435
delete(String) (Java method), 125
delete(String, Object) (Java method), 299, 311
delete(String, Object, InstanceSecurityRestriction) (Java method), 294
delete(String[], Object[], InstanceSecurityRestriction) (Java method), 294
delete(T) (Java method), 294, 299, 311
DELETE_USER_PERMISSION (Java field), 413
deleteActivitiesForTask(Long) (Java method), 544
deleteAll() (Java method), 299, 311
deleteAll(Entity) (Java method), 291
deleteAll(String, Object) (Java method), 294
deleteAll(String, Object, InstanceSecurityRestriction) (Java method), 294
deleteAll(String[], Object[], InstanceSecurityRestriction) (Java method), 294
deleteByBundle(String) (Java method), 126
deleteEntity(Long) (Java method), 303, 316
DeleteMode (Java enum), 161
deletePermission(String) (Java method), 441
deleteRole(RoleDto) (Java method), 442
deleteTask(Long) (Java method), 547
deleteUser(UserDto) (Java method), 444

DeliveryStatus (Java enum), 136
 DEP_WAIT_TIME_ENV (Java field), 86
 DEP_WAIT_TIME_KEY (Java field), 86
 DESC (Java field), 350
 descOrder(String) (Java method), 285
 DESCRIPTION_FIELD (Java field), 538
 deserialize(JsonElement, Type, JsonDeserializationContext) (Java method), 538
 deserialize(JsonParser, DeserializationContext) (Java method), 164, 538, 539
 Deserializer (Java class), 164
 DEVELOPMENT (Java field), 376
 Direction (Java enum), 350
 DISPLAY_NAME (Java field), 214, 331
 DISPLAY_NAME_FIELD (Java field), 538
 doBegin(Object, TransactionDefinition) (Java method), 272
 doCleanupAfterCompletion(Object) (Java method), 272
 doCreateConnection() (Java method), 152
 doInTransaction(TransactionCallback) (Java method), 299, 311
 DOUBLE (Java field), 227, 242, 515
 doWhenClassNotFound(String) (Java method), 265, 344
 DraftData (Java class), 213
 DraftResult (Java class), 216
 DraftResult(boolean, boolean) (Java constructor), 216
 DtoHelper (Java class), 216
 duration() (Java method), 93

E

EDIT_USER_PERMISSION (Java field), 413
 ElementOrder (Java class), 478
 EMAIL (Java field), 83
 EMAIL_ADMIN_ROLE (Java field), 415
 EmailAuditService (Java interface), 142
 EmailExistsException (Java class), 425
 EmailExistsException(String) (Java constructor), 426
 EmailRecord (Java class), 137
 EmailRecord() (Java constructor), 137
 EmailRecord(String, String, String, String, DateTime, DeliveryStatus) (Java constructor), 137
 EmailRecordComparator (Java class), 140
 EmailRecordComparator(Boolean, String) (Java constructor), 140
 EmailRecords (Java class), 140
 EmailRecords() (Java constructor), 141
 EmailRecords(Integer, Integer, Integer, List) (Java constructor), 141
 EmailRecordSearchCriteria (Java class), 131
 EmailRecordService (Java interface), 143
 EmailSenderService (Java interface), 144
 EMPTY_COLLECTION (Java field), 531
 EMPTY_TRASH_JOB (Java field), 333
 emptyTrash() (Java method), 313, 327

EmptyTrashException (Java class), 247
 EmptyTrashException(Throwable) (Java constructor), 247
 encodePassword(String) (Java method), 409
 endOfDay(Date) (Java method), 101
 ENDS_WITH (Java field), 336
 ENDSWITH (Java field), 512
 Entity (Java annotation), 154
 Entity (Java class), 172
 ENTITY (Java field), 337, 338, 341
 Entity() (Java constructor), 172
 Entity(String) (Java constructor), 172
 Entity(String, String, String, SecurityMode) (Java constructor), 172
 Entity(String, String, String, String, SecurityMode, Set) (Java constructor), 172
 ENTITY_LIST_FILE (Java field), 307
 EntityAlreadyExistException (Java class), 247
 EntityAlreadyExistException() (Java constructor), 248
 EntityAudit (Java class), 177
 EntityBuilder (Java interface), 157
 EntityChangedException (Java class), 248
 EntityChangedException() (Java constructor), 248
 EntityCreationException (Java class), 248
 EntityCreationException(String) (Java constructor), 248
 EntityCreationException(Throwable) (Java constructor), 248
 EntityDeletedException (Java class), 248
 EntityDeletedException() (Java constructor), 249
 EntityDraft (Java class), 178
 EntityDraft() (Java constructor), 178
 EntityDto (Java class), 217
 EntityDto() (Java constructor), 217
 EntityDto(Long, String) (Java constructor), 217
 EntityDto(Long, String, SecurityMode, Set) (Java constructor), 217
 EntityDto(Long, String, String, SecurityMode, Set) (Java constructor), 218
 EntityDto(Long, String, String, String, SecurityMode, Set) (Java constructor), 218
 EntityDto(Long, String, String, String, String, SecurityMode, Set) (Java constructor), 218
 EntityDto(Long, String, String, String, String, SecurityMode, Set, String) (Java constructor), 218
 EntityDto(Long, String, String, String, String, SecurityMode, Set, String, boolean) (Java constructor), 218
 EntityDto(String) (Java constructor), 217
 EntityDto(String, SecurityMode, Set) (Java constructor), 217
 EntityDto(String, String, String, String, SecurityMode, Set) (Java constructor), 218
 EntityInfo (Java class), 180
 EntityInfrastructureBuilder (Java interface), 158

- EntityInfrastructureException (Java class), 249
 - EntityInfrastructureException(Throwable) (Java constructor), 249
 - EntityMetadataBuilder (Java interface), 159
 - EntityNotFoundException (Java class), 249
 - EntityNotFoundException() (Java constructor), 249
 - EntityReadOnlyException (Java class), 249
 - EntityReadOnlyException() (Java constructor), 249
 - EntitySchemaMismatchException (Java class), 250
 - EntitySchemaMismatchException() (Java constructor), 250
 - EntityService (Java interface), 302
 - EntityServiceImpl (Java class), 315
 - EntityType (Java enum), 181
 - ENUM_CLASS_NAME (Java field), 335
 - EnumBuilder (Java interface), 160
 - EnumHelper (Java class), 342
 - ENVIRONMENT (Java field), 376
 - EQ (Java field), 336
 - EQ_IGNORE_CASE (Java field), 336
 - EQ_NUMBER (Java field), 513
 - EqualProperty (Java class), 281
 - EqualProperty(String, T) (Java constructor), 281
 - EQUALS (Java field), 512
 - equals(Object) (Java method), 94, 99, 106, 112, 114, 115, 122, 135, 138, 146, 163, 211, 213, 218, 221, 223, 226, 228, 229, 232, 234, 235, 238, 240, 243, 245, 362, 374, 377, 380, 385, 393, 395, 422, 428, 429, 448, 477, 482, 486, 488–491, 493, 495, 496, 499, 501–503, 505, 506, 509, 514, 517, 519, 521, 524, 526–528, 530, 531, 533, 535, 541
 - equalsSubject(String) (Java method), 531
 - ERROR (Java field), 83, 137, 523
 - error(String, String) (Java method), 84
 - error(String, String, DateTime) (Java method), 84
 - ERROR_REQUIRED (Java field), 482
 - errorsFromLastRun(Task) (Java method), 544
 - evalConfigSteps(Map) (Java method), 546
 - evaluate(Object) (Java method), 348
 - evaluateTemplateString(String) (Java method), 542
 - EVENT_RELAY_CLASS_NAME (Java field), 110
 - EVENT_TYPE_KEY_NAME (Java field), 145
 - EventInfo (Java class), 387
 - EventInfo() (Java constructor), 387
 - EventListener (Java interface), 148
 - EventListenerRegistryService (Java interface), 148
 - EventParameter (Java class), 501
 - EventParameter() (Java constructor), 501
 - EventParameter(EventParameterRequest) (Java constructor), 501
 - EventParameter(String, String) (Java constructor), 501
 - EventParameter(String, String, ParameterType) (Java constructor), 501
 - EventParameterRequest (Java class), 490
 - EventParameterRequest(String, String) (Java constructor), 490
 - EventParameterRequest(String, String, String) (Java constructor), 490
 - EventRelay (Java interface), 149
 - EVERYONE (Java field), 352
 - evictMotechCoreSettingsCache() (Java method), 120
 - evictMotechSettingsCache() (Java method), 126
 - execute(JobExecutionContext) (Java method), 322, 403
 - execute(Map) (Java method), 274
 - execute(Map, QueryParams) (Java method), 274
 - execute(Query) (Java method), 289
 - execute(Query, InstanceSecurityRestriction) (Java method), 284
 - execute(Query, Object, InstanceSecurityRestriction) (Java method), 284
 - execute(Task, TaskActionInformation, TaskContext) (Java method), 543
 - executeCount(Map) (Java method), 274
 - executeLookup(String, Map, QueryParams) (Java method), 297, 298
 - executeQuery(QueryExecution) (Java method), 300, 311
 - executeSQLQuery(SqlQueryExecution) (Java method), 300, 311
 - executeWithArray(Query, List) (Java method), 284
 - executeWithArray(Query, Object[], InstanceSecurityRestriction) (Java method), 284
 - executeWithFilter(Query, Filter, InstanceSecurityRestriction) (Java method), 284
 - EXIST (Java field), 513
 - exists(String, Object) (Java method), 294
 - exists(String[], Object[]) (Java method), 294
 - exportTask(Long) (Java method), 547
 - ExtensibleDataObject (Java class), 97
 - EXTRACTION_FAILED (Java field), 452
 - extractJarInformationFromPath() (Java method), 453
- ## F
- FALSE (Java field), 341
 - Field (Java annotation), 154
 - Field (Java class), 182
 - FIELD (Java field), 214
 - Field() (Java constructor), 182
 - Field(Entity, String, String, boolean, boolean, String, String, Set) (Java constructor), 182
 - Field(Entity, String, String, Set) (Java constructor), 182
 - Field(Entity, String, String, Type, boolean, boolean) (Java constructor), 182
 - FIELD_ID (Java field), 214
 - FieldBasicDto (Java class), 221
 - FieldBasicDto() (Java constructor), 221
 - FieldBasicDto(String, String) (Java constructor), 221

- FieldBasicDto(String, String, boolean, Object, String) (Java constructor), 221
- FieldDto (Java class), 223
- FieldDto() (Java constructor), 223
- FieldDto(Long, Long, TypeDto, FieldBasicDto, boolean, FieldValidationDto) (Java constructor), 223
- FieldDto(Long, Long, TypeDto, FieldBasicDto, boolean, List, FieldValidationDto, List, List) (Java constructor), 223
- FieldHelper (Java class), 343
- FieldHolder (Java class), 187
- FieldHolder(Field) (Java constructor), 187
- FieldHolder(FieldDto) (Java constructor), 187
- FieldHolder(List, List) (Java constructor), 187
- FieldInstanceDto (Java class), 225
- FieldInstanceDto() (Java constructor), 226
- FieldInstanceDto(Long, Long, FieldBasicDto) (Java constructor), 226
- FieldMetadata (Java class), 188
- FieldMetadata() (Java constructor), 188
- FieldMetadata(Field, String) (Java constructor), 188
- FieldMetadata(Field, String, String) (Java constructor), 188
- FieldMetadata(MetadataDto) (Java constructor), 188
- FieldNotFoundException (Java class), 250
- FieldNotFoundException() (Java constructor), 250
- FieldParameter (Java class), 502
- FieldParameter() (Java constructor), 502
- FieldParameter(String, String) (Java constructor), 502
- FieldParameter(String, String, ParameterType) (Java constructor), 502
- FieldSetting (Java class), 189
- FieldSetting() (Java constructor), 189
- FieldSetting(Field, TypeSetting) (Java constructor), 189
- FieldSetting(Field, TypeSetting, String) (Java constructor), 190
- FieldUsedInLookupException (Java class), 250
- FieldUsedInLookupException(String, String) (Java constructor), 250
- FieldValidation (Java class), 191
- FieldValidation() (Java constructor), 191
- FieldValidation(Field, TypeValidation) (Java constructor), 191
- FieldValidation(Field, TypeValidation, String, boolean) (Java constructor), 191
- FieldValidationDto (Java class), 227
- FieldValidationDto() (Java constructor), 227
- FieldValidationDto(ValidationCriterionDto) (Java constructor), 227
- FILE (Java field), 117
- FILE_CHANGED_EVENT_SUBJECT (Java field), 110
- FILE_CREATED_EVENT_SUBJECT (Java field), 110
- FILE_DELETED_EVENT_SUBJECT (Java field), 110
- FILE_PATH (Java field), 110
- FileAccessType (Java enum), 116
- fileChanged(FileChangeEvent) (Java method), 123
- fileCreated(FileChangeEvent) (Java method), 123
- fileDeleted(FileChangeEvent) (Java method), 123
- FileSystemAwareUIHttpContext (Java class), 377
- FileSystemAwareUIHttpContext(HttpContext, String) (Java constructor), 377
- Filter (Java class), 259, 503
- Filter() (Java constructor), 260, 503
- Filter(EventParameter, boolean, String, String) (Java constructor), 503
- filter(Filter) (Java method), 300, 311
- filter(Filter, QueryParams) (Java method), 294, 300, 311
- filter(Filter, QueryParams, InstanceSecurityRestriction) (Java method), 295
- Filter(String, FilterType) (Java constructor), 260
- Filter(String, String, ParameterType, boolean, String, String) (Java constructor), 503
- filterForQuery() (Java method), 260
- FilterSet (Java class), 504
- FilterSet() (Java constructor), 504
- FilterSet(List) (Java constructor), 505
- FilterType (Java enum), 261
- find(String, String, String, String, Range, Set, QueryParams) (Java method), 144
- findAllEmailRecords() (Java method), 143
- findAllSecurityRules() (Java method), 443
- findAnnotationOnBean(String, Class) (Java method), 366
- findBundleByName(BundleContext, String) (Java method), 383
- findBundleBySymbolicName(BundleContext, String) (Java method), 383
- findByBundle(String) (Java method), 131
- findByEmail(String) (Java method), 440
- findByExpirationDate(Range) (Java method), 440
- findById(Long) (Java method), 300, 312
- findById(long) (Java method), 143
- findByModule(String) (Java method), 131
- findByModuleAndFileName(String, String) (Java method), 131
- findByModuleName(String) (Java method), 539
- findByName(String) (Java method), 540
- findByOpenId(String) (Java method), 440
- findByOrigin(String) (Java method), 439
- findByPermissionName(String) (Java method), 435, 439
- findByRecipientAddress(String) (Java method), 144
- findByRole(String) (Java method), 437, 440
- findByRoleName(String) (Java method), 436, 439
- findByTimeout(Range) (Java method), 86
- findByUsername(String) (Java method), 437, 440
- findDeclaredField(CtClass, String) (Java method), 263
- findDeclaredMethod(CtClass, String) (Java method), 263
- findEmailRecords(EmailRecordSearchCriteria) (Java method), 143

`findEntityFieldByName(Long, String)` (Java method), 303, 316
`findExistingConfigs()` (Java method), 464
`findField(CtClass, String)` (Java method), 263
`findFieldByName(Long, String)` (Java method), 303, 316
`findFilename(String)` (Java method), 454
`findForToken(String)` (Java method), 438, 440
`findForUser(String)` (Java method), 438, 440
`findMany(Class, String, Map)` (Java method), 309, 321
`findMany(Class, String, Map, QueryParams)` (Java method), 309, 321
`findMany(String, String, Map)` (Java method), 309, 321
`findMany(String, String, Map, QueryParams)` (Java method), 309, 321
`findMdsBundle(BundleContext)` (Java method), 346
`findMdsEntitiesBundle(BundleContext)` (Java method), 346
`findMethod(CtClass, String)` (Java method), 263
`findOne(Class, String, Map)` (Java method), 309, 321
`findOne(String, String, Map)` (Java method), 310, 321
`findService(Class)` (Java method), 328
`findTasksDependentOnModule(String)` (Java method), 547
`findTasksForTrigger(TriggerEvent)` (Java method), 547
`findTasksForTriggerSubject(String)` (Java method), 547
`findTrashById(Object, Object)` (Java method), 314, 327
`findTrashInstanceById(Object, Object)` (Java method), 300, 312
`findTrigger(String)` (Java method), 547
`findType(Class)` (Java method), 315, 323
`findUserByEmail(String)` (Java method), 437
`findUserByOpenId(String)` (Java method), 437
`findValidations(TypeDto, Class)` (Java method), 315, 323
`FIRST_RUN` (Java field), 465
`fixEnhancerIssuesInMetadata(JDOMetadata)` (Java method), 160
`ForgotController` (Java class), 468
`forgotPost(String)` (Java method), 468
`FORMAT` (Java field), 511
`format(Object)` (Java method), 353
`formatDateTime(DateTime)` (Java method), 104
`formatPeriod(Period)` (Java method), 104
`FRACTION` (Java field), 331
`Friday` (Java field), 98
`fromAdditionalData()` (Java method), 506
`fromHeaders(MessageHeaders, Message)` (Java method), 153
`fromTrigger()` (Java method), 506

