

# Envirosense API

---

# Package **envirosense.configuration**

This package holds classes for configuring the application.

## envirosense.configuration Class ApplicationConfiguration

```
java.lang.Object
├── WebMvcConfigurerAdapter
│   └── envirosense.configuration.ApplicationConfiguration
```

```
public class ApplicationConfiguration
    extends WebMvcConfigurerAdapter
```

### Fields

#### PROPERTY\_NAME\_DATABASE\_DRIVER

```
private static final java.lang.String PROPERTY_NAME_DATABASE_DRIVER
```

Constant value: **db.driver**

#### PROPERTY\_NAME\_DATABASE\_URL

```
private static final java.lang.String PROPERTY_NAME_DATABASE_URL
```

Constant value: **db.url**

#### PROPERTY\_NAME\_DATABASE\_USERNAME

```
private static final java.lang.String PROPERTY_NAME_DATABASE_USERNAME
```

Constant value: **db.username**

#### PROPERTY\_NAME\_DATABASE\_PASSWORD

```
private static final java.lang.String PROPERTY_NAME_DATABASE_PASSWORD
```

Constant value: **db.password**

#### PROPERTY\_NAME\_HIBERNATE\_DIALECT

```
private static final java.lang.String PROPERTY_NAME_HIBERNATE_DIALECT
```

Constant value: **hibernate.dialect**

#### PROPERTY\_NAME\_HIBERNATE\_SHOW\_SQL

```
private static final java.lang.String PROPERTY_NAME_HIBERNATE_SHOW_SQL
```

Constant value: **hibernate.show\_sql**

---

## PROPERTY\_NAME\_HIBERNATE\_HBML2DDL

```
private static final java.lang.String PROPERTY_NAME_HIBERNATE_HBML2DDL
```

Constant value: `hibernate.hbm2ddl.auto`

---

## PROPERTY\_NAME\_ENTITYMANAGER\_PACKAGES\_TO\_SCAN

```
private static final java.lang.String PROPERTY_NAME_ENTITYMANAGER_PACKAGES_TO_SCAN
```

Constant value: `entityManager.packages.to.scan`

---

## env

```
private Environment env
```

## Constructors

### ApplicationConfiguration

```
public ApplicationConfiguration()
```

## Methods

### addResourceHandlers

```
public void addResourceHandlers(ResourceHandlerRegistry registry)
```

---

### configureDefaultServletHandling

```
public void configureDefaultServletHandling(DefaultServletHandlerConfigurer configurer)
```

---

### addViewControllers

```
public void addViewControllers(ViewControllerRegistry registry)
```

---

### dataSource

```
public javax.sql.DataSource dataSource()
```

---

(continued on next page)

(continued from last page)

## **entityManagerFactory**

```
public LocalContainerEntityManagerFactoryBean entityManagerFactory()
```

---

## **transactionManager**

```
public JpaTransactionManager transactionManager()
```

---

## **viewResolver**

```
public InternalResourceViewResolver viewResolver()
```

---

## **messageSource**

```
public ResourceBundleMessageSource messageSource()
```

---

## **propertyPlaceholderConfigurer**

```
public static PropertySourcesPlaceholderConfigurer propertyPlaceholderConfigurer()
```