G
`ge(Time)` (Java method), 99
`generate()` (Java method), 308, 319
`generate(ExecutionContext, Object, ExtensionMetadata[])` (Java method), 268
`GENERATED_FIELD_NAMES` (Java field), 341
`generateDataProvider()` (Java method), 161
`generateDeclareParameter(int)` (Java method), 280, 283, 287, 288
`generateDummyEntities(int, boolean)` (Java method), 276, 279
`generateDummyEntities(int, int, int, boolean)` (Java method), 276, 279
`generateDummyInstances(Long, int)` (Java method), 276, 279
`generateDummyInstances(Long, int, int, int)` (Java method), 276, 279
`generateFilter(int)` (Java method), 280–283, 287, 288
`generateHeader(Bundle)` (Java method), 478
`generateSchema()` (Java method), 273
`genericSignature(Class, Class)` (Java method), 263
`genericSignature(Class, String)` (Java method), 264
`genericSignature(String, String)` (Java method), 264
`GET` (Java field), 412
`get(Long)` (Java method), 297, 298
`get(Object)` (Java method), 382
`get(QueryParams)` (Java method), 297, 298
`GET_OR_SET_END_INDEX` (Java field), 347
`getAction(TaskActionInformation)` (Java method), 497
`getActionEventFor(TaskActionInformation)` (Java method), 547
`getActionParameters()` (Java method), 486, 493
`getActions()` (Java method), 240, 517
`getActionTaskEvents()` (Java method), 489, 497
`getActionType()` (Java method), 78
`getActiveMessages()` (Java method), 84
`getActivemqConfig()` (Java method), 454
`getActivemqProperties()` (Java method), 457, 461
`getActivity()` (Java method), 389
`getActivityType()` (Java method), 521
`getAdminConfirmPassword()` (Java method), 474
`getAdminEmail()` (Java method), 474
`getAdminLogin()` (Java method), 474
`getAdminPassword()` (Java method), 474
`getAdvancedSettings(Long)` (Java method), 303, 316
`getAdvancedSettings(Long, boolean)` (Java method), 303, 317
`getAfter()` (Java method), 478
`getAfterTimeUnit()` (Java method), 171
`getAfterTimeValue()` (Java method), 171
`getAliases(String)` (Java method), 367
`getAll()` (Java method), 119
`getAllActivities()` (Java method), 544
`getAllChannels()` (Java method), 540
`getAllEntities()` (Java method), 324
`getAllLogMappings()` (Java method), 379
`getAllMessages()` (Java method), 84
`getAllModuleProperties(String, Map)` (Java method), 126
`getAllTasks()` (Java method), 547

[getAllTypes\(\)](#) (Java method), 315, 323
[getAngularModule\(\)](#) (Java method), 448
[getAngularModules\(\)](#) (Java method), 362
[getAngularModulesStr\(\)](#) (Java method), 362
[getAnnotations\(\)](#) (Java method), 209
[getArgs\(\)](#) (Java method), 530, 536
[getAutowireCapableBeanFactory\(\)](#) (Java method), 367
[getAvailableLocales\(HttpServletRequest\)](#) (Java method), 469
[getBasic\(\)](#) (Java method), 223, 226
[getBean\(\)](#) (Java method), 150
[getBean\(Class\)](#) (Java method), 367
[getBean\(String\)](#) (Java method), 367
[getBean\(String, Class\)](#) (Java method), 367
[getBean\(String, Object\)](#) (Java method), 367
[getBeanDefinitionCount\(\)](#) (Java method), 367
[getBeanDefinitionNames\(\)](#) (Java method), 367
[getBeanNamesForType\(Class\)](#) (Java method), 367
[getBeanNamesForType\(Class, boolean, boolean\)](#) (Java method), 367
[getBeansOfType\(Class\)](#) (Java method), 367
[getBeansOfType\(Class, boolean, boolean\)](#) (Java method), 368
[getBeansWithAnnotation\(Class\)](#) (Java method), 368
[getBefore\(\)](#) (Java method), 478
[getBody\(\)](#) (Java method), 88
[getBrowsing\(\)](#) (Java method), 211
[getBrowsingSettings\(\)](#) (Java method), 173
[getBundle\(\)](#) (Java method), 122, 362
[getBundleApplicationContext\(\)](#) (Java method), 358
[getBundleContext\(\)](#) (Java method), 324, 358
[getBundleId\(\)](#) (Java method), 448
[getBundleList\(\)](#) (Java method), 358
[getBundleName\(\)](#) (Java method), 416, 428
[getBundleSymbolicName\(\)](#) (Java method), 452
[getBundleVersion\(\)](#) (Java method), 452
[getBytecode\(\)](#) (Java method), 168
[getChannel\(String\)](#) (Java method), 540
[getChannelIcon\(String\)](#) (Java method), 540
[getChannelModuleName\(\)](#) (Java method), 498
[getChannelName\(\)](#) (Java method), 533
[getClass\(Object, EntityType, BundleContext\)](#) (Java method), 326
[getClass\(String, EntityType, BundleContext\)](#) (Java method), 326
[getClassDefinition\(ClassData\)](#) (Java method), 265
[getClassDefinition\(T\)](#) (Java method), 344
[getClasses\(\)](#) (Java method), 542
[getClassForType\(String\)](#) (Java method), 88
[getClassLoader\(\)](#) (Java method), 368
[getClassName\(\)](#) (Java method), 168, 173, 180, 218
[getClassType\(\)](#) (Java method), 295, 300, 312
[getConfig\(\)](#) (Java method), 470
[getConfigFileChecksum\(\)](#) (Java method), 457, 461
[getConfigLocation\(\)](#) (Java method), 120
[getConfigSource\(\)](#) (Java method), 114, 126
[getConsumerCount\(\)](#) (Java method), 79
[getContentLength\(\)](#) (Java method), 448
[getContext\(\)](#) (Java method), 378
[getContextLocation\(Bundle\)](#) (Java method), 383
[getContextPath\(\)](#) (Java method), 382
[getContextPath\(Bundle\)](#) (Java method), 383
[getCorrectType\(AnnotatedElement\)](#) (Java method), 347
[getCorrectType\(Member\)](#) (Java method), 347
[getCreationDate\(\)](#) (Java method), 197
[getCreator\(\)](#) (Java method), 197
[getCriteria\(\)](#) (Java method), 228
[getCriterion\(String\)](#) (Java method), 228
[getCriticalMessage\(\)](#) (Java method), 362, 370
[getCriticalMessage\(String\)](#) (Java method), 470
[getCronDays\(\)](#) (Java method), 387
[getCronExpression\(\)](#) (Java method), 386
[getCss\(\)](#) (Java method), 479
[getCurrentBundleSymbolicName\(\)](#) (Java method), 358
[getCurrentSchemaVersion\(String\)](#) (Java method), 303, 317, 324
[getCurrentUser\(\)](#) (Java method), 444
[getCustomOperator\(\)](#) (Java method), 232
[getCustomOperators\(\)](#) (Java method), 193
[getData\(\)](#) (Java method), 97
[getDatabaseUrls\(\)](#) (Java method), 476
[getDataNucleusProperties\(\)](#) (Java method), 162
[getDataProviderId\(\)](#) (Java method), 506
[getDataSource\(Long, Long, String\)](#) (Java method), 524
[getDataSourceObjectValue\(String, String, String\)](#) (Java method), 545
[getDataSources\(\)](#) (Java method), 525
[getDataSources\(Long\)](#) (Java method), 525
[getDate\(\)](#) (Java method), 81, 521
[getDateTime\(DateTime\)](#) (Java method), 99
[getDeclaringClass\(AccessibleObject\)](#) (Java method), 347
[getDefault\(\)](#) (Java method), 266
[getDefaultName\(\)](#) (Java method), 205, 243
[getDefaultSecurityConfiguration\(\)](#) (Java method), 441
[getDefaultSettings\(\)](#) (Java method), 466
[getDefaultURL\(\)](#) (Java method), 362
[getDefaultValue\(\)](#) (Java method), 183, 207, 221
[getDeleteMode\(\)](#) (Java method), 163, 165, 171
[getDeliveryStatus\(\)](#) (Java method), 138
[getDeliveryStatuses\(\)](#) (Java method), 132
[getDeliveryTime\(\)](#) (Java method), 138
[getDeliveryTimeRange\(\)](#) (Java method), 132
[getDequeueCount\(\)](#) (Java method), 79
[getDescription\(\)](#) (Java method), 205, 244, 486, 489, 491, 497, 517, 531
[getDestination\(\)](#) (Java method), 79
[getDetachedField\(T, String\)](#) (Java method), 295, 300, 312
[getDetails\(\)](#) (Java method), 190, 191

`getDifferenceOfDatesInYears(Date)` (Java method), 101
`getDirection()` (Java method), 350
`getDisplayedFields()` (Java method), 167, 213
`getDisplayFields(Long)` (Java method), 303, 317
`getDisplayName()` (Java method), 183, 205, 209, 222, 244, 245, 368, 486, 488–491, 497, 503, 510, 514, 528, 532, 533
`getDraftOwnerUsername()` (Java method), 178
`getDrafts()` (Java method), 173
`getDriver()` (Java method), 112
`getEmail()` (Java method), 422, 424, 433
`getEmptyTrash()` (Java method), 171
`getEndDate()` (Java method), 387, 389
`getEndTime()` (Java method), 146, 386, 393
`getEnhancedClassData(String)` (Java method), 266
`getEnhancedClasses(boolean)` (Java method), 266
`getEnqueueCount()` (Java method), 79
`getEntities()` (Java method), 324
`getEntitiesWithLookups()` (Java method), 304, 317
`getEntity()` (Java method), 167, 183, 193, 201, 203
`getEntity(Long)` (Java method), 304, 317, 325
`getEntityByClassName(String)` (Java method), 304, 317
`getEntityDraft(Long)` (Java method), 304, 317
`getEntityDraft(Long, String)` (Java method), 304, 317
`getEntityFields(Long)` (Java method), 304, 317
`getEntityForEdit(Long)` (Java method), 304, 317
`getEntityId()` (Java method), 211, 223
`getEntityLookups(Long)` (Java method), 304, 317
`getEntityName()` (Java method), 180
`getEntityName(String)` (Java method), 329
`getEntityPrefix()` (Java method), 277, 279
`getEntitySchemaVersion(Object)` (Java method), 325
`getEntityTypeSuffix(String)` (Java method), 329
`getEntityVersion()` (Java method), 173
`getEnumName()` (Java method), 170
`getEnvironment()` (Java method), 368, 376
`getEventInfoList()` (Java method), 391
`getEventKey()` (Java method), 490, 501
`getEventParameters()` (Java method), 491, 535
`getExistingConfigFiles()` (Java method), 115
`getExpirationDate()` (Java method), 425
`getExpired()` (Java method), 439
`getExpiredCount()` (Java method), 80
`getExpression()` (Java method), 503
`getExternalId()` (Java method), 422, 424, 433
`getFailureCause()` (Java method), 536
`getField()` (Java method), 188, 190, 191, 260, 350, 509
`getField(Long)` (Java method), 173
`getField(String)` (Java method), 173
`getFieldAndAccessorsForElement(AccessibleObject)` (Java method), 347
`getFieldDtos()` (Java method), 173
`getFieldIds()` (Java method), 236
`getFieldKey()` (Java method), 502
`getFieldName()` (Java method), 349
`getFieldName(AnnotatedElement)` (Java method), 347
`getFieldName(Member)` (Java method), 347
`getFieldNameChanges()` (Java method), 178
`getFieldNameFromGetterSetterName(String)` (Java method), 347
`getFieldPrefix()` (Java method), 277, 279
`getFields()` (Java method), 173, 193, 201, 203, 240, 510, 521, 528
`getFields(Long)` (Java method), 304, 318
`getFieldType(Field, EntityType)` (Java method), 196, 198, 199
`getFile(String, FileAccessType)` (Java method), 115
`getFilename()` (Java method), 122, 452
`getFilePath()` (Java method), 457, 461
`getFilterableFields()` (Java method), 167, 213
`getFilterChainProxy()` (Java method), 442
`getFilters()` (Java method), 505, 525
`getForgotViewData(HttpServletRequest)` (Java method), 469
`getFromAddress()` (Java method), 132, 135, 138
`getGenericSignature(Field, EntityType)` (Java method), 196–199
`getGenericType(AnnotatedElement)` (Java method), 347
`getGenericType(AnnotatedElement, int)` (Java method), 348
`getGenericType(Member, int)` (Java method), 348
`getGetterName(String, CtClass)` (Java method), 262
`getHistoryClassData(String)` (Java method), 266
`getHistoryClassName(String)` (Java method), 329
`getHistoryForInstance(Object, QueryParams)` (Java method), 306, 325
`getHost()` (Java method), 460
`getHour()` (Java method), 99
`getHours()` (Java method), 106
`getHttpContext(HttpContext, Bundle)` (Java method), 378
`getI18n()` (Java method), 362
`getIcon()` (Java method), 448
`getId()` (Java method), 78, 95, 138, 146, 171, 173, 183, 188, 190, 191, 194, 197, 201, 203, 205, 207, 209–211, 218, 224, 226, 229, 232, 234, 236, 244, 368, 419, 430, 517, 527
`getId(T)` (Java method), 300
`getIdentifier()` (Java method), 148, 150
`getIdentity()` (Java method), 90, 94
`getImplementationTitle()` (Java method), 452
`getImplementationVendorID()` (Java method), 452
`getImplementationVersion()` (Java method), 453
`getIndexes()` (Java method), 211
`getInflightCount()` (Java method), 80
`getInfo()` (Java method), 389
`getInfrastructure()` (Java method), 180
`getInstance()` (Java method), 345, 358
`getInstanceClassName(Object)` (Java method), 326

- `getInstanceId()` (Java method), 226
- `getInstanceId(Object)` (Java method), 325
- `getInstancesFromTrash(String, QueryParams)` (Java method), 314, 327
- `getInterfaceName()` (Java method), 180
- `getInterfaceName(String)` (Java method), 266, 329
- `getJarList()` (Java method), 453
- `getJdoMetadata()` (Java method), 292
- `getJobType()` (Java method), 389
- `getJs()` (Java method), 479
- `getKey()` (Java method), 189, 190, 234, 238, 351, 488, 495, 503, 506
- `getKeyType(String)` (Java method), 527, 535
- `getLang()` (Java method), 477
- `getLangLocalisation(HttpServletRequest)` (Java method), 469
- `getLanguage()` (Java method), 457, 461, 474
- `getLastModificationDate()` (Java method), 178
- `getLastRun()` (Java method), 458, 461
- `getLevel()` (Java method), 78, 82
- `getLib()` (Java method), 479
- `getListenerCount(String)` (Java method), 148
- `getListeners(String)` (Java method), 149
- `getLocale()` (Java method), 422, 424, 425, 433
- `getLocale(String)` (Java method), 444
- `getLocation()` (Java method), 115, 449
- `getLock()` (Java method), 356
- `getLoggers()` (Java method), 380
- `getLoginMode()` (Java method), 458, 461, 474
- `getLoginModeValue()` (Java method), 461
- `getLoginViewData(HttpServletRequest)` (Java method), 469
- `getLogLevel()` (Java method), 374
- `getLogLevels()` (Java method), 379
- `getLogName()` (Java method), 374
- `getLookup()` (Java method), 499
- `getLookupById(Long)` (Java method), 173
- `getLookupByName(Long, String)` (Java method), 304, 318
- `getLookupByName(String)` (Java method), 173
- `getLookupDtos()` (Java method), 174
- `getLookupFieldById(Long)` (Java method), 194
- `getLookupFieldByName(String)` (Java method), 194
- `getLookupFields()` (Java method), 229, 529
- `getLookupIds()` (Java method), 236
- `getLookupName()` (Java method), 194, 229
- `getLookupPrefix()` (Java method), 277, 279
- `getLookups()` (Java method), 174, 183, 202, 224
- `getManipulations()` (Java method), 506
- `getMappingFieldName()` (Java method), 349
- `getMax()` (Java method), 94
- `getMembers(Class, Predicate)` (Java method), 348
- `getMenu(HttpServletRequest)` (Java method), 470
- `getMessage()` (Java method), 132, 136, 138, 256, 259, 521, 530, 536, 537
- `getMessage(MessageSourceResolvable, Locale)` (Java method), 368
- `getMessage(String, Object[], Locale)` (Java method), 368
- `getMessage(String, Object[], String, Locale)` (Java method), 368
- `getMessageId()` (Java method), 81
- `getMessageKey()` (Java method), 254
- `getMessageMaxRedeliveryCount()` (Java method), 153
- `getMessageRedeliveryCount()` (Java method), 147
- `getMessageRedeliveryDelay()` (Java method), 153
- `getMessages(HttpServletRequest)` (Java method), 466
- `getMetadata()` (Java method), 183, 224
- `getMetadata(String)` (Java method), 183, 187, 224
- `getMetadata(String, String)` (Java method), 187
- `getMetadataById(Long)` (Java method), 183
- `getMethod()` (Java method), 150
- `getMethodName()` (Java method), 194, 229
- `getMethodParameterType(Type, ComboBoxHolder)` (Java method), 326
- `getMethodsRequired()` (Java method), 419, 430
- `getMimeType()` (Java method), 448
- `getMimeType(String)` (Java method), 378
- `getMin()` (Java method), 94
- `getMinute()` (Java method), 99
- `getMinutes()` (Java method), 106
- `getModificationDate()` (Java method), 177, 197
- `getModifiedBy()` (Java method), 197
- `getModule()` (Java method), 122, 168, 174, 180, 219
- `getModuleData(String)` (Java method), 371
- `getModuleDataByAngular(String)` (Java method), 371
- `getModuleDataByBundle(Bundle)` (Java method), 371
- `getModuleId(Bundle)` (Java method), 383
- `getModuleName()` (Java method), 78, 82, 362, 449, 454, 489, 497, 533
- `getModulePath(BundleName)` (Java method), 376
- `getModuleProperties(String, String, Properties)` (Java method), 126
- `getModuleSettings()` (Java method), 165
- `getModuleVersion()` (Java method), 489, 497, 533
- `getMotechEvent()` (Java method), 386, 387, 393, 396
- `getMotechSecurityConfiguration()` (Java method), 436
- `getName()` (Java method), 90, 117, 174, 180, 184, 208, 209, 219, 222, 232, 238, 283, 382, 390, 449, 457, 499, 517, 527
- `getNamespace()` (Java method), 87, 168, 174, 180, 219, 364
- `getNextFireDate()` (Java method), 390
- `getNotificationRules()` (Java method), 85
- `getObject()` (Java method), 86
- `getObjectId()` (Java method), 499, 506, 541
- `getObjects()` (Java method), 527, 543
- `getObjectType()` (Java method), 87, 506

getObjectValue() (Java method), 541
getOpenId() (Java method), 422, 433
getOpenIdUsers() (Java method), 437, 444
getOperator() (Java method), 281, 503
getOptions() (Java method), 238
getOrder() (Java method), 285, 478, 488, 495, 526
getOrigin() (Java method), 419, 430
getOriginalKey() (Java method), 507
getOriginalRoleName() (Java method), 429
getOsgiFrameworkStorage() (Java method), 114
getOwner() (Java method), 197
getOwnerUsername() (Java method), 177
getPackage(String) (Java method), 330
getPackageRoot() (Java method), 88
getPage() (Java method), 141, 285
getPageSize() (Java method), 285
getParameters() (Java method), 147, 387
getParams() (Java method), 254
getParent() (Java method), 368
getParentBeanFactory() (Java method), 368
getParentEntity() (Java method), 179
getParentVersion() (Java method), 179
getPassword() (Java method), 112, 422, 433, 472, 473
getPasswordConfirmation() (Java method), 473
getPath() (Java method), 215, 453, 478
getPattern() (Java method), 419, 431
getPermissionAccess() (Java method), 419, 431
getPermissionName() (Java method), 416, 428
getPermissionNames() (Java method), 417, 429
getPermissions() (Java method), 435, 441
getPersistenceManager() (Java method), 295
getPersistenceManagerFactory() (Java method), 325
getPlatformSettings() (Java method), 127, 454, 461
getPrefix() (Java method), 507
getPrimitive(Class) (Java method), 353
getPriority() (Java method), 419, 431
getProperties() (Java method), 122, 165
getProperties(File) (Java method), 119
getProperties(String) (Java method), 162, 454
getPropertyName() (Java method), 268, 269, 272, 273
getProperty(String) (Java method), 455
getProperty(String, String) (Java method), 455
getPropertyDescriptors(Object) (Java method), 351
getProtocol() (Java method), 419, 431
getProvider(String) (Java method), 545
getProviderById(Long) (Java method), 545
getProviderId() (Java method), 499
getProviderName() (Java method), 458, 461, 475, 499
getProviderObject(String) (Java method), 527
getProviders() (Java method), 545
getProviderUrl() (Java method), 458, 461, 475
getQuartzScheduler() (Java method), 397
getQuartzSchedulerFactoryBean() (Java method), 397
getQueryParams() (Java method), 132
getQueueSize() (Java method), 80
getQueueUrl() (Java method), 475
getQueueUrls() (Java method), 476
getRangeLookupFields() (Java method), 194
getRawConfig(String) (Java method), 455
getRawConfig(String, String, Resource) (Java method), 127
getRecipient() (Java method), 78
getRecords() (Java method), 141
getRedelivered() (Java method), 81
getReference() (Java method), 349
getRegisteredModules() (Java method), 371
getRelatedClass() (Java method), 199
getRelatedClassName(Field, EntityType) (Java method), 199
getRelatedField() (Java method), 199
getRelationshipValue(ObjectReference) (Java method), 328
getRepeatCount() (Java method), 393
getRepeatIntervalInMilliseconds() (Java method), 393
getRepository() (Java method), 181, 300
getRepositoryName(String) (Java method), 266, 330
getResetViewData(HttpServletRequest) (Java method), 471
getResource(String) (Java method), 369, 377, 378
getResource(String, ClassLoader) (Java method), 270
getResourceFileName(Resource) (Java method), 162, 455
getResourcePath() (Java method), 362, 382
getResourceRootDirectoryPath() (Java method), 377
getResources(String) (Java method), 369
getResources(String, ClassLoader) (Java method), 270
getRestId() (Java method), 181
getRestOptions() (Java method), 174, 211
getRole(String) (Java method), 443
getRoleForAccess() (Java method), 362, 370
getRoleName() (Java method), 417, 429
getRoles() (Java method), 422, 424, 433, 436, 443
getRoles(String) (Java method), 444
getRoot() (Java method), 380
getRootLogLevel() (Java method), 379
getRows() (Java method), 141
getRuleById(Long) (Java method), 436
getRules() (Java method), 436
getRulesByOrigin(String) (Java method), 437
getScheduledJobDetailedInfo(JobBasicInfo) (Java method), 399, 404
getScheduledJobsBasicInfo() (Java method), 399, 405
getScheduledJobTimings(String, String, Date, Date) (Java method), 399, 405
getScheduledJobTimingsWithPrefix(String, String, Date, Date) (Java method), 399, 405
getSchedulerUrl() (Java method), 475
getSchedulerUrls() (Java method), 477
getSecurityMembers() (Java method), 174, 219

getSecurityMode() (Java method), 174, 219
 getSecurityRules() (Java method), 418, 430
 getServerHost() (Java method), 458, 462
 getServerUrl() (Java method), 458, 462
 getService(BundleContext, String) (Java method), 277, 279
 getService(Class) (Java method), 358
 getServiceForInterfaceName(BundleContext, String) (Java method), 353
 getServiceImplName(String) (Java method), 266
 getServiceInterface() (Java method), 486, 493, 519
 getServiceMethod() (Java method), 486, 493, 520
 getServiceName() (Java method), 181
 getServiceName(String) (Java method), 330
 getServletConfig() (Java method), 87, 365
 getServletContext() (Java method), 88, 365, 369
 getSetLookupFields() (Java method), 194
 getSetterName(String) (Java method), 262
 getSetting(String) (Java method), 187, 224
 getSetting(String, String) (Java method), 187
 getSettingAsArray(String) (Java method), 188
 getSettingAsBoolean(String) (Java method), 188
 getSettingByName(String) (Java method), 184
 getSettings() (Java method), 184, 205, 224
 getSettingsList() (Java method), 77
 getSettingsURL() (Java method), 362, 449
 getSimpleName(String) (Java method), 330
 getSingleHistoryInstance(Object, Long) (Java method), 306, 326
 getSqlConfig() (Java method), 114
 getSqlProperties(Properties) (Java method), 108
 getSqlQuery() (Java method), 289
 getStackTraceElement() (Java method), 521
 getStandaloneInstance() (Java method), 345
 getStandaloneInstance(ClassLoader) (Java method), 345
 getStartDate() (Java method), 387, 390, 396
 getStartTime() (Java method), 386, 393
 getStartupDate() (Java method), 369
 getStartupFormValidator(StartupForm, MotechUserService) (Java method), 483
 getStartupViewData(HttpServletRequest) (Java method), 471
 getState() (Java method), 449
 getStatus() (Java method), 390
 getStatusMsgTimeout() (Java method), 458, 462
 getSteps() (Java method), 525
 getStringComboboxFields() (Java method), 174
 getStringValue(String) (Java method), 382
 getSubject() (Java method), 132, 136, 138, 147, 388, 486, 491, 532, 533
 getSubMenu() (Java method), 363
 getSuffixedId() (Java method), 95
 getSuperClass() (Java method), 174, 219
 getSupportClasses() (Java method), 89
 getSupportedLanguages() (Java method), 466, 469
 getSupportedSchemes() (Java method), 420, 431
 getSymbolicName() (Java method), 382, 449, 455
 getSymbolicNames(BundleContext) (Java method), 549
 getTask() (Java method), 522, 545
 getTask(Long) (Java method), 547
 getTaskActivities(Long) (Java method), 544
 getTaskConfig() (Java method), 517
 getTaskErrors() (Java method), 537
 getTaskPossibleErrors() (Java method), 516
 getTenantId() (Java method), 114
 GETTER_PREFIX (Java field), 346
 getTestInt() (Java method), 275
 getTestString() (Java method), 275
 getText() (Java method), 82, 136
 getTime() (Java method), 387
 getTime(HttpServletRequest) (Java method), 468
 getTimeout() (Java method), 82
 getTimestamp() (Java method), 81
 getTimeUnit() (Java method), 163, 165
 getTimeValue() (Java method), 163, 165
 getToAddress() (Java method), 133, 136, 139
 getToken() (Java method), 425, 473
 getTooltip() (Java method), 184, 222
 getTotal() (Java method), 141
 getTracking() (Java method), 174, 211
 getTrash() (Java method), 381
 getTrashClassData(String) (Java method), 266
 getTrashClassName(String) (Java method), 330
 getTrigger() (Java method), 517
 getTrigger(TaskTriggerInformation) (Java method), 497
 getTriggerParameters() (Java method), 545
 getTriggerTaskEvents() (Java method), 489, 497
 getTriggerValue(String) (Java method), 545
 getType() (Java method), 169, 184, 224, 232, 238, 245, 260, 488, 491, 500, 503, 514, 529
 getType(String) (Java method), 369
 getType(TypeValidation) (Java method), 315, 323
 getTypeClass() (Java method), 205, 244
 getTypeClassName() (Java method), 206
 getTypeSettingOptions() (Java method), 208
 getUIDisplayPosition() (Java method), 184
 getUploadSize() (Java method), 458, 462
 getUptime() (Java method), 468
 getUrl() (Java method), 112, 363, 370
 getUrlResource() (Java method), 116
 getUseGenericParams() (Java method), 194
 getUser(HttpServletRequest) (Java method), 468, 470
 getUser(String) (Java method), 444
 getUserAccess() (Java method), 420, 431
 getUserByEmail(String) (Java method), 444
 getUserLang(HttpServletRequest) (Java method), 469
 getUserLocale(HttpServletRequest) (Java method), 466
 getUsername() (Java method), 422, 424, 434, 472, 477

getUsername() (Java method), 112, 352, 425
getUserRoles() (Java method), 352
getUsers() (Java method), 438, 444
getValidation() (Java method), 224
getValidationByName(String) (Java method), 184
getValidationErrors() (Java method), 517
getValidations() (Java method), 184, 206
getValidators() (Java method), 483
getValue() (Java method), 189–191, 234, 238, 246, 283, 351, 509
getValue(Field, Object, Object, EntityType, ObjectReference) (Java method), 329
getValue(KeyInformation) (Java method), 542
getValue(String) (Java method), 215
getValueAsString() (Java method), 238
getValues() (Java method), 170, 215, 520
getValueType() (Java method), 208, 210
getVersion() (Java method), 122, 178, 382, 420, 431, 449
getWrapperForPrimitive(Class) (Java method), 353
greaterThanOrEqualTo(DateTime, List) (Java method), 101
GT (Java field), 336, 513
GT_EQ (Java field), 336

H

handle(MotechEvent) (Java method), 148, 150, 548, 549
HandlerPredicates (Java class), 541
handleSecurity(HttpServletRequest, HttpServletResponse) (Java method), 378
HAS_ANY_MDS_ROLE (Java field), 339
HAS_DATA_ACCESS (Java field), 339
HAS_DATA_OR_SCHEMA_ACCESS (Java field), 339
HAS_SCHEMA_ACCESS (Java field), 339
HAS_SETTINGS_ACCESS (Java field), 339
hasActiveAdminUser() (Java method), 444
hashCode() (Java method), 94, 99, 106, 112, 114, 116, 122, 136, 139, 147, 163, 211, 213, 219, 222, 224, 226, 228, 229, 232, 234, 236, 239, 240, 244, 246, 363, 374, 377, 381, 386, 393, 396, 423, 428, 429, 449, 477, 482, 487, 488, 490–493, 495, 497, 500–502, 504, 505, 507, 509, 510, 514, 517, 520, 522, 525–527, 529, 530, 532, 533, 535, 541
hasInterface(CtClass, CtClass) (Java method), 264
hasListener(String) (Java method), 149
hasManipulations() (Java method), 507
hasPermission(String) (Java method), 417
hasPlatformConfigurationFile() (Java method), 116
hasPrimitive(Class) (Java method), 353
hasRegisteredChannel() (Java method), 517
hasRole(String) (Java method), 422, 424
hasService() (Java method), 486, 493, 520
hasSettings() (Java method), 206
hasStatus(int) (Java method), 449

hasSubject() (Java method), 487, 532, 533
hasUser(String) (Java method), 444
hasValidation() (Java method), 206
hasValidationErrors() (Java method), 517
HEAD (Java field), 412
Header (Java class), 478
HeaderOrder (Java class), 479
HISTORY (Java field), 182
HistoryService (Java interface), 305
HistoryServiceImpl (Java class), 325
HistoryTrashClassHelper (Java class), 326
Hour (Java field), 107
HOURS (Java field), 166
HTTP (Java field), 414
HttpContextFactory (Java class), 377
HTTPMethod (Java enum), 412
HTTPS (Java field), 414
HttpServiceTracker (Java class), 359
HttpServiceTracker(BundleContext, Map) (Java constructor), 359
HttpServiceTrackers (Java class), 359

I

ICON_LOCATIONS (Java field), 447
ID_DISPLAY_FIELD_NAME (Java field), 341
ID_FIELD_NAME (Java field), 341
IdentityProvider (Java interface), 90
Ignore (Java annotation), 154
IllegalLookupException (Java class), 250
IllegalLookupException(String) (Java constructor), 250
IllegalLookupReturnTypeException (Java class), 251
IllegalLookupReturnTypeException(String) (Java constructor), 251
IMPLEMENTATION_TITLE (Java field), 452
IMPLEMENTATION_VENDOR_ID (Java field), 452
IMPLEMENTATION_VERSION (Java field), 452
importTask(String) (Java method), 547
incrementMessageRedeliveryCount() (Java method), 147
inDays() (Java method), 106
index(HttpServletRequest) (Java method), 468
indexOfDisplayedField(Long) (Java method), 213
INFO (Java field), 83
info(String, String) (Java method), 85
info(String, String, DateTime) (Java method), 85
inheritsFromCustomClass(Class) (Java method), 264
init() (Java method), 123, 298, 327, 397
initFrameworkServlet() (Java method), 366
initHandler() (Java method), 453
initializeProxyChain() (Java method), 442
initializeSecurityState() (Java method), 300
inMillis() (Java method), 107
inMinutes() (Java method), 107
inRange(DateTime, DateTime, DateTime) (Java method), 101

- insertField(Integer, Integer) (Java method), 230
- insertField(Integer, Long) (Java method), 230
- InSet (Java annotation), 155
- INSTALLED (Java field), 450
- InstanceSecurityRestriction (Java class), 343
- INTEGER (Java field), 227, 242, 331, 515
- InvalidTokenException (Java class), 426
- InvalidTokenException() (Java constructor), 426
- InvalidTokenException(String) (Java constructor), 426
- IS_END_INDEX (Java field), 347
- isAbstractClass() (Java method), 175, 219
- isActive() (Java method), 420, 423, 424, 431, 434
- isActualEntity() (Java method), 175, 178, 179
- isAllowCreate() (Java method), 202, 204, 240
- isAllowDelete() (Java method), 202, 204, 240
- isAllowMultipleSelections() (Java method), 170
- isAllowRead() (Java method), 202, 204, 240
- isAllowUpdate() (Java method), 202, 204, 241
- isAllowUserSupplied() (Java method), 170
- isAssignable(Class, List) (Java method), 89
- isAssignableFrom(Class, String) (Java method), 270
- isAssignableFrom(String, Class) (Java method), 270
- isAssignableFrom(String, String) (Java method), 270
- isBaseEntity() (Java method), 175
- isBefore(Time) (Java method), 99
- isBeingTracked(Bundle) (Java method), 360, 373
- isBlob() (Java method), 206
- isBlueprintEnabled() (Java method), 382
- isBootstrapConfigRequired() (Java method), 466
- isBundleMdsDependent(Bundle) (Java method), 346
- isByCreator() (Java method), 343
- isByOwner() (Java method), 343
- isCascadeDelete() (Java method), 200
- isCascadePersist() (Java method), 200
- isCascadeUpdate() (Java method), 200
- isChangesMade() (Java method), 179, 216
- isClassPersistenceCapable(Class) (Java method), 272
- isCombobox() (Java method), 206, 244
- isConfigRequired() (Java method), 466
- isCreate() (Java method), 215, 236
- isDDE() (Java method), 169, 175, 219
- isDDEReady(String) (Java method), 267
- isDeletable() (Java method), 417, 429
- isDelete() (Java method), 236
- isDeleted() (Java method), 420, 431
- isDraft() (Java method), 175, 179
- isEdit() (Java method), 215
- isEmpty() (Java method), 344
- isEmptyTrash() (Java method), 163, 165
- isEnabled() (Java method), 192, 246, 518
- isEnabled(boolean) (Java method), 524
- isEnum() (Java method), 170
- isEnumList() (Java method), 170
- isExposedViaRest() (Java method), 184, 194, 230
- isFailIfDataNotFound() (Java method), 500
- isFailIfNotFound() (Java method), 541
- isFile() (Java method), 117
- isFileSupported(File) (Java method), 119
- isForAdvanced() (Java method), 215
- isForField() (Java method), 215
- isForSecurity() (Java method), 215
- isFrameworkBundle(Bundle) (Java method), 346
- isGeneratePassword() (Java method), 434
- isGetter(Member) (Java method), 348
- isHistoryClassName(String) (Java method), 330
- isIgnorePastFiresAtStart() (Java method), 386, 387, 393
- isInDevelopmentMode() (Java method), 376
- isInitialized() (Java method), 365
- isInterfaceClass() (Java method), 169
- isLastEvent() (Java method), 147
- isLessThanADay() (Java method), 107
- isManyToMany() (Java method), 200
- isMdsBundle(Bundle) (Java method), 346
- isMdsEntitiesBundle(Bundle) (Java method), 346
- isModified() (Java method), 219
- isModuleRegistered(String) (Java method), 371
- isNeedsAttention() (Java method), 363, 370
- isNegationOperator() (Java method), 504
- isNotValid(ServiceReference) (Java method), 89
- isNullOrEmpty(String) (Java method), 105
- isOneToMany() (Java method), 200
- isOneToOne() (Java method), 200
- isOnOrAfter(DateTime, DateTime) (Java method), 101
- isOnOrBefore(DateTime, DateTime) (Java method), 102
- isOpenId() (Java method), 457
- isOperatorAMethod() (Java method), 281
- isOrderSet() (Java method), 285
- isOutdated() (Java method), 179, 216, 219
- isOwningSide() (Java method), 200
- isPagingSet() (Java method), 286
- isParametrized() (Java method), 543
- isPasswordValid(String, String) (Java method), 409
- isPlatformCoreConfigFile(File) (Java method), 120
- isPlatformInitialized() (Java method), 458, 462
- isPrimitive(Class) (Java method), 353
- isPrimitive(String) (Java method), 354
- isPrototype(String) (Java method), 369
- isRangeParam(Field) (Java method), 194
- isRaw() (Java method), 122
- isRead() (Java method), 236
- isReadOnly() (Java method), 77, 184, 194, 220, 224, 230
- isReferenced() (Java method), 230
- isRelationship() (Java method), 206, 244
- isRemove() (Java method), 215
- isRepository() (Java method), 457
- isRequired() (Java method), 184, 222, 488, 495
- isRest() (Java method), 420, 431
- isSecurityLaunch() (Java method), 477