---

## **hibernateProperties**

```
private java.util.Properties hibernateProperties()
```

## envirosense.configuration

### Class EmailConfiguration

java.lang.Object

└─envirosense.configuration.EmailConfiguration

public class **EmailConfiguration**  
extends java.lang.Object

#### Fields

##### PROPERTY\_NAME\_MAIL\_PROTOCOL

public static final java.lang.String PROPERTY\_NAME\_MAIL\_PROTOCOL

Constant value: **mail.transport.protocol**

##### PROPERTY\_NAME\_MAIL\_HOST

public static final java.lang.String PROPERTY\_NAME\_MAIL\_HOST

Constant value: **mail.smtps.host**

##### PROPERTY\_NAME\_MAIL\_PORT

public static final java.lang.String PROPERTY\_NAME\_MAIL\_PORT

Constant value: **mail.smtps.port**

##### PROPERTY\_NAME\_MAIL\_AUTH

public static final java.lang.String PROPERTY\_NAME\_MAIL\_AUTH

Constant value: **mail.smtps.auth**

##### PROPERTY\_NAME\_MAIL\_USERNAME

public static final java.lang.String PROPERTY\_NAME\_MAIL\_USERNAME

Constant value: **mail.username**

##### PROPERTY\_NAME\_MAIL\_PASSWORD

public static final java.lang.String PROPERTY\_NAME\_MAIL\_PASSWORD

Constant value: **mail.password**

---

## PROPERTY\_NAME\_MAIL\_FROM

```
public static final java.lang.String PROPERTY_NAME_MAIL_FROM
```

Constant value: **mail.from**

---

## env

```
private Environment env
```

## Constructors

### EmailConfiguration

```
public EmailConfiguration()
```

## Methods

### javaMailService

```
public JavaMailSender javaMailService()
```

## envirosense.configuration Class Initializer

```
java.lang.Object
└--envirosense.configuration.Initializer
```

```
public class Initializer
    extends java.lang.Object
```

### Fields

#### DISPATCHER\_SERVLET\_NAME

```
private static final java.lang.String DISPATCHER_SERVLET_NAME
```

Constant value: **dispatcher**

### Constructors

#### Initializer

```
public Initializer()
```

### Methods

#### onStartup

```
public void onStartup(ServletContext servletContext)
    throws ServletException
```



## envirosense.configuration Class SecurityConfiguration

```
java.lang.Object
  |
  +--WebSecurityConfigurerAdapter
      |
      +--envirosense.configuration.SecurityConfiguration
```

```
public class SecurityConfiguration
    extends WebSecurityConfigurerAdapter
```

Security configuration

### Fields

#### userDetailsAuth

```
private envirosense.service.UserDetailsAuthService userDetailsAuth
```

### Constructors

#### SecurityConfiguration

```
public SecurityConfiguration()
```

### Methods

#### md5PasswordEncoder

```
public Md5PasswordEncoder md5PasswordEncoder()
```

#### configureGlobalSecurity

```
public void configureGlobalSecurity(AuthenticationManagerBuilder auth)
    throws java.lang.Exception
```

#### authenticationProvider

```
public DaoAuthenticationProvider authenticationProvider()
```

(continued from last page)

**configure**

```
protected void configure(HttpSecurity http)
    throws java.lang.Exception
```

---

**configure**

```
public void configure(WebSecurity web)
    throws java.lang.Exception
```

## envirosense.configuration

### Class SecurityWebApplicationInitializer

java.lang.Object

└ AbstractSecurityWebApplicationInitializer

└─envirosense.configuration.SecurityWebApplicationInitializer

---

public class **SecurityWebApplicationInitializer**  
extends AbstractSecurityWebApplicationInitializer

Enables spring security in the application container

---

## Constructors

### SecurityWebApplicationInitializer

public **SecurityWebApplicationInitializer**()

## envirosense.configuration Class SlackConfiguration

```
java.lang.Object
  |
  +--envirosense.configuration.SlackConfiguration
```

```
public class SlackConfiguration
  extends java.lang.Object
```

### Fields

#### PROPERTY\_NAME\_SLACK\_WEBHOOK\_URL

```
public static final java.lang.String PROPERTY_NAME_SLACK_WEBHOOK_URL
```

Constant value: **slack.hookurl**

#### env

```
private Environment env
```

### Constructors

#### SlackConfiguration

```
public SlackConfiguration()
```

### Methods

#### slackApi

```
public SlackApi slackApi()
```

---

# Package **envirosense.controller**

Defines UI controllers

## envirosense.controller Class EventController

java.lang.Object

└-envirosense.controller.EventController

```
public class EventController
    extends java.lang.Object
```

### Fields

#### eventService

envirosense.service.EventService **eventService**

### Constructors

#### EventController

```
public EventController()
```

### Methods

#### adminAllEvents

```
public ModelAndView adminAllEvents()
```

Returns the events view for the admin

**Returns:**

Events view for the admin

#### userAllEvents

```
public ModelAndView userAllEvents(java.security.Principal principal)
```

Returns the events view for the logged in user

**Parameters:**

principal - The logged in user

**Returns:**

Events view

## envirosense.controller

### Class HomeController

java.lang.Object

└--envirosense.controller.HomeController

---

public class **HomeController**  
extends java.lang.Object

Controllers for mapping the home page of the application

---

## Constructors

### HomeController

public **HomeController**()

## Methods

### homePage

public ModelAndView **homePage**()

Directs to the home page view using the specified mapping

**Returns:**

the home page view

## envirosense.controller

### Class LoginController

java.lang.Object

└--envirosense.controller.LoginController

---

public class **LoginController**  
extends java.lang.Object

Controller to map the login screen for the application

---

## Constructors

### LoginController

public **LoginController**()