`isServiceInterfaceRegistered(String)` (Java method), 267
`isSetParam(Field)` (Java method), 195
`isSetter(Member)` (Java method), 348
`isSingleObjectReturn()` (Java method), 195, 230
`isSingleton()` (Java method), 87
`isSingleton(String)` (Java method), 369
`isString()` (Java method), 170
`isStringList()` (Java method), 170
`isSubClassOfMdsEntity()` (Java method), 175
`isTracked()` (Java method), 185
`isTrashClassName(String)` (Java method), 330
`isTrashMode()` (Java method), 314, 327
`isTypeMatch(String, Class)` (Java method), 369
`isUIDisplayable()` (Java method), 185
`isUIFilterable()` (Java method), 185
`isUpdate()` (Java method), 236
`isUseGenericParam()` (Java method), 232
`isUseOriginalFireTimeAfterMisfire()` (Java method), 393
`isValid()` (Java method), 487, 516
`isValid(ServiceReference)` (Java method), 89
`isValid(String)` (Java method), 117

J

`JAR_FILE_EXTENSION` (Java field), 453
`JarGeneratorService` (Java interface), 307
`JarGeneratorServiceImpl` (Java class), 319
`JarInformation` (Java class), 451
`JarInformation(File)` (Java constructor), 452
`JarInformationHandler` (Java class), 453
`JarInformationHandler(String)` (Java constructor), 453
`JavassistBuilder` (Java class), 261
`JavassistHelper` (Java class), 263
`JavassistLoader` (Java class), 265
`JavassistLoader(MDSClassLoader)` (Java constructor), 265
`JOB_GROUP_NAME` (Java field), 322, 404
`JOB_ID_KEY` (Java field), 399
`JobBasicInfo` (Java class), 388
`JobBasicInfo()` (Java constructor), 389
`JobBasicInfo(String, String, String, String, String, String, String, String)` (Java constructor), 389
`JobDetailedInfo` (Java class), 391
`JobDetailedInfo()` (Java constructor), 391
`JobDetailedInfo(List)` (Java constructor), 391
`JobId` (Java class), 391
`JobId(MotechEvent, String)` (Java constructor), 391
`JobId(String, String, String)` (Java constructor), 391
`JOBTYPES_CRON` (Java field), 388
`JOBTYPES_REPEATING` (Java field), 388
`JOBTYPES_RUNONCE` (Java field), 388
`JodaFormatter` (Java class), 104
`JodaFormatter()` (Java constructor), 104
`JOIN` (Java field), 511
`JSON_EXTENSION` (Java field), 110

K

`KeyEvaluator` (Java class), 542
`KeyEvaluator(TaskContext)` (Java constructor), 542
`KeyInformation` (Java class), 505

L

`LANGUAGE` (Java field), 110, 474
`le(Time)` (Java method), 100
`LESS_DAYS_FROM_NOW` (Java field), 513
`LESS_MONTHS_FROM_NOW` (Java field), 513
`lessThan(Date, List)` (Java method), 102
`Level` (Java enum), 83
`LIST` (Java field), 242, 515
`listEntities()` (Java method), 304, 318
`listEntities(boolean)` (Java method), 305, 318
`listRawConfigNames(String)` (Java method), 127
`listWorkInProgress()` (Java method), 305, 318
`load(DigestInputStream)` (Java method), 458, 462
`loadBootstrapConfig()` (Java method), 120, 127
`loadBundle(Bundle)` (Java method), 360, 451
`loadClass(Bundle, String, ClassPool)` (Java method), 264
`loadClass(ClassData)` (Java method), 265
`loadClass(T)` (Java method), 344
`loadConfig()` (Java method), 128
`loadDefaultConfig()` (Java method), 128, 464
`Loader` (Java class), 344
`LoaderException` (Java class), 251
`LoaderException(Throwable)` (Java constructor), 251
`loadLoggerDbConfiguration()` (Java method), 360
`loadMotechSettings()` (Java method), 464
`loadRoles(ApplicationContext)` (Java method), 446
`loadRules(ApplicationContext)` (Java method), 446
`LOCAL_DATE` (Java field), 242
`LocaleController` (Java class), 469
`LocaleService` (Java interface), 466
`log(EmailRecord)` (Java method), 143
`Log4JBundleLoader` (Java class), 360
`logError(String)` (Java method), 93
`logError(String, Exception)` (Java method), 93
`logError(String, Object)` (Java method), 93
`logger()` (Java method), 93
`Loggers` (Java class), 380
`Loggers()` (Java constructor), 380
`Loggers(List, LogMapping)` (Java constructor), 380
`login(HttpServletRequest)` (Java method), 469, 470
`LOGIN_MODE` (Java field), 474
`LoginController` (Java class), 469
`logInfo(String, Object)` (Java method), 93
`LoginForm` (Java class), 472
`LoginMode` (Java class), 456
`LOGINMODE` (Java field), 110
`LogMapping` (Java class), 374
`LogMapping()` (Java constructor), 374
`LogMapping(String, String)` (Java constructor), 374

- logObjectIfNotNull(Object) (Java method), 405
 - logout(HttpServletRequest, HttpServletResponse, Authentication) (Java method), 409
 - LONG (Java field), 242, 515
 - Lookup (Java annotation), 155
 - Lookup (Java class), 192, 508
 - Lookup() (Java constructor), 192, 508
 - Lookup(LookupDto, List) (Java constructor), 193
 - Lookup(String, boolean, boolean, List) (Java constructor), 192
 - Lookup(String, boolean, boolean, List, boolean, String) (Java constructor), 193
 - Lookup(String, boolean, boolean, List, boolean, String, List, List, Map) (Java constructor), 193
 - Lookup(String, boolean, boolean, List, boolean, String, List, List, Map, Map) (Java constructor), 193
 - Lookup(String, boolean, boolean, List, Entity) (Java constructor), 193
 - Lookup(String, String) (Java constructor), 508
 - lookup(String, String, Map) (Java method), 90
 - lookupCountMethod(String) (Java method), 344
 - LookupDto (Java class), 228
 - LookupDto() (Java constructor), 228
 - LookupDto(Long, String, boolean, boolean, List, boolean, String) (Java constructor), 229
 - LookupDto(String, boolean, boolean) (Java constructor), 228
 - LookupDto(String, boolean, boolean, List, boolean) (Java constructor), 228
 - LookupDto(String, boolean, boolean, List, boolean, String) (Java constructor), 229
 - LookupExecutionException (Java class), 251
 - LookupExecutionException() (Java constructor), 251
 - LookupExecutionException(Throwable) (Java constructor), 251
 - LookupExecutor (Java class), 274
 - LookupExecutor(MotechDataService, LookupDto, Map) (Java constructor), 274
 - LookupExecutorException (Java class), 251
 - LookupExecutorException(String, Throwable) (Java constructor), 252
 - LookupField (Java annotation), 155
 - LookupFieldDto (Java class), 231
 - LookupFieldDto() (Java constructor), 231
 - LookupFieldDto(Long, String, Type) (Java constructor), 231
 - LookupFieldDto(Long, String, Type, String) (Java constructor), 231
 - LookupFieldDto(Long, String, Type, String, boolean) (Java constructor), 232
 - LookupFieldsParameter (Java class), 509
 - LookupFieldsParameter() (Java constructor), 509
 - LookupFieldsParameter(String, List) (Java constructor), 509
 - lookupMethod(String) (Java method), 345
 - LookupName (Java class), 344
 - LookupNameIsRepeatedException (Java class), 252
 - LookupNameIsRepeatedException() (Java constructor), 252
 - LookupNotFoundException (Java class), 252
 - LookupNotFoundException() (Java constructor), 252
 - LookupReferencedException (Java class), 252
 - LookupReferencedException(String) (Java constructor), 252
 - LookupWrongParameterTypeException (Java class), 252
 - LookupWrongParameterTypeException(String) (Java constructor), 253
 - LT (Java field), 336, 513
 - LT_EQ (Java field), 337
- ## M
- Mail (Java class), 135
 - Mail(String, String, String, String) (Java constructor), 135
 - main(String[]) (Java method), 403
 - makeDummyInstance(Long) (Java method), 277, 279
 - MANAGE_PERMISSION_PERMISSION (Java field), 413
 - MANAGE_ROLE_PERMISSION (Java field), 413
 - MANAGE_USER_PERMISSION (Java field), 413
 - ManagementDataProvider (Java class), 549
 - ManagementDataProvider(TaskDataProviderService) (Java constructor), 549
 - ManagementDataProvider(TaskTriggerHandler, TaskDataProviderService) (Java constructor), 549
 - Manifest (Java class), 334
 - MANIFEST_VERSION (Java field), 335
 - manipulate(String, String) (Java method), 542
 - ManipulationTarget (Java enum), 510
 - ManipulationType (Java enum), 511
 - MANY_TO_MANY_RELATIONSHIP (Java field), 242
 - MANY_TO_ONE_RELATIONSHIP (Java field), 242
 - ManyToManyRelationship (Java class), 196
 - ManyToOneRelationship (Java class), 196
 - MAP (Java field), 242, 515
 - MAP_KEY_TYPE (Java field), 335
 - MAP_VALUE_TYPE (Java field), 335
 - markAsProcessed(ApplicationContext) (Java method), 356
 - MATCHES (Java field), 337
 - matches(StatusMessage) (Java method), 78
 - MatchesProperty (Java class), 282
 - MatchesProperty(String, String) (Java constructor), 282
 - MAX (Java field), 331
 - MAX_REPEAT_COUNT (Java field), 322
 - MDS_ADMIN (Java field), 415
 - MDS_BUNDLE_NAME (Java field), 332
 - MDS_BUNDLE_SYMBOLIC_NAME (Java field), 332
 - MDS_COMMON_CONTEXT (Java field), 307

- MDS_DELETE_MODE (Java field), 333
- MDS_EMPTY_TRASH (Java field), 334
- MDS_ENTITIES_CONTEXT (Java field), 307
- MDS_ENTITIES_CONTEXT_TEMPLATE (Java field), 307
- MDS_ENTITIES_NAME (Java field), 333
- MDS_ENTITIES_SYMBOLIC_NAME (Java field), 333
- MDS_MIGRATION_NAME (Java field), 333
- MDS_MIGRATION_SYMBOLIC_NAME (Java field), 333
- MDS_TIME_UNIT (Java field), 334
- MDS_TIME_VALUE (Java field), 334
- MdsBundleHelper (Java class), 346
- MDSClassLoader (Java class), 345
- MDSClassLoader() (Java constructor), 345
- MDSClassLoader(ClassLoader) (Java constructor), 345
- MDSClassLoaderResolver (Java class), 269
- MDSClassLoaderResolver() (Java constructor), 269
- MDSClassLoaderResolver(ClassLoader) (Java constructor), 269
- MDSClassLoaderResolverImpl (Java class), 271
- MDSClassLoaderResolverImpl() (Java constructor), 271
- MDSClassLoaderResolverImpl(ClassLoader) (Java constructor), 271
- MdsConfig (Java class), 162
- MdsConfig() (Java constructor), 162
- MDSConstructor (Java interface), 160
- MDSDataProviderBuilder (Java interface), 161
- MdsDummyDataGenerator (Java interface), 275
- MdsDummyDataGeneratorImpl (Java class), 278
- MdsDummyDataGeneratorImpl(EntityService, JarGeneratorService, BundleContext) (Java constructor), 278
- MdsEntity (Java class), 197
- MdsException (Java class), 253
- MdsException(String) (Java constructor), 253
- MdsException(String, String) (Java constructor), 253
- MdsException(String, String, Throwable) (Java constructor), 254
- MdsException(String, Throwable) (Java constructor), 253
- MdsJdoAnnotationReader (Java class), 271
- MdsJdoAnnotationReader(MetaDataManager) (Java constructor), 271
- MdsJDOEnhancer (Java class), 246
- MdsJDOEnhancer(Properties, ClassLoader) (Java constructor), 247
- MDSLookupService (Java interface), 309
- MdsLookupServiceImpl (Java class), 320
- MdsRestFacade (Java interface), 296
- MdsRestFacadeImpl (Java class), 297
- MdsScheduledJob (Java class), 322
- MdsSchedulerException (Java class), 254
- MdsSchedulerException(String, Throwable) (Java constructor), 254
- MdsSchedulerService (Java interface), 310
- MdsSchedulerServiceImpl (Java class), 322
- MdsSchedulerServiceImpl(BundleContext) (Java constructor), 323
- MdsTransactionManager (Java class), 272
- MemberUtil (Java class), 346
- MenuBuilder (Java class), 479
- mergeMaps(Map, Map) (Java method), 92
- mergeWithDefaults(Properties) (Java method), 462
- MetadataDto (Java class), 233
- MetadataDto() (Java constructor), 233
- MetadataDto(Long, String, String) (Java constructor), 234
- MetadataDto(String, String) (Java constructor), 234
- MetadataHolder (Java class), 292
- MetadataKeys (Java class), 335
- MethodHandler (Java class), 542
- MethodHandler(ActionEvent, Map) (Java constructor), 542
- MIN (Java field), 331
- Minute (Java field), 107
- MODIFICATION_DATE_DISPLAY_FIELD_NAME (Java field), 342
- MODIFICATION_DATE_FIELD_NAME (Java field), 342
- ModificationDateValueGenerator (Java class), 272
- MODIFIED_BY_DISPLAY_FIELD_NAME (Java field), 342
- MODIFIED_BY_FIELD_NAME (Java field), 342
- ModifiedByValueGenerator (Java class), 272
- modify(DateTime) (Java method), 269
- modify(String) (Java method), 274
- modify(T) (Java method), 268
- MODULE (Java field), 331
- MODULE_FILE (Java field), 334
- moduleBackToNormal(String) (Java method), 372
- moduleBackToNormal(String, String) (Java method), 372
- ModuleController (Java class), 470
- moduleId() (Java method), 373
- moduleNameForRest(String) (Java method), 330
- moduleNeedsAttention(String, String) (Java method), 372
- moduleNeedsAttention(String, String, String) (Java method), 372
- ModulePropertiesRecord (Java class), 121
- ModulePropertiesRecord() (Java constructor), 121
- ModulePropertiesRecord(Map, String, String, String, String, boolean) (Java constructor), 121
- ModulePropertiesRecord(Properties, String, String, String, String, boolean) (Java constructor), 121
- ModulePropertiesService (Java interface), 131
- ModuleRegistrationData (Java class), 361
- ModuleRegistrationData() (Java constructor), 361

- ModuleRegistrationData(String, Map) (Java constructor), 361
- ModuleRegistrationData(String, String) (Java constructor), 361
- ModuleRegistrationData(String, String, List, Map) (Java constructor), 361
- MODULES_WITH_SUBMENU (Java field), 371
- MODULES_WITHOUT_SUBMENU (Java field), 371
- MODULES_WITHOUT_UI (Java field), 371
- ModuleSettings (Java class), 162
- Monday (Java field), 98
- MONTHS (Java field), 166
- MORE_DAYS_FROM_NOW (Java field), 513
- MORE_MONTHS_FROM_NOW (Java field), 513
- MOTech_EVENT (Java field), 152
- MOTech_EVENT_CLASS_NAME (Java field), 110
- MOTech_MDS_PROPERTIES (Java field), 307
- MotechAccessVoter (Java class), 407
- MotechBasicAuthenticationEntryPoint (Java class), 408
- MotechBasicAuthenticationEntryPoint(SettingsFacade) (Java constructor), 408
- MotechCachingConnectionFactory (Java class), 152
- MotechClassPool (Java class), 266
- MotechConfigurationException (Java class), 108
- MotechConfigurationException(String) (Java constructor), 108
- MotechConfigurationException(String, Exception) (Java constructor), 108
- MotechDataRepository (Java class), 293
- MotechDataRepository(Class) (Java constructor), 293
- MotechDataService (Java interface), 310
- MotechEnumUtils (Java class), 90
- MotechEvent (Java class), 145
- MotechEvent() (Java constructor), 145
- MotechEvent(String) (Java constructor), 146
- MotechEvent(String, Map) (Java constructor), 146
- MotechEventConfig (Java class), 153
- MotechEventHeaderMapper (Java class), 153
- MotechEventTransformer (Java class), 153
- MotechException (Java class), 91
- MotechException(String) (Java constructor), 92
- MotechException(String, Throwable) (Java constructor), 92
- MotechExtenderConfigFactory (Java class), 86
- MotechJsonReader (Java class), 96
- MotechJsonReader() (Java constructor), 96
- MotechJsonReader(FieldNamingStrategy) (Java constructor), 96
- MotechListener (Java annotation), 149
- MotechListenerAbstractProxy (Java class), 150
- MotechListenerAbstractProxy(String, Object, Method) (Java constructor), 150
- MotechListenerEventProxy (Java class), 151
- MotechListenerEventProxy(String, Object, Method) (Java constructor), 151
- MotechListenerNamedParametersProxy (Java class), 151
- MotechListenerNamedParametersProxy(String, Object, Method) (Java constructor), 151
- MotechListenerType (Java enum), 152
- MotechLoginUrlAuthenticationEntryPoint (Java class), 408
- MotechLogoutSuccessHandler (Java class), 409
- MotechMapUtils (Java class), 92
- MotechObject (Java class), 92
- MotechOsgiApplicationContextCreator (Java class), 87
- MotechOsgiConfigurableApplicationContext (Java class), 87
- MotechOsgiConfigurableApplicationContext(String[]) (Java constructor), 87
- MotechOsgiWebApplicationContext (Java class), 364
- MotechOsgiWebApplicationContext() (Java constructor), 364
- MotechParam (Java annotation), 152
- MotechPasswordEncoder (Java class), 409
- MotechPermission (Java class), 416
- MotechPermission() (Java constructor), 416
- MotechPermission(String, String) (Java constructor), 416
- MotechPermissionsDataService (Java interface), 439
- MotechPermissionService (Java interface), 441
- MotechPlatformState (Java enum), 465
- MotechProperties (Java class), 97
- MotechProxyManager (Java class), 441
- MotechRestBasicAuthenticationEntryPoint (Java class), 409
- MotechRestBasicAuthenticationEntryPoint(SettingsFacade) (Java constructor), 409
- MotechRole (Java class), 417
- MotechRole() (Java constructor), 417
- MotechRole(String, List, boolean) (Java constructor), 417
- MotechRolesDataService (Java interface), 439
- MotechRoleService (Java interface), 442
- MotechScheduledJob (Java class), 402
- MotechScheduler (Java class), 403
- MotechSchedulerActionProxyService (Java interface), 398
- MotechSchedulerActionProxyServiceImpl (Java class), 403
- MotechSchedulerActionProxyServiceImpl(MotechSchedulerService) (Java constructor), 403
- MotechSchedulerException (Java class), 396
- MotechSchedulerException() (Java constructor), 396
- MotechSchedulerException(String) (Java constructor), 396
- MotechSchedulerException(String, Throwable) (Java constructor), 396
- MotechSchedulerException(Throwable) (Java constructor), 396

MotechSchedulerFactoryBean (Java class), 397
MotechSchedulerFactoryBean(ApplicationContext, Properties) (Java constructor), 397
MotechSchedulerService (Java interface), 398
MotechSchedulerServiceImpl (Java class), 404
MotechSchedulerServiceImpl(MotechSchedulerFactoryBean, SettingsFacade) (Java constructor), 404
MotechSecurityConfiguration (Java class), 418
MotechSecurityConfiguration() (Java constructor), 418
MotechSecurityConfiguration(List) (Java constructor), 418
MotechSettings (Java interface), 457
MotechSuccessHandler (Java class), 410
MotechURL (Java class), 460
MotechURL(String) (Java constructor), 460
MotechURLSecurityRule (Java class), 418
MotechURLSecurityRuleDataService (Java interface), 439
MotechURLSecurityService (Java interface), 443
MotechUser (Java class), 421
MotechUser() (Java constructor), 421
MotechUser(String, String, String, String, List, String, Locale) (Java constructor), 422
MotechUserProfile (Java class), 424
MotechUserProfile(MotechUser) (Java constructor), 424
MotechUsersDataService (Java interface), 440
MotechUserService (Java interface), 443
moveFromTrash(Object, Object) (Java method), 314, 327
moveToTrash(Object, Long) (Java method), 314, 328

N

NAME (Java field), 214, 332
NAMED_PARAMETERS (Java field), 152
NAMESPACE (Java field), 332
NanoStopWatch (Java class), 93
NanoStopWatch() (Java constructor), 93
NEED_BOOTSTRAP_CONFIG (Java field), 465
NEED_CONFIG (Java field), 465
NEQ (Java field), 337
newDate(Date) (Java method), 102
newDate(DateTime) (Java method), 102
newDate(int, int, int) (Java method), 102
newDateTime(Date) (Java method), 102
newDateTime(int, int, int) (Java method), 102
newDateTime(int, int, int, int, int, int) (Java method), 103
newDateTime(int, int, int, Time) (Java method), 102
newDateTime(LocalDate) (Java method), 102
newDateTime(LocalDate, int, int, int) (Java method), 102
newDateTime(LocalDate, Time) (Java method), 102
nextApplicableWeekDay(DateTime, List) (Java method), 103
nextApplicableWeekDayIncludingFromDate(DateTime, List) (Java method), 103
NO (Java field), 261

NO_DB (Java field), 465
NO_METHODS_REQUIRED_EXCEPTION_MESSAGE (Java field), 410
NO_PATTERN_EXCEPTION_MESSAGE (Java field), 410
NO_PROTOCOL_EXCEPTION_MESSAGE (Java field), 410
NO_SECURITY (Java field), 414
NO_SUPPORTED_SCHEMES_EXCEPTION_MESSAGE (Java field), 411
NonAdminUserException (Java class), 426
NonAdminUserException(String) (Java constructor), 426
NORMAL_RUN (Java field), 465
NoSuchTypeException (Java class), 254
NoSuchTypeException() (Java constructor), 255
NotificationRule (Java class), 78
NotificationRule() (Java constructor), 78
NotificationRule(String, ActionType, Level, String) (Java constructor), 78
NotificationRulesDataService (Java interface), 84
NotInSet (Java annotation), 155
now() (Java method), 100, 103, 105
nowUTC() (Java method), 103
NULL (Java field), 531
NumberPredicate (Java class), 348
NumberPredicate(Number) (Java constructor), 348

O

OATH (Java field), 414
objectEquals(Long, Long, String) (Java method), 500
ObjectNotFoundException (Java class), 255
ObjectNotFoundException() (Java constructor), 255
ObjectReadException (Java class), 255
ObjectReadException() (Java constructor), 255
ObjectReadException(Throwable) (Java constructor), 255
ObjectReference (Java class), 348
ObjectReference(String, Object, String) (Java constructor), 349
ObjectUpdateException (Java class), 255
ObjectUpdateException() (Java constructor), 255
ObjectUpdateException(Throwable) (Java constructor), 256
onAuthenticationSuccess(HttpServletRequest, HttpServletResponse, Authentication) (Java method), 410
ONE_TO_MANY_RELATIONSHIP (Java field), 242
ONE_TO_ONE_RELATIONSHIP (Java field), 243
oneTimeTokenOpenId(String) (Java method), 445
OneToManyRelationship (Java class), 198
OneToOneRelationship (Java class), 198
OPEN_ID (Java field), 414, 456
OpenIdUserValidator (Java class), 481
OpenIdUserValidator(UrlValidator) (Java constructor), 481

- Operators (Java class), 336
- OperatorType (Java enum), 512
- OPTIONS (Java field), 412
- Order (Java class), 349
- Order(String) (Java constructor), 349
- Order(String, Direction) (Java constructor), 350
- Order(String, String) (Java constructor), 350
- org.motechproject.admin.domain (package), 77
- org.motechproject.admin.messages (package), 82
- org.motechproject.admin.service (package), 84
- org.motechproject.bundle.extender (package), 86
- org.motechproject.commons.api (package), 88
- org.motechproject.commons.api.json (package), 96
- org.motechproject.commons.api.model (package), 97
- org.motechproject.commons.date (package), 97
- org.motechproject.commons.date.model (package), 97
- org.motechproject.commons.date.util (package), 100
- org.motechproject.commons.date.util.datetime (package), 105
- org.motechproject.commons.date.valueobjects (package), 106
- org.motechproject.commons.sql.service (package), 107
- org.motechproject.config.core (package), 108
- org.motechproject.config.core.constants (package), 108
- org.motechproject.config.core.domain (package), 111
- org.motechproject.config.core.filestore (package), 118
- org.motechproject.config.core.filters (package), 119
- org.motechproject.config.core.service (package), 120
- org.motechproject.config.domain (package), 121
- org.motechproject.config.monitor (package), 123
- org.motechproject.config.service (package), 124
- org.motechproject.email.builder (package), 131
- org.motechproject.email.contract (package), 135
- org.motechproject.email.domain (package), 136
- org.motechproject.email.service (package), 142
- org.motechproject.event (package), 145
- org.motechproject.event.listener (package), 148
- org.motechproject.event.listener.annotations (package), 149
- org.motechproject.event.queue (package), 152
- org.motechproject.mds.annotations (package), 154
- org.motechproject.mds.builder (package), 157
- org.motechproject.mds.config (package), 161
- org.motechproject.mds.domain (package), 166
- org.motechproject.mds.dto (package), 210
- org.motechproject.mds.enhancer (package), 246
- org.motechproject.mds.ex (package), 247
- org.motechproject.mds.ex.rest (package), 258
- org.motechproject.mds.filter (package), 259
- org.motechproject.mds.javassist (package), 261
- org.motechproject.mds.jdo (package), 268
- org.motechproject.mds.lookup (package), 274
- org.motechproject.mds.performance.domain (package), 275
- org.motechproject.mds.performance.service (package), 275
- org.motechproject.mds.performance.service.impl (package), 278
- org.motechproject.mds.query (package), 280
- org.motechproject.mds.repository (package), 289
- org.motechproject.mds.rest (package), 296
- org.motechproject.mds.service (package), 299
- org.motechproject.mds.service.impl (package), 315
- org.motechproject.mds.service.impl.history (package), 324
- org.motechproject.mds.util (package), 329
- org.motechproject.osgi.web (package), 355
- org.motechproject.osgi.web.domain (package), 374
- org.motechproject.osgi.web.exception (package), 375
- org.motechproject.osgi.web.ext (package), 376
- org.motechproject.osgi.web.service (package), 378
- org.motechproject.osgi.web.settings (package), 380
- org.motechproject.osgi.web.util (package), 381
- org.motechproject.scheduler.builder (package), 383
- org.motechproject.scheduler.contract (package), 385
- org.motechproject.scheduler.exception (package), 396
- org.motechproject.scheduler.factory (package), 397
- org.motechproject.scheduler.service (package), 398
- org.motechproject.scheduler.service.impl (package), 402
- org.motechproject.security.annotations (package), 407
- org.motechproject.security.authentication (package), 407
- org.motechproject.security.builder (package), 410
- org.motechproject.security.constants (package), 412
- org.motechproject.security.domain (package), 416
- org.motechproject.security.ex (package), 425
- org.motechproject.security.model (package), 427
- org.motechproject.security.repository (package), 435
- org.motechproject.security.service (package), 441
- org.motechproject.server.api (package), 447
- org.motechproject.server.config (package), 454
- org.motechproject.server.config.domain (package), 456
- org.motechproject.server.config.service (package), 464
- org.motechproject.server.startup (package), 465
- org.motechproject.server.ui (package), 466
- org.motechproject.server.ui.ex (package), 467
- org.motechproject.server.web.controller (package), 467
- org.motechproject.server.web.form (package), 472
- org.motechproject.server.web.helper (package), 478
- org.motechproject.server.web.validator (package), 480
- org.motechproject.tasks.annotations (package), 484
- org.motechproject.tasks.contract (package), 485
- org.motechproject.tasks.domain (package), 492
- org.motechproject.tasks.ex (package), 536
- org.motechproject.tasks.json (package), 537
- org.motechproject.tasks.repository (package), 539
- org.motechproject.tasks.service (package), 539
- org.motechproject.tasks.util (package), 549
- OSGI_FRAMEWORK_STORAGE (Java field), 113

OsgiDispatcherServlet (Java class), [365](#)
OsgiDispatcherServlet(BundleContext) (Java constructor), [365](#)
OsgiDispatcherServlet(BundleContext, ConfigurableWebApplicationContext) (Java constructor), [366](#)
OsgiWebApplicationContext (Java class), [366](#)
OsgiWebApplicationContext(ApplicationContext, ConfigurableWebApplicationContext) (Java constructor), [366](#)
OutboundEventGateway (Java interface), [153](#)
OWNER (Java field), [352](#)
OWNER_DISPLAY_FIELD_NAME (Java field), [342](#)
OWNER_FIELD_NAME (Java field), [342](#)
OwnerValueGenerator (Java class), [273](#)
OWNING_SIDE (Java field), [336](#)

P

PACKAGE_JDO (Java field), [308](#)
Packages (Java class), [337](#)
PackagesGenerated (Java class), [338](#)
Pair (Java interface), [350](#)
PARAM_DISCARDED_MOTech_EVENT (Java field), [145](#)
PARAM_INVALID_MOTech_EVENT (Java field), [145](#)
PARAM_REDELIVERY_COUNT (Java field), [145](#)
Parameter (Java class), [513](#)
Parameter() (Java constructor), [513](#)
Parameter(String, ParameterType) (Java constructor), [514](#)
ParameterType (Java enum), [514](#)
paramsDeclarationForQuery() (Java method), [260](#)
parse(Object, Class) (Java method), [354](#)
parse(Object, String) (Java method), [354](#)
parse(Object, String, ClassLoader) (Java method), [354](#)
parse(Object, String, String) (Java method), [354](#)
parse(Object, String, String, ClassLoader) (Java method), [354](#)
parse(String) (Java method), [206](#), [507](#)
parse(String, Locale) (Java method), [104](#)
parseAll(String) (Java method), [508](#)
parseDateTime(String) (Java method), [104](#)
ParseException (Java class), [97](#)
ParseException(String) (Java constructor), [97](#)
parseIntToBool(Integer) (Java method), [354](#)
parseList(List, Class) (Java method), [354](#)
parseNumber(Object, String) (Java method), [354](#)
parsePeriod(String) (Java method), [104](#)
parseString(String, Class) (Java method), [354](#)
parseString(String, Class, Class) (Java method), [355](#)
parseString(String, String) (Java method), [354](#)
parseStringToMap(String) (Java method), [355](#)
parseTime(String, String) (Java method), [100](#)
PASSWORD (Java field), [472](#)
PASSWORD_CONFIRMATION (Java field), [472](#)
PasswordRecoveriesDataService (Java interface), [440](#)

PasswordRecovery (Java class), [424](#)
passwordRecoveryRequest(String) (Java method), [446](#)
PasswordRecoveryService (Java interface), [445](#)
PAST_7_DAYS (Java field), [261](#)
PATH (Java field), [214](#)
PERIOD (Java field), [243](#)
PermissionDto (Java class), [427](#)
PermissionDto() (Java constructor), [427](#)
PermissionDto(MotechPermission) (Java constructor), [427](#)
PermissionDto(String, String) (Java constructor), [428](#)
PermissionNames (Java class), [413](#)
PERSIST (Java field), [332](#)
PersistedUserValidator (Java class), [481](#)
PersistedUserValidator(MotechUserService) (Java constructor), [481](#)
PLATFORM_CORE_CONFIG_FILTER (Java field), [119](#)
PLATFORM_SETTINGS_CHANGED_EVENT_SUBJECT (Java field), [110](#)
PLUSDAYS (Java field), [511](#)
POSITIVE (Java field), [239](#)
POST (Java field), [412](#)
postMessage(StatusMessage) (Java method), [85](#)
postMessage(String, String, Level) (Java method), [85](#)
postMessage(String, String, Level, DateTime) (Java method), [85](#)
postProcessAfterInitialization(Object, String) (Java method), [407](#), [485](#)
postProcessBeforeInitialization(Object, String) (Java method), [407](#), [485](#)
postProcessWebApplicationContext(ConfigurableWebApplicationContext) (Java method), [366](#)
prefixEnumValue(String) (Java method), [343](#)
prefixEnumValues(List) (Java method), [343](#)
prepareHistoryClass(Entity) (Java method), [158](#)
prepareTrashClass(Entity) (Java method), [158](#)
processAnnotations(ApplicationContext) (Java method), [407](#), [485](#)
processExistingConfigs(List) (Java method), [128](#)
PROPERTIES_EXTENSION (Java field), [110](#)
PROPERTIES_FILE_EXTENSION (Java field), [121](#)
PropertiesReader (Java class), [119](#)
Property (Java class), [282](#)
Property(String, T) (Java constructor), [282](#)
PropertyBuilder (Java class), [283](#)
PropertyUtil (Java class), [351](#)
Protocol (Java enum), [414](#)
PROVIDER_NAME (Java field), [111](#), [474](#)
PROVIDER_URL (Java field), [111](#), [474](#)
publishEvent(ApplicationEvent) (Java method), [369](#)
publishWarningActivity(String, String) (Java method), [545](#)
PUT (Java field), [412](#)

Q

[QueryExecution](#) (Java interface), [284](#)
[QueryExecutor](#) (Java class), [284](#)
[QueryParams](#) (Java class), [285](#)
[QueryParams\(Integer, Integer\)](#) (Java constructor), [285](#)
[QueryParams\(Integer, Integer, Order\)](#) (Java constructor), [285](#)
[QueryParams\(Order\)](#) (Java constructor), [285](#)
[QueryUtil](#) (Java class), [286](#)
[QUEUE_URL](#) (Java field), [474](#)
[QueueMBean](#) (Java class), [79](#)
[QueueMBean\(String\)](#) (Java constructor), [79](#)
[QueueMessage](#) (Java class), [80](#)
[QueueMessage\(String, Boolean, Date\)](#) (Java constructor), [81](#)
[QueueURLValidator](#) (Java class), [481](#)
[QueueURLValidator\(\)](#) (Java constructor), [481](#)
[QueueURLValidator\(UrlValidator\)](#) (Java constructor), [482](#)

R

[Range](#) (Java class), [94](#)
[RANGE](#) (Java field), [233](#)
[Range\(T, T\)](#) (Java constructor), [94](#)
[RangeProperty](#) (Java class), [287](#)
[RangeProperty\(String, Range\)](#) (Java constructor), [287](#)
[RAW_DIR](#) (Java field), [111](#)
[rawConfigExists\(String, String\)](#) (Java method), [128](#)
[READ](#) (Java field), [201](#)
[READABLE](#) (Java field), [116](#)
[readFromFile\(String, Type\)](#) (Java method), [96](#)
[readFromStream\(InputStream, Type\)](#) (Java method), [96](#)
[readFromStreamOnlyExposeAnnotations\(InputStream, Type\)](#) (Java method), [96](#)
[readFromString\(String, Type\)](#) (Java method), [96](#)
[readFromString\(String, Type, Map\)](#) (Java method), [96](#)
[readFromStringOnlyExposeAnnotations\(String, Type\)](#) (Java method), [96](#)
[rebuildProxyChain\(\)](#) (Java method), [442](#)
[RECEIVED](#) (Java field), [137](#)
[reconfigure\(\)](#) (Java method), [380](#)
[record\(Object\)](#) (Java method), [306](#), [326](#)
[REDIRECT_BOOTSTRAP](#) (Java field), [467](#)
[REDIRECT_HOME](#) (Java field), [467](#)
[REDIRECT_STARTUP](#) (Java field), [467](#)
[refreshAllUsersContextIfActive\(\)](#) (Java method), [447](#)
[refreshUserContextIfActive\(String\)](#) (Java method), [447](#)
[regenerateMdsDataBundle\(boolean\)](#) (Java method), [308](#), [319](#)
[regenerateMdsDataBundle\(boolean, boolean\)](#) (Java method), [308](#), [319](#)
[regenerateMdsDataBundleAfterDdeEnhancement\(String\)](#) (Java method), [308](#), [320](#)
[REGEXP](#) (Java field), [332](#)
[register\(String, String, String, String, List, Locale\)](#) (Java method), [445](#)
[register\(String, String, String, String, List, Locale, boolean, String\)](#) (Java method), [445](#)
[registerAllProperties\(\)](#) (Java method), [455](#)
[registerAllRawConfig\(\)](#) (Java method), [455](#)
[registerChannel\(ChannelRequest\)](#) (Java method), [540](#)
[registerChannel\(InputStream, String, String\)](#) (Java method), [540](#)
[registerDDE\(String\)](#) (Java method), [267](#)
[registeredEnums\(\)](#) (Java method), [267](#)
[registeredInterfaces\(\)](#) (Java method), [267](#)
[registerEnhancedClassData\(ClassData\)](#) (Java method), [267](#)
[registerEnum\(String\)](#) (Java method), [267](#)
[registerHandlerFor\(String\)](#) (Java method), [548](#), [549](#)
[registerHistoryClassData\(ClassData\)](#) (Java method), [267](#)
[registerListener\(EventListener, List\)](#) (Java method), [149](#)
[registerListener\(EventListener, String\)](#) (Java method), [149](#)
[registerModule\(ModuleRegistrationData\)](#) (Java method), [372](#)
[registerProperties\(String, Properties\)](#) (Java method), [455](#)
[registerProvider\(InputStream\)](#) (Java method), [546](#)
[registerProvider\(String\)](#) (Java method), [546](#)
[registerServiceInterface\(String, String\)](#) (Java method), [267](#)
[registersProperties\(String, String\)](#) (Java method), [129](#)
[registerTrashClassData\(ClassData\)](#) (Java method), [267](#)
[registerUserClassLoader\(ClassLoader\)](#) (Java method), [270](#)
[RELATED_CLASS](#) (Java field), [336](#)
[RELATED_FIELD](#) (Java field), [336](#)
[Relationship](#) (Java class), [199](#)
[RelationshipHolder](#) (Java class), [199](#)
[RelationshipHolder\(ClassData, Field\)](#) (Java constructor), [199](#)
[RelationshipHolder\(Field\)](#) (Java constructor), [199](#)
[reloadClassLoader\(\)](#) (Java method), [345](#)
[reloadMetadata\(\)](#) (Java method), [293](#)
[remove\(MotechRole\)](#) (Java method), [436](#)
[remove\(MotechSecurityConfiguration\)](#) (Java method), [437](#)
[remove\(MotechUser\)](#) (Java method), [438](#)
[remove\(Object\)](#) (Java method), [306](#), [326](#)
[remove>PasswordRecovery\)](#) (Java method), [439](#)
[REMOVE_INDEX](#) (Java field), [214](#)
[removeAction\(String\)](#) (Java method), [241](#)
[removeAll\(\)](#) (Java method), [525](#)
[removeAngularModule\(String\)](#) (Java method), [363](#)
[removeDataProvider\(String\)](#) (Java method), [548](#)
[removeDataSources\(\)](#) (Java method), [525](#)
[removeDeclaredFieldIfExists\(CtClass, String\)](#) (Java method), [264](#)

`removeDeclaredMethodIfExists(CtClass, String)` (Java method), 264

`removeDefaults(Properties)` (Java method), 462

`removedService(ServiceReference, Object)` (Java method), 357, 359, 373

`removeField(Integer)` (Java method), 230

`removeField(Long)` (Java method), 175, 230

`removeField(Number)` (Java method), 236, 241

`removeFieldIfExists(CtClass, String)` (Java method), 264

`removeFilterableField(Number)` (Java method), 213

`removeFilterSets()` (Java method), 525

`removeFromProcessed(ApplicationContext)` (Java method), 356

`removeIndex(Integer)` (Java method), 211

`removeLogger(String)` (Java method), 380

`removeLookup(Long)` (Java method), 175

`removeLookup(Number)` (Java method), 236

`removeMessage(StatusMessage)` (Java method), 85

`removeMetadata(Integer)` (Java method), 224

`removeMethodIfExists(CtClass, String)` (Java method), 264

`removeModuleRecords(List)` (Java method), 129

`removeNotificationRule(Long)` (Java method), 85

`removeNotificationRule(String)` (Java method), 85

`removePermission(String)` (Java method), 417

`removeProperties(String, String)` (Java method), 129

`removeTrackerFor(Bundle)` (Java method), 360, 373

`removeValidationError(String)` (Java method), 518

`render(Map, HttpServletRequest, HttpServletResponse)` (Java method), 359

`RenderException` (Java class), 375

`RenderException(String)` (Java constructor), 375

`RenderException(String, Throwable)` (Java constructor), 375

`RenderException(Throwable)` (Java constructor), 375

`RepeatingJobId` (Java class), 392

`RepeatingJobId(MotechEvent)` (Java constructor), 392

`RepeatingJobId(String, String)` (Java constructor), 392

`RepeatingSchedulableJob` (Java class), 392

`RepeatingSchedulableJob()` (Java constructor), 392

`RepeatingSchedulableJob(MotechEvent, Date, Date, Integer, Long, boolean)` (Java constructor), 392

`RepeatingSchedulableJob(MotechEvent, Date, Date, Long, boolean)` (Java constructor), 393

`REPOSITORY` (Java field), 337, 338, 456

`REQUIRE` (Java field), 239

`RequiredFieldValidator` (Java class), 482

`RequiredFieldValidator(String, String)` (Java constructor), 482

`requiresConfigurationFiles()` (Java method), 129

`requiresFiltering()` (Java method), 260

`rescheduleJob(String, String, String)` (Java method), 399, 405

`ReservedKeywordException` (Java class), 256

`ReservedKeywordException(String)` (Java constructor), 256

`reset(ResetForm, HttpServletRequest)` (Java method), 471

`ResetController` (Java class), 470

`ResetForm` (Java class), 472

`ResetFormValidator` (Java class), 482

`resetPassword(String, String, String)` (Java method), 446

`resetView(HttpServletRequest)` (Java method), 471

`RESOLVED` (Java field), 450

`resolveManyToManyRelationship(Field, Object, Object)` (Java method), 329

`RESOURCE_PATH` (Java field), 381

`resourceFolder()` (Java method), 373

`resourceMappings()` (Java method), 355, 373

`RestBadBodyFormatException` (Java class), 258

`RestBadBodyFormatException(String)` (Java constructor), 258

`RestBadBodyFormatException(String, Throwable)` (Java constructor), 258

`RestExposed` (Java annotation), 155

`restId(String, String, String)` (Java method), 330

`RestIgnore` (Java annotation), 156

`RestLookupExecutionForbiddenException` (Java class), 258

`RestLookupExecutionForbiddenException(String)` (Java constructor), 258

`RestLookupNotFoundException` (Java class), 259

`RestLookupNotFoundException(String)` (Java constructor), 259

`RestNotSupportedException` (Java class), 259

`RestNotSupportedException(String, String, String)` (Java constructor), 259

`RestOperation` (Java enum), 200

`RestOperationNotSupportedException` (Java class), 259

`RestOperationNotSupportedException(String)` (Java constructor), 259

`RestOperations` (Java annotation), 156

`RestOptions` (Java class), 201

`RestOptions()` (Java constructor), 201

`RestOptions(Entity)` (Java constructor), 201

`RestOptionsDto` (Java class), 235

`RestOptionsDto()` (Java constructor), 235

`RestOptionsDto(boolean, boolean, boolean, boolean)` (Java constructor), 235

`RestrictionProperty` (Java class), 287

`RestrictionProperty(InstanceSecurityRestriction, String)` (Java constructor), 288

`RETRIEVAL_RETRIES_COUNT` (Java field), 322

`retrieve(Entity, String)` (Java method), 291

`retrieve(Object)` (Java method), 295

`retrieve(String, Object)` (Java method), 295, 301, 312

`retrieve(String, Object, InstanceSecurityRestriction)` (Java method), 295

`retrieve(String[], Object[])` (Java method), 295

- `retrieve(String[], Object[], InstanceSecurityRestriction)` (Java method), 295
- `retrieveAll()` (Java method), 295, 301, 312
- `retrieveAll(Class)` (Java method), 310, 321
- `retrieveAll(Class, QueryParams)` (Java method), 310, 322
- `retrieveAll(Entity)` (Java method), 291
- `retrieveAll(InstanceSecurityRestriction)` (Java method), 295
- `retrieveAll(List)` (Java method), 301, 313
- `retrieveAll(List, InstanceSecurityRestriction)` (Java method), 296
- `retrieveAll(List, QueryParams)` (Java method), 301, 313
- `retrieveAll(List, QueryParams, InstanceSecurityRestriction)` (Java method), 296
- `retrieveAll(QueryParams)` (Java method), 301, 312
- `retrieveAll(QueryParams, InstanceSecurityRestriction)` (Java method), 296
- `retrieveAll(String)` (Java method), 291, 310, 322
- `retrieveAll(String, Object)` (Java method), 296
- `retrieveAll(String, Object, InstanceSecurityRestriction)` (Java method), 296
- `retrieveAll(String, QueryParams)` (Java method), 310, 322
- `retrieveAll(String[], Object[], InstanceSecurityRestriction)` (Java method), 296
- `retrieveAll(String[], Object[], QueryParams, InstanceSecurityRestriction)` (Java method), 296
- `retrieveByClassName(String)` (Java method), 290, 292
- `retrieveById(Long)` (Java method), 290
- `retrieveRegisteredBundleNames()` (Java method), 129
- `retrieveUserByCredentials(String, String)` (Java method), 445
- `revertFromTrash(Object, Object)` (Java method), 301, 312
- `ROLE_ACCESS_PREFIX` (Java field), 415
- `RoleDto` (Java class), 428
- `RoleDto()` (Java constructor), 428
- `RoleDto(MotechRole)` (Java constructor), 429
- `RoleDto(String, List)` (Java constructor), 429
- `RoleDto(String, List, boolean)` (Java constructor), 429
- `RoleHasUserException` (Java class), 426
- `RoleHasUserException(String)` (Java constructor), 426
- `Roles` (Java class), 338
- `ROLES` (Java field), 352
- `ROLES_ADMIN` (Java field), 415
- `ROOT_LOGGER_NAME` (Java field), 379
- `RunOnceJobId` (Java class), 394
- `RunOnceJobId(MotechEvent)` (Java constructor), 395
- `RunOnceJobId(String, String)` (Java constructor), 395
- `RunOnceSchedulableJob` (Java class), 395
- `RunOnceSchedulableJob(MotechEvent, Date)` (Java constructor), 395
- `safeScheduleJob(CronSchedulableJob)` (Java method), 400, 405
- `safeScheduleRepeatingJob(RepeatingSchedulableJob)` (Java method), 400, 405
- `safeScheduleRunOnceJob(RunOnceSchedulableJob)` (Java method), 400, 405
- `safeSetProperty(Object, String, Object)` (Java method), 351
- `safeUnscheduleAllJobs(String)` (Java method), 400, 405
- `safeUnscheduleJob(String, String)` (Java method), 400, 405
- `safeUnscheduleRepeatingJob(String, String)` (Java method), 400, 405
- `safeUnscheduleRunOnceJob(String, String)` (Java method), 401, 406
- `sameAs(Object)` (Java method), 122
- `Sample` (Java class), 275
- `Sample()` (Java constructor), 275
- `Sample(Integer, String)` (Java constructor), 275
- `SampleService` (Java interface), 278
- `Saturday` (Java field), 98
- `save(BootstrapConfig)` (Java method), 130
- `save(Task)` (Java method), 548
- `saveBootstrapConfig(BootstrapConfig)` (Java method), 121
- `saveConfigProperties(String, Properties)` (Java method), 455
- `saveDraftEntityChanges(Long, DraftData)` (Java method), 305, 318
- `saveDraftEntityChanges(Long, DraftData, String)` (Java method), 305, 318
- `saveModuleSettings(ModuleSettings)` (Java method), 165
- `saveNotificationRules(List)` (Java method), 85
- `savePlatformSetting(String, String)` (Java method), 458, 462
- `savePlatformSettings(MotechSettings)` (Java method), 130, 455
- `savePlatformSettings(Properties)` (Java method), 130
- `saveRawConfig(String, Resource)` (Java method), 455
- `saveRawConfig(String, String, String, String, InputStream)` (Java method), 130
- `saveRule(NotificationRule)` (Java method), 85
- `scheduleCronJob(String, Map, String, DateTime, DateTime, Boolean)` (Java method), 398, 403
- `scheduleDayOfWeekJob(DayOfWeekSchedulableJob)` (Java method), 401, 406
- `scheduleDayOfWeekJob(String, Map, DateTime, DateTime, List, DateTime, Boolean)` (Java method), 398, 403
- `scheduleEmptyTrashJob()` (Java method), 315, 328
- `scheduleJob(CronSchedulableJob)` (Java method), 401, 406
- `SCHEDULER_SYMBOLIC_NAME` (Java field), 323
- `scheduleRepeatingJob(long)` (Java method), 310, 323

S

- `safeDefineClass(String, byte[])` (Java method), 345
- `safeGetProperty(Object, String)` (Java method), 351

- `scheduleRepeatingJob(RepeatingSchedulableJob)` (Java method), 401, 406
- `scheduleRepeatingJob(String, Map, DateTime, DateTime, Integer, Long, Boolean, Boolean)` (Java method), 398, 404
- `SchedulerInstantiationException` (Java class), 396
- `SchedulerInstantiationException(Throwable)` (Java constructor), 397
- `SchedulerShutdownException` (Java class), 397
- `SchedulerShutdownException(Throwable)` (Java constructor), 397
- `scheduleRunOnceJob(RunOnceSchedulableJob)` (Java method), 401, 406
- `scheduleRunOnceJob(String, Map, DateTime)` (Java method), 398, 404
- `SCHEMA_ACCESS` (Java field), 339
- `SchemaGenerator` (Java class), 273
- `SchemaGenerator(JDOPersistenceManagerFactory)` (Java constructor), 273
- `schemaVersion(Class)` (Java method), 327
- `Scheme` (Java enum), 414
- `SECURITY` (Java field), 214
- `SECURITY_ADMIN_ROLE` (Java field), 416
- `SECURITY_REALM_KEY` (Java field), 408, 409
- `SecurityAnnotationBeanPostProcessor` (Java class), 407
- `SecurityAnnotationBeanPostProcessor(MotechPermissionsService)` (Java constructor), 407
- `SecurityConfigConstants` (Java class), 415
- `SecurityConfigDto` (Java class), 430
- `SecurityConfigException` (Java class), 427
- `SecurityConfigException(String)` (Java constructor), 427
- `SecurityException` (Java class), 256
- `SecurityException()` (Java constructor), 256
- `SecurityMode` (Java enum), 351
- `SecurityRoleLoader` (Java class), 446
- `SecurityRoleLoader(MotechRoleService)` (Java constructor), 446
- `SecurityRuleBuilder` (Java class), 410
- `SecurityRuleDto` (Java class), 430
- `SecurityRuleLoader` (Java class), 446
- `SecurityUtil` (Java class), 352
- `send(Mail)` (Java method), 145
- `sendEventMessage(MotechEvent)` (Java method), 149, 154
- `sendLoginInformation(String, String)` (Java method), 445
- `SENT` (Java field), 137
- `serialize(ModuleSettings, JsonGenerator, SerializerProvider)` (Java method), 165
- `Serializer` (Java class), 164
- `SERVER_URL` (Java field), 111
- `ServerLogService` (Java interface), 378
- `SERVICE` (Java field), 337, 338
- `SERVICE_IMPL` (Java field), 338
- `SERVICE_INTERFACE_FIELD` (Java field), 538
- `SERVICE_METHOD_FIELD` (Java field), 538
- `SERVICE_NAME` (Java field), 89
- `ServiceNotFoundException` (Java class), 256
- `ServiceNotFoundException()` (Java constructor), 257
- `ServiceUtil` (Java class), 352
- `ServletRegistrationException` (Java class), 375
- `ServletRegistrationException(String)` (Java constructor), 375
- `ServletRegistrationException(String, Throwable)` (Java constructor), 376
- `ServletRegistrationException(Throwable)` (Java constructor), 376
- `SET` (Java field), 233
- `setAbstractClass(boolean)` (Java method), 175, 220
- `setActionParameters(SortedSet)` (Java method), 493
- `setActions(List)` (Java method), 241, 518
- `setActionTaskEvents(List)` (Java method), 497
- `setActionType(ActionType)` (Java method), 78
- `setActive(boolean)` (Java method), 420, 423, 431, 434
- `setActivity(String)` (Java method), 390
- `setActivityType(TaskActivityType)` (Java method), 522
- `setAdminConfirmPassword(String)` (Java method), 475
- `setAdminEmail(String)` (Java method), 475
- `setAdminLogin(String)` (Java method), 475
- `setAdminPassword(String)` (Java method), 475
- `setAdminUsername(String)` (Java method), 478
- `setAfterTimeUnit(TimeUnit)` (Java method), 171
- `setAfterTimeValue(int)` (Java method), 171
- `setAllEntities(AllEntities)` (Java method), 298, 301, 318, 320, 325
- `setAllEntityAudits(AllEntityAudits)` (Java method), 318
- `setAllEntityDrafts(AllEntityDrafts)` (Java method), 318
- `setAllowCreate(boolean)` (Java method), 202, 204
- `setAllowDelete(boolean)` (Java method), 202, 204
- `setAllowRead(boolean)` (Java method), 202, 204
- `setAllowUpdate(boolean)` (Java method), 202, 204
- `setAllSecurityRules(AllMotechSecurityRules)` (Java method), 447
- `setAllTypes(AllTypes)` (Java method), 318, 324
- `setAllTypeValidations(AllTypeValidations)` (Java method), 324
- `setAngularModule(String)` (Java method), 449
- `setAnnotations(List)` (Java method), 210
- `setArgs(List)` (Java method), 530
- `setAuthenticationManager(AuthenticationManager)` (Java method), 411
- `setBasic(FieldBasicDto)` (Java method), 225, 226
- `setBasicAuthenticationEntryPoint(AuthenticationEntryPoint)` (Java method), 411
- `setBefore(String)` (Java method), 478
- `setBody(Resource)` (Java method), 89
- `setBody(String)` (Java method), 89
- `setBrokerUrl(String)` (Java method), 152
- `setBrowsing(BrowsingSettingsDto)` (Java method), 211

- setBundle(Bundle) (Java method), 363
- setBundle(String) (Java method), 122
- setBundleContext(BundleContext) (Java method), 319, 320, 325, 358, 359, 468, 470, 543, 548
- setBundleName(String) (Java method), 416, 428
- setByCreator(boolean) (Java method), 344
- setByOwner(boolean) (Java method), 344
- setChangesMade(boolean) (Java method), 179, 216
- setChannelDecisionManager(ChannelDecisionManager) (Java method), 411
- setChannelName(String) (Java method), 533
- setClassName(String) (Java method), 175, 181, 220
- setConfig(List) (Java method), 162
- setConfigFileChecksum(String) (Java method), 459, 462
- setConfigFiles(List) (Java method), 456
- setConfigLocation(String) (Java method), 88, 365
- setConfigurationScanner(ConfigurationScanner) (Java method), 87
- setConfigurationService(ConfigurationService) (Java method), 456
- setConsumerCount(long) (Java method), 80
- setCoreConfigurationService(CoreConfigurationService) (Java method), 464
- setCountResult(Query) (Java method), 286
- setCreate(boolean) (Java method), 215, 236
- setCreationDate(DateTime) (Java method), 197
- setCreator(String) (Java method), 197
- setCriteria(List) (Java method), 228
- setCriticalMessage(String) (Java method), 363, 370
- setCss(List) (Java method), 479
- setCustomOperator(String) (Java method), 232
- setCustomOperators(Map) (Java method), 195
- setDataProviders(Map) (Java method), 548
- setDataService(MotechDataService) (Java method), 298
- setDataService(MotechPermissionsDataService) (Java method), 435
- setDataService(MotechRolesDataService) (Java method), 436
- setDataService(MotechURLSecurityRuleDataService) (Java method), 437
- setDataService(MotechUsersDataService) (Java method), 438
- setDataService>PasswordRecoveriesDataService) (Java method), 439
- setDataSources(List) (Java method), 525
- setDate(DateTime) (Java method), 82, 522
- setDefaultName(String) (Java method), 206, 244
- setDefaultURL(String) (Java method), 363
- setDefaultValue(Object) (Java method), 222
- setDefaultValue(String) (Java method), 185, 208
- setDeletable(boolean) (Java method), 417, 429
- setDelete(boolean) (Java method), 237
- setDeleted(boolean) (Java method), 420, 432
- setDeleteMode>DeleteMode) (Java method), 163, 171
- setDeleteMode(String) (Java method), 163
- setDequeueCount(long) (Java method), 80
- setDescription(String) (Java method), 206, 244, 497, 518, 532
- setDestination(String) (Java method), 80
- setDetails>TypeSetting) (Java method), 190
- setDetails>TypeValidation) (Java method), 192
- setDisplayedFields(List) (Java method), 213
- setDisplayName(String) (Java method), 185, 206, 210, 222, 244, 246, 498, 504, 510, 514, 529, 532, 534
- setDraftOwnerUsername(String) (Java method), 179
- setDrafts(List) (Java method), 175
- setEdit(boolean) (Java method), 215
- setEmail(String) (Java method), 423, 425, 434
- setEmptyTrash(Boolean) (Java method), 164
- setEmptyTrash(boolean) (Java method), 171
- setEmptyTrash(String) (Java method), 164
- setEnabled(boolean) (Java method), 192, 246, 518
- setEndDate(String) (Java method), 390
- setEndTime(Date) (Java method), 147, 394
- setEnqueueCount(long) (Java method), 80
- setEntity(Entity) (Java method), 167, 185, 195, 202, 204
- setEntityId(Long) (Java method), 211, 225
- setEntityName(String) (Java method), 181
- setEntityPrefix(String) (Java method), 278, 280
- setEntityService(EntityService) (Java method), 301
- setEntityValidator(EntityValidator) (Java method), 319
- setEntityVersion(Long) (Java method), 176
- setEventAdmin(EventAdmin) (Java method), 464
- setEventInfoList(List) (Java method), 391
- setEventKey(String) (Java method), 501
- setEventParameters(List) (Java method), 535
- setExpirationDate(DateTime) (Java method), 425
- setExpiredCount(long) (Java method), 80
- setExposedViaRest(boolean) (Java method), 185, 195, 230
- setExpression(String) (Java method), 504
- setExternalId(String) (Java method), 423, 434
- setFailIfDataNotFound(boolean) (Java method), 500
- setField(Field) (Java method), 189, 190, 192
- setField(Object, String, List) (Java method), 343
- setField(String) (Java method), 260, 509, 522
- setFieldIds(List) (Java method), 237
- setFieldKey(String) (Java method), 502
- setFieldName(String) (Java method), 349
- setFieldNameChanges(Map) (Java method), 179
- setFieldPrefix(String) (Java method), 278, 280
- setFields(List) (Java method), 176, 195, 241, 510, 522, 529
- setFileMonitor(DefaultFileMonitor) (Java method), 124
- setFilename(String) (Java method), 122
- setFilePath(String) (Java method), 459, 462
- setFilterableFields(List) (Java method), 213

setFilters(List) (Java method), 505, 525
setFromAddress(String) (Java method), 139
setGeneratePassword(boolean) (Java method), 434
setHasRegisteredChannel(boolean) (Java method), 518
setHistoryService(HistoryService) (Java method), 301, 328
setHour(Integer) (Java method), 100
setId(Long) (Java method), 79, 139, 172, 176, 185, 189, 190, 192, 195, 198, 202, 204, 207–210, 212, 220, 225, 226, 230, 232, 234, 237, 244, 420, 432, 518, 527
setId(UUID) (Java method), 147
setIgnorePastFiresAtStart(boolean) (Java method), 394
setIndexes(List) (Java method), 212
setInflightCount(long) (Java method), 80
setInfo(String) (Java method), 390
setInstanceId(Long) (Java method), 226
setInterfaceName(String) (Java method), 181
setJobType(String) (Java method), 390
setJs(List) (Java method), 479
setKey(String) (Java method), 189, 234, 495, 504
setLanguage(String) (Java method), 459, 463, 475
setLastEvent(boolean) (Java method), 147
setLastModificationDate(DateTime) (Java method), 179
setLastRun(DateTime) (Java method), 459, 463
setLevel(Level) (Java method), 79, 82
setLib(List) (Java method), 479
setLocale(Locale) (Java method), 423, 425, 434
setLocale(String, Locale) (Java method), 445
setLocaleService(LocaleService) (Java method), 468, 470
setLog4jConf(String) (Java method), 360
setLoggers(List) (Java method), 381
setLoginAuthenticationEntryPoint(AuthenticationEntryPoint) (Java method), 411
setLoginMode(String) (Java method), 475
setLoginModeValue(String) (Java method), 459, 463
setLogLevel(String) (Java method), 375
setLogName(String) (Java method), 375
setLookup(Object) (Java method), 500
setLookupFields(List) (Java method), 230, 529
setLookupIds(List) (Java method), 237
setLookupName(String) (Java method), 195, 231
setLookupPrefix(String) (Java method), 278, 280
setLookups(List) (Java method), 176, 225
setLookups(Set) (Java method), 185
setMappingFieldName(String) (Java method), 349
setMDSConstructor(MDSConstructor) (Java method), 319
setMdsConstructor(MDSConstructor) (Java method), 320
setMdsDataProvider(MDSDataProvider) (Java method), 320
setMdsSchedulerService(MdsSchedulerService) (Java method), 328
setMenuBuilder(MenuBuilder) (Java method), 470
setMessage(String) (Java method), 139, 522, 530
setMetadata(List) (Java method), 185, 225
setMetadataHolder(MetadataHolder) (Java method), 320
setMethodName(String) (Java method), 195, 231
setMethodsRequired(List) (Java method), 420, 432
setMinute(Integer) (Java method), 100
setModificationDate(DateTime) (Java method), 178, 198
setModified(boolean) (Java method), 220
setModifiedBy(String) (Java method), 198
setModule(String) (Java method), 123, 176, 181, 220
setModuleName(String) (Java method), 79, 82, 363, 449, 456, 490, 498, 534
setModuleVersion(String) (Java method), 490, 498, 534
setMonitor(EntitiesBundleMonitor) (Java method), 320
setMotechEvent(MotechEvent) (Java method), 394
setMotechLogoutHandler(MotechLogoutSuccessHandler) (Java method), 411
setName(String) (Java method), 176, 185, 208, 209, 220, 222, 233, 239, 390, 500, 518, 528
setNamespace(String) (Java method), 88, 176, 181, 220, 365
setNeedsAttention(boolean) (Java method), 363, 370
setNegationOperator(boolean) (Java method), 504
setNextFireDate(String) (Java method), 390
setObjectId(Long) (Java method), 500
setObjects(List) (Java method), 528
setOpenId(String) (Java method), 423, 434
setOpenIDAuthenticationFilter(OpenIDAuthenticationFilter) (Java method), 411
setOperator(String) (Java method), 504
setOptions(List) (Java method), 239
setOrder(int) (Java method), 488
setOrder(Integer) (Java method), 495, 526
setOrder(String) (Java method), 478
setOrigin(String) (Java method), 420, 432
setOriginalRoleName(String) (Java method), 429
setOutdated(boolean) (Java method), 216, 220
setOwner(String) (Java method), 198
setOwnerUsername(String) (Java method), 178
setParameters(Map) (Java method), 388
setParentEntity(Entity) (Java method), 179
setParentVersion(Long) (Java method), 180
setPassword(String) (Java method), 152, 423, 434, 472, 473
setPasswordConfirmation(String) (Java method), 473
setPath(String) (Java method), 479
setPattern(String) (Java method), 421, 432
setPermissionAccess(List) (Java method), 421, 432
setPermissionName(String) (Java method), 416, 428
setPermissionNames(List) (Java method), 418, 430
setPersistenceManagerFactory(PersistenceManagerFactory) (Java method), 293, 296, 325
setPlatformInitialized(boolean) (Java method), 459, 463
setPlatformSetting(String, String) (Java method), 130

- setPlatformSettings(Map) (Java method), 463
- setPrimary(ClassLoader) (Java method), 270
- setPriority(int) (Java method), 421, 432
- setProperties(EntityDraft, Entity) (Java method), 291
- setProperties(EntityDraft, Entity, String) (Java method), 291
- setProperties(Map) (Java method), 123
- SetProperty (Java class), 288
- SetProperty(String, Set) (Java constructor), 288
- setProperty(String, String) (Java method), 456
- setProtocol(Protocol) (Java method), 421
- setProtocol(String) (Java method), 432
- setProviderId(Long) (Java method), 500
- setProviderName(String) (Java method), 459, 463, 475, 500
- setProviderUrl(String) (Java method), 459, 463, 476
- setProxy(FilterChainProxy) (Java method), 442
- setProxyManager(MotechProxyManager) (Java method), 447
- setQueryParams(Query, QueryParams) (Java method), 286
- setQueueSize(long) (Java method), 80
- setQueueUrl(String) (Java method), 476
- setRangeLookupFields(List) (Java method), 195
- setRaw(boolean) (Java method), 123
- setRawConfigFiles(List) (Java method), 456
- setRead(boolean) (Java method), 237
- setReadOnly(boolean) (Java method), 77, 186, 195, 220, 225, 231
- setRecipient(String) (Java method), 79
- setReference(Object) (Java method), 349
- setReferenced(boolean) (Java method), 231
- setRemove(boolean) (Java method), 216
- setRepeatCount(Integer) (Java method), 394
- setRepeatIntervalInMilliseconds(Long) (Java method), 394
- setRepository(MotechDataRepository) (Java method), 301
- setRepository(String) (Java method), 181
- setRequired(boolean) (Java method), 186, 222, 495
- setResourceLoader(ResourceLoader) (Java method), 464
- setResourcePath(String) (Java method), 363
- setRest(boolean) (Java method), 421, 432
- setRestOptions(RestOptions) (Java method), 176
- setRestOptions(RestOptionsDto) (Java method), 212
- setRoleForAccess(List) (Java method), 364, 370
- setRoleForAccess(String) (Java method), 364, 370
- setRoleName(String) (Java method), 418, 430
- setRoles(List) (Java method), 423, 434
- setRoot(LogMapping) (Java method), 381
- setRows(List) (Java method), 142
- setRuntimeClassLoader(ClassLoader) (Java method), 271
- setSchedulerUrl(String) (Java method), 476
- setSecurity(SecurityMode, List) (Java method), 176
- setSecurityMembers(Set) (Java method), 176, 220
- setSecurityMode(SecurityMode) (Java method), 176, 221
- setSecurityRuleBuilder(SecurityRuleBuilder) (Java method), 442
- setSecurityRules(List) (Java method), 418, 430
- setSecurityRulesDAO(AllMotechSecurityRules) (Java method), 442
- setServerUrl(String) (Java method), 459, 463
- setServiceInterface(String) (Java method), 493, 520
- setServiceMethod(String) (Java method), 493, 520
- setServiceName(String) (Java method), 181
- setServletConfig(ServletConfig) (Java method), 88, 365
- setServletContext(ServletContext) (Java method), 88, 365
- setSessionHandler(SessionHandler) (Java method), 410
- setSetLookupFields(List) (Java method), 196
- setSettings(List) (Java method), 186, 207, 225
- setSettingsFacade(SettingsFacade) (Java method), 411
- setSettingsList(List) (Java method), 77
- setSettingsService(SettingsService) (Java method), 328
- setSettingsURL(String) (Java method), 364, 450
- setSingleObjectReturn(boolean) (Java method), 196, 231
- setSourceInstance(DateTimeSource) (Java method), 100
- setStackTraceElement(String) (Java method), 522
- setStartDate(String) (Java method), 390
- setStartTime(Date) (Java method), 394
- setStartupFormValidatorFactory(StartupFormValidatorFactory) (Java method), 471
- setStartupManager(StartupManager) (Java method), 468
- setStatus(String) (Java method), 391
- setStatusMsgTimeout(String) (Java method), 459, 463
- setSubject(String) (Java method), 139, 388, 532, 534
- setSubMenu(Map) (Java method), 364
- setSuperClass(String) (Java method), 177, 221
- setSupportedSchemes(List) (Java method), 421, 432
- setTask(Long) (Java method), 522
- setTaskConfig(TaskConfig) (Java method), 518
- setTaskPossibleErrors(String) (Java method), 516
- SETTER_PREFIX (Java field), 347
- setTestInt(Integer) (Java method), 275
- setTestString(String) (Java method), 275
- setText(String) (Java method), 82
- setTimeout(DateTime) (Java method), 82
- setTimeUnit(String) (Java method), 164
- setTimeUnit(TimeUnit) (Java method), 164
- setTimeValue(Integer) (Java method), 164
- setTimeValue(String) (Java method), 164
- setTimeZone(DateTime) (Java method), 103
- setTimeZoneUTC(DateTime) (Java method), 103
- SettingDto (Java class), 237
- SettingDto() (Java constructor), 237
- SettingDto(String, Object) (Java constructor), 238
- SettingDto(String, Object, TypeDto, SettingOptions) (Java constructor), 238
- SettingOptions (Java enum), 239

Settings (Java class), 340
SETTINGS (Java field), 111
SETTINGS_ACCESS (Java field), 340
SETTINGS_CACHE_NAME (Java field), 124
SETTINGS_FILE_NAME (Java field), 111
SettingsDto (Java class), 515
SettingService (Java interface), 464
SettingsFacade (Java class), 454
SettingsRecord (Java class), 460
SettingsRecord() (Java constructor), 460
SettingsService (Java interface), 165
setToAddress(String) (Java method), 139
setToken(String) (Java method), 425, 473
setTooltip(String) (Java method), 186, 222
setTotal(Integer) (Java method), 142
setTracked(boolean) (Java method), 186
setTracking(Tracking) (Java method), 177
setTracking(TrackingDto) (Java method), 212
setTransactionManager(JdoTransactionManager) (Java method), 301
setTrash(List) (Java method), 381
setTrashFlag(Object, Object, boolean) (Java method), 306, 326
setTrashService(TrashService) (Java method), 302
setTrigger(TaskTriggerInformation) (Java method), 518
setTriggerTaskEvents(List) (Java method), 498
setType(FilterType) (Java method), 260
setType(ParameterType) (Java method), 504, 514
setType(String) (Java method), 500, 529
setType(Type) (Java method), 186, 233
setType(TypeDto) (Java method), 225, 239, 246
setTypeClass(Class) (Java method), 207
setTypeClass(String) (Java method), 245
setTypeSettingOptions(List) (Java method), 208
setUIDisplayable(boolean) (Java method), 186
setUIDisplayPosition(Long) (Java method), 186
setUIFilterable(boolean) (Java method), 186
setUiFrameworkService(UIFrameworkService) (Java method), 470
setUpdate(boolean) (Java method), 237
setUploadSize(String) (Java method), 459, 463
setUpSecurityContext() (Java method), 278, 280
setUrl(String) (Java method), 364, 370
setUseGenericParam(boolean) (Java method), 233
setUseGenericParams(Map) (Java method), 196
setUseOriginalFireTimeAfterMisfire(boolean) (Java method), 394
setUserAccess(List) (Java method), 421, 432
setUserLang(HttpServletRequest, HttpServletResponse, String, String, String) (Java method), 469
setUserLocale(HttpServletRequest, HttpServletResponse, Locale) (Java method), 467
setUserName(String) (Java method), 423, 435, 472
setUsername(String) (Java method), 153, 425

setUsernamePasswordAuthenticationFilter(UsernamePasswordAuthenticationFilter) (Java method), 412
setValidation(FieldValidationDto) (Java method), 225
setValidationErrors(Set) (Java method), 518
setValidations(List) (Java method), 186, 207
setValue(Object) (Java method), 239, 246
setValue(String) (Java method), 189, 190, 192, 234, 509
setValues(Map) (Java method), 216, 520
setValueType(Type) (Java method), 208, 210
setVelocityEngine(VelocityEngine) (Java method), 320
setVersion(Long) (Java method), 178
setVersion(String) (Java method), 123, 421, 433
shouldIgnoreThisProperty() (Java method), 280, 283, 287, 288
shutdown() (Java method), 398
SingleResultFromLookupExpectedException (Java class), 257
SingleResultFromLookupExpectedException(String) (Java constructor), 257
SMS (Java field), 83
SPLIT (Java field), 511
SQL_DRIVER (Java field), 113
SQL_PASSWORD (Java field), 113
SQL_QUERY (Java field), 342
SQL_URL (Java field), 113
SQL_USER (Java field), 113
SQLDBConfig (Java class), 118
SQLDBConfig(String, String, String, String) (Java constructor), 118
SqlDBManager (Java interface), 107
SqlQueryExecution (Java interface), 288
STANDARD (Java field), 182
start() (Java method), 94, 359, 373
start(BundleContext) (Java method), 355, 356, 374
STARTING (Java field), 450
STARTS_WITH (Java field), 337
STARTSWITH (Java field), 513
STARTUP (Java field), 465
startup() (Java method), 466, 471
StartupController (Java class), 471
StartupForm (Java class), 473
StartupFormValidator (Java class), 483
StartupFormValidator() (Java constructor), 483
StartupFormValidatorFactory (Java class), 483
StartupManager (Java class), 465
StartupSuggestionsForm (Java class), 476
StartupSuggestionsForm() (Java constructor), 476
State (Java enum), 450
status() (Java method), 472
STATUS_BLOCKED (Java field), 389
STATUS_ERROR (Java field), 389
STATUS_MSG_TIMEOUT (Java field), 111
STATUS_OK (Java field), 389
STATUS_PAUSED (Java field), 389

StatusController (Java class), 471
 StatusMessage (Java class), 81
 StatusMessage() (Java constructor), 81
 StatusMessage(String, String, Level) (Java constructor), 81
 StatusMessage(String, String, Level, DateTime) (Java constructor), 81
 StatusMessagesDataService (Java interface), 86
 StatusMessageService (Java interface), 84
 stop() (Java method), 124
 stop(BundleContext) (Java method), 356, 357, 374
 STOPPING (Java field), 450
 STRING (Java field), 227, 243, 510
 STRING_MAX_LENGTH (Java field), 340
 StringUtil (Java class), 104
 SUBJECT_FIELD (Java field), 538
 submenuBackToNormal(String) (Java method), 364
 SubmenuInfo (Java class), 369
 SubmenuInfo(String) (Java constructor), 370
 subMenuNeedsAttention(String) (Java method), 364
 submitForm(StartupForm) (Java method), 471
 SUBSTRING (Java field), 511
 SUCCESS (Java field), 523
 SUFFIX_REPEATJOBID (Java field), 392
 SUFFIX_RUNONCEJOBID (Java field), 395
 suggestActivemqUrl() (Java method), 480
 SuggestionHelper (Java class), 480
 Sunday (Java field), 98
 supportAnyOperation() (Java method), 237
 supports(Class) (Java method), 408
 supports(ConfigAttribute) (Java method), 408
 supports(String) (Java method), 89, 90
 supportsAnyOperation() (Java method), 203
 supportsAnyRestOperations() (Java method), 177
 SYMBOLIC_NAME_PREFIX (Java field), 333
 SYMBOLIC_NAME_SUFFIX (Java field), 335
 SystemIdentityProvider (Java class), 94

T

Task (Java class), 516
 Task() (Java constructor), 516
 Task(String, TaskTriggerInformation, List) (Java constructor), 516
 Task(String, TaskTriggerInformation, List, TaskConfig, boolean, boolean) (Java constructor), 516
 TaskAction (Java annotation), 484
 TaskActionExecutor (Java class), 543
 TaskActionExecutor(TaskService, TaskActivityService, EventRelay) (Java constructor), 543
 TaskActionInformation (Java class), 519
 TaskActionInformation() (Java constructor), 519
 TaskActionInformation(String, String, String, String, String) (Java constructor), 519
 TaskActionInformation(String, String, String, String, String, Map) (Java constructor), 519
 TaskActionInformation(String, String, String, String, String, String, String, Map) (Java constructor), 519
 TaskActionParam (Java annotation), 484
 TaskActivitiesDataService (Java interface), 539
 TaskActivity (Java class), 520
 TaskActivity() (Java constructor), 520
 TaskActivity(String, List, Long, TaskActivityType) (Java constructor), 521
 TaskActivity(String, List, Long, TaskActivityType, String) (Java constructor), 521
 TaskActivity(String, Long, TaskActivityType) (Java constructor), 520
 TaskActivity(String, String, Long, TaskActivityType) (Java constructor), 521
 TaskActivityService (Java interface), 543
 TaskActivityType (Java enum), 522
 TaskAnnotationBeanPostProcessor (Java class), 485
 TaskAnnotationBeanPostProcessor(BundleContext, ChannelService) (Java constructor), 485
 TaskBuilder (Java class), 523
 TaskBuilder() (Java constructor), 523
 TaskChannel (Java annotation), 485
 TaskConfig (Java class), 524
 TaskConfigDeserializer (Java class), 538
 TaskConfigStep (Java class), 526
 TaskContext (Java class), 544
 TaskContext(Task, MotechEvent, TaskActivityService) (Java constructor), 544
 TaskDataProvider (Java class), 526
 TaskDataProvider() (Java constructor), 526
 TaskDataProvider(String, List) (Java constructor), 526
 TaskDataProviderObject (Java class), 528
 TaskDataProviderObject() (Java constructor), 528
 TaskDataProviderObject(String, String, List, List) (Java constructor), 528
 TaskDataProviderService (Java interface), 545
 TaskDeserializer (Java class), 539
 TaskError (Java class), 529
 TaskError() (Java constructor), 529
 TaskError(String, String) (Java constructor), 530
 TaskError(TaskErrorType, String) (Java constructor), 529
 TaskErrorType (Java enum), 530
 TaskEvent (Java class), 531
 TaskEvent() (Java constructor), 531
 TaskEvent(String, String, String) (Java constructor), 531
 TaskEventInformation (Java class), 532
 TaskEventInformation() (Java constructor), 532
 TaskEventInformation(String, String, String, String, String) (Java constructor), 533

- TaskFilterExecutor (Java class), 546
- TaskHandlerException (Java class), 536
- TaskHandlerException(TaskFailureCause, String) (Java constructor), 536
- TaskHandlerException(TaskFailureCause, String, String) (Java constructor), 536
- TaskHandlerException(TaskFailureCause, String, Throwable, String) (Java constructor), 536
- TaskInitializer (Java class), 546
- TaskInitializer(TaskContext) (Java constructor), 546
- TaskNotFoundException (Java class), 537
- TaskNotFoundException(Long) (Java constructor), 537
- TasksDataService (Java interface), 539
- TaskService (Java interface), 547
- TaskTriggerHandler (Java class), 548
- TaskTriggerHandler(TaskService, TaskActivityService, EventListenerRegistryService, EventRelay, SettingsFacade) (Java constructor), 548
- TaskTriggerInformation (Java class), 534
- TaskTriggerInformation() (Java constructor), 534
- TaskTriggerInformation(String, String, String, String, String) (Java constructor), 534
- TaskTriggerInformation(TaskTriggerInformation) (Java constructor), 534
- Tenant (Java class), 94
- Tenant(TenantIdentity) (Java constructor), 95
- TENANT_ID (Java field), 113
- TenantIdentity (Java class), 95
- TenantIdentity() (Java constructor), 95
- TenantIdentity(IdentityProvider) (Java constructor), 95
- TEXTAREA (Java field), 515
- THIS_MONTH (Java field), 261
- THIS_YEAR (Java field), 261
- Thursday (Java field), 98
- Time (Java class), 98
- TIME (Java field), 243, 515
- Time() (Java constructor), 98
- time(DateTime) (Java method), 103
- Time(int, int) (Java constructor), 98
- Time(Integer, Integer) (Java constructor), 99
- Time(LocalTime) (Java constructor), 99
- Time(String) (Java constructor), 99
- timeStr() (Java method), 100
- TimeTypeConverter (Java class), 273
- TimeUnit (Java enum), 166
- timeZone() (Java method), 101, 105
- toClassPath(Class) (Java method), 264
- toClassPath(String) (Java method), 264
- toClassPath(String, boolean) (Java method), 265
- toDatastoreType(Time) (Java method), 273
- TODAY (Java field), 261
- today() (Java method), 101, 103, 105, 106
- toDto() (Java method), 167, 177, 180, 186, 189, 191, 192, 196, 203, 204, 207
- toEnumSet(Class, Set) (Java method), 91
- toEnumSet(Class, String) (Java method), 91
- toGenericParam(Class) (Java method), 265
- toGenericParam(String) (Java method), 265
- toJSON() (Java method), 89, 90
- TOLOWER (Java field), 511
- toMemberType(String) (Java method), 274
- tomorrow() (Java method), 103
- toMotechEvent() (Java method), 498
- toRange(Object, String) (Java method), 355
- toResource() (Java method), 116
- toSet(Object, String) (Java method), 355
- toString() (Java method), 100, 112, 114, 116, 117, 123, 136, 140, 142, 147, 164, 169, 210, 212, 213, 221, 222, 225, 227, 228, 231, 235, 237, 239, 241, 245, 246, 350, 375, 386, 392, 394, 396, 417, 423, 428, 430, 460, 477, 487, 488, 490–492, 494, 496, 498, 501, 502, 504, 505, 510, 514, 519, 520, 522, 525, 526, 528–530, 532, 534, 535
- toString(Set) (Java method), 91
- toStringSet(Set) (Java method), 91
- TOUPPER (Java field), 511
- TRACE (Java field), 413
- Tracking (Java class), 203
- Tracking() (Java constructor), 203
- Tracking(Entity) (Java constructor), 203
- TrackingDto (Java class), 240
- TransactionalMotechDataService (Java class), 313
- transform(MotechEvent) (Java method), 153
- TRASH (Java field), 161, 182
- TrashClassNotFoundException (Java class), 257
- TrashClassNotFoundException(String) (Java constructor), 257
- trashFlag(Class) (Java method), 327
- TrashService (Java interface), 313
- TrashServiceImpl (Java class), 327
- TRIGGER_PREFIX (Java field), 506
- TriggerEvent (Java class), 534
- TriggerEvent() (Java constructor), 535
- TriggerEvent(String, String, String, List) (Java constructor), 535
- TriggerEvent(TriggerEventRequest) (Java constructor), 535
- TriggerEventRequest (Java class), 491
- TriggerEventRequest(String, String, String, List) (Java constructor), 491
- TriggerHandler (Java interface), 549
- TriggerNotFoundException (Java class), 537
- TriggerNotFoundException(String) (Java constructor), 537
- trimTrashHistorySuffix(String) (Java method), 330
- TRUE (Java field), 342
- Tuesday (Java field), 98

- Type (Java class), 205
 - Type (Java enum), 233
 - Type() (Java constructor), 205
 - Type(Class) (Java constructor), 205
 - Type(String, String, Class) (Java constructor), 205
 - TYPE_CLASS (Java field), 214
 - TypeAlreadyExistsException (Java class), 257
 - TypeAlreadyExistsException() (Java constructor), 257
 - TypeDto (Java class), 241
 - TypeDto() (Java constructor), 243
 - TypeDto(Long, String, String, String, String) (Java constructor), 243
 - TypeDto(String, String, String, String) (Java constructor), 243
 - TypeHelper (Java class), 353
 - TypeNotFoundException (Java class), 257
 - TypeNotFoundException(String) (Java constructor), 258
 - TypeService (Java interface), 315
 - TypeServiceImpl (Java class), 323
 - TypeSetting (Java class), 207
 - TypeSetting() (Java constructor), 207
 - TypeSetting(String) (Java constructor), 207
 - TypeSettingOption (Java class), 208
 - TypeSettingOption(String) (Java constructor), 208
 - TypeValidation (Java class), 209
 - TypeValidation() (Java constructor), 209
 - TypeValidation(String, Type) (Java constructor), 209
 - TypeValidationAlreadyExistsException (Java class), 258
 - TypeValidationAlreadyExistsException() (Java constructor), 258
- ## U
- UI (Java field), 117
 - UIDisplayable (Java annotation), 156
 - UIFilterable (Java annotation), 156
 - UIFrameworkService (Java interface), 371
 - UiHttpContext (Java class), 378
 - UiHttpContext(HttpContext) (Java constructor), 378
 - UIServiceTracker (Java class), 372
 - UIServiceTracker(BundleContext, ModuleRegistrationData) (Java constructor), 372
 - UIServiceTracker(BundleContextWrapper, ModuleRegistrationData) (Java constructor), 372
 - UIServiceTrackers (Java class), 373
 - unbind(Object, Map) (Java method), 550
 - underscore() (Java method), 377
 - UNICODE (Java field), 515
 - UNINSTALLED (Java field), 450
 - UNKNOWN (Java field), 161, 166, 450, 512, 515
 - unregister() (Java method), 359
 - unregisterEnhancedData(String) (Java method), 267
 - unregisterModule(String) (Java method), 372
 - unregisterProperties(String) (Java method), 456
 - unscheduleAllJobs(String) (Java method), 401, 406
 - unscheduleJob(JobId) (Java method), 402, 406
 - unscheduleJob(String, String) (Java method), 402, 406
 - unscheduleJobs(String) (Java method), 398, 404
 - unscheduleRepeatingJob() (Java method), 310, 323
 - unscheduleRepeatingJob(String, String) (Java method), 402, 406
 - unscheduleRunOnceJob(String, String) (Java method), 402, 406
 - unsetPrimary() (Java method), 271
 - unwrap() (Java method), 281, 283, 287, 288
 - UPDATE (Java field), 201, 332
 - update(EntityDraft) (Java method), 291
 - update(FieldDto) (Java method), 187
 - update(InputStream) (Java method), 297, 298
 - update(LookupDto, List) (Java method), 196
 - update(MetadataDto) (Java method), 189
 - update(MotechRole) (Java method), 436
 - update(MotechUser) (Java method), 438
 - update>PasswordRecovery) (Java method), 439
 - update(RestOptionsDto) (Java method), 203
 - update(T) (Java method), 296, 302, 312
 - updateAdvancedSetting(AdvancedSettingsDto) (Java method), 177
 - updateComboboxValues(Long, Map) (Java method), 305, 319
 - updateConfigLocation(String) (Java method), 130
 - updateDraft(Long) (Java method), 305, 319
 - updateFields(Long, Map) (Java method), 160
 - updateFileMonitor() (Java method), 124
 - updateFromDraft(EntityDraft) (Java method), 177
 - updateFromProperties(Properties) (Java method), 460, 463
 - updateFromTransient(T) (Java method), 302, 312
 - updatePropertiesAfterReinstallation(String, String, String, String, Properties, Properties) (Java method), 131
 - updateRecordFields(Object, Object) (Java method), 329
 - updateRestOptions(Long, RestOptionsDto) (Java method), 305, 319
 - updateRestOptions(RestOptionsDto) (Java method), 177
 - updateRole(RoleDto) (Java method), 443
 - updateScheduledJob(MotechEvent) (Java method), 402, 406
 - updateSecurityConfiguration(SecurityConfigDto) (Java method), 443
 - updateSettings(String, String, Properties) (Java method), 460, 464
 - updateSqlProperties() (Java method), 108
 - updateUserDetailsWithoutPassword(UserDto) (Java method), 445
 - updateUserDetailsWithPassword(UserDto) (Java method), 445
 - UPLOAD_SIZE (Java field), 111
 - URL_PATTERN (Java field), 460

URLENCODE (Java field), 512
useFilter(Query, Filter) (Java method), 286
useFilter(Query, List) (Java method), 286
useFilter(Query, List, InstanceSecurityRestriction) (Java method), 287
useFilter(Query, String[], Object[]) (Java method), 286
useFilter(Query, String[], Object[], InstanceSecurityRestriction) (Java method), 286
useFilterFromPattern(Query, String, List) (Java method), 287
USER_ACCESS_PREFIX (Java field), 415
USER_ADMIN_ROLE (Java field), 416
UserContextService (Java interface), 447
UserDto (Java class), 433
UserDto() (Java constructor), 433
UserDto(MotechUser) (Java constructor), 433
UserInfo (Java class), 477
UserInfo(String, boolean, String) (Java constructor), 477
USERNAME_PASSWORD (Java field), 415
UsernameValueGenerator (Java class), 274
UserNotFoundException (Java class), 427
UserNotFoundException() (Java constructor), 427
UserNotFoundException(String) (Java constructor), 427
UserRegistrationValidator (Java class), 484
UserRegistrationValidator(PersistedUserValidator, OpenIdUserValidator) (Java constructor), 484
UserRoleNames (Java class), 415
USERS (Java field), 352
Util (Java class), 340

V

validate(ResetForm) (Java method), 483
validate(StartupForm, ConfigSource) (Java method), 483
validate(StartupForm, List, ConfigSource) (Java method), 480–482, 484
validateCredentials() (Java method), 302
validateCredentials(T) (Java method), 302
validateEmptyOrWhitespace(Errors, String, String) (Java method), 484
validateNoJavaKeyword(String) (Java method), 355
validateToken(String) (Java method), 446
validateTokenAndLoginUser(String, HttpServletRequest, HttpServletResponse) (Java method), 446
ValidationCriterionDto (Java class), 245
ValidationCriterionDto() (Java constructor), 245
ValidationCriterionDto(String, TypeDto) (Java constructor), 245
ValidationCriterionDto(String, TypeDto, Object, boolean) (Java constructor), 245
ValidationException (Java class), 537
ValidationException(String, Set) (Java constructor), 537
ValidationUtil (Java class), 355
ValidationUtils (Java class), 484
VALUE (Java field), 214, 233, 332

value() (Java method), 392
VALUE_GENERATOR (Java field), 342
valueAsString() (Java method), 246
ValueGetter (Java class), 328
ValueGetter(BasePersistenceService, BundleContext) (Java constructor), 328
valueOf(String) (Java method), 100, 117, 457
valuesForQuery() (Java method), 260
VERSION (Java field), 531
VIEW_BASIC_EMAIL_LOGS_PERMISSION (Java field), 413
VIEW_DETAILED_EMAIL_LOGS_PERMISSION (Java field), 413
VIEW_SECURITY (Java field), 414
vote(Authentication, Object, Collection) (Java method), 408

W

waitForContext(int) (Java method), 88, 365
WallTime (Java class), 106
WallTime(String) (Java constructor), 106
WallTimeUnit (Java enum), 107
WARN (Java field), 83
warn(String, String) (Java method), 86
warn(String, String, DateTime) (Java method), 86
WARNING (Java field), 523
WebBundleUtil (Java class), 383
WebUIBundleActivator (Java class), 373
Wednesday (Java field), 98
Week (Java field), 107
WeeklyCronJobExpressionBuilder (Java class), 384
WeeklyCronJobExpressionBuilder(DayOfWeek) (Java constructor), 384
WeeklyCronJobExpressionBuilder(int) (Java constructor), 384
WEEKS (Java field), 166
withDeliveryStatuses(DeliveryStatus) (Java method), 133
withDeliveryStatuses(Set) (Java method), 133
withDescription(String) (Java method), 524
withFromAddress(String) (Java method), 133
withId(Long) (Java method), 524
withMessage(String) (Java method), 133
withMessageTime(DateTime) (Java method), 134
withMessageTimeRange(Range) (Java method), 134
withName(String) (Java method), 524
withQueryParams(QueryParams) (Java method), 134
withRepeatIntervalInDays(int) (Java method), 384
withServiceName(String) (Java method), 541
withSubject(String) (Java method), 134
withTaskConfig(TaskConfig) (Java method), 524
withTime(Time) (Java method), 384
withToAddress(String) (Java method), 134
withTrigger(TaskTriggerInformation) (Java method), 524
WRITABLE (Java field), 116

Y

YEARS (Java field), [166](#)

YES (Java field), [261](#)