## Methods

### login

public ModelAndView **login**()

Directs to the login view using the /login mapping

**Returns:**

the login view



## envirosense.controller

### Class ReportViewController

```
java.lang.Object
|
+--envirosense.controller.ReportViewController
```

---

```
public class ReportViewController
extends java.lang.Object
```

---

## Constructors

### ReportViewController

```
public ReportViewController()
```

## Methods

### reportPage

```
public ModelAndView reportPage()
```

Returns the reports view

**Returns:**

Reports view

## envirosense.controller

### Class UserController

java.lang.Object

└─envirosense.controller.UserController

```
public class UserController  
extends java.lang.Object
```

Administrative CRUD Controllers for managing users

#### Fields

##### userService

envirosense.service.UserService **userService**

#### Constructors

##### UserController

```
public UserController()
```

#### Methods

##### saveUser

```
public ModelAndView saveUser(UserDTO userDTO)
```

Save a user to the database

###### Parameters:

user - the user to be saved

###### Returns:

the user management page

##### updateUser

```
public ModelAndView updateUser(UserDTO user)
```

Update a user to the database

###### Parameters:

user - the user to be updated

###### Returns:

the user management page

## saveUsers

```
public ModelAndView saveUsers(java.util.List users)
```

Save or update a list of two or more users to the database

**Parameters:**

`users` - the list of users to be saved or updated

**Returns:**

the user management page

---

## deleteUser

```
public ModelAndView deleteUser(User user)
```

Delete a user from the database

**Parameters:**

`user` - the user to be deleted from the database

**Returns:**

the user management page

---

## resetPassword

```
public ModelAndView resetPassword(User user,  
    java.lang.String newPassword)
```

Reset the password for a user. The password should be encrypted or hashed on the client side before the POST request. The `UserService` will handle the hashed or encrypted password

**Parameters:**

`user` - the user whose account password will be reset

`newPassword` - the encrypted or hashed new password

**Returns:**

the user management page

---

## getUserByEmail

```
public ModelAndView getUserByEmail(java.lang.String email)
```

Get the page for the specified user by email

**Parameters:**

`email` - the unique email address of a user account

**Returns:**

the user management page for the specified user

---

## getUserByFirstName

```
public ModelAndView getUserByFirstName(java.lang.String name)
```

Get the page for the set of users containing the specified first name

**Parameters:**

(continued from last page)

name - the first name of the users to search

**Returns:**

the user management page for the specified users

---

## **getUserByLastName**

```
public ModelAndView getUserByLastName( java.lang.String name)
```

Get the page for the set of users containing the specified last name

**Parameters:**

name - the surname of the users to search

**Returns:**

the user management page for the specified users

---

## **getActiveUsers**

```
public ModelAndView getActiveUsers( )
```

Get the page for the set for users with an active status

**Returns:**

the user management page for the active users

---

## **getInactiveUsers**

```
public ModelAndView getInactiveUsers( )
```

Get the page for the set for users with an inactive status

**Returns:**

the user management page for the inactive users

---

## **getAllUsers**

```
public ModelAndView getAllUsers( )
```

Controller for getting and showing all users

**Returns:**

All users in the system

---

---

**Package**

# **envirosense.controller.api**

Defines various controllers for the API HTTP end-points

## envirosense.controller.api Class BluetoothBeaconController

java.lang.Object

└─envirosense.controller.api.BluetoothBeaconController

```
public class BluetoothBeaconController  
extends java.lang.Object
```

### Fields

#### beaconRepository

envirosense.repository.BluetoothBeaconRepository **beaconRepository**

### Constructors

#### BluetoothBeaconController

```
public BluetoothBeaconController()
```

### Methods

#### all

```
public <any> all()
```

Returns all beacons in the database

##### Returns:

All beacons in the database

#### findById

```
public <any> findById(java.lang.String id)
```

Returns the beacon with the specified ID

##### Parameters:

id - The ID of the beacon to return

##### Returns:

The beacon with the specified ID

#### findByUser

```
public <any> findByUser(java.lang.String email)
```

(continued from last page)

Returns all the beacons whose owner has the email specified

**Parameters:**

`email` - The email of the user who owns the beacons to be returned

**Returns:**

All the beacons whose owner has the email specified

## envirosense.controller.api Class EventApiController

```
java.lang.Object
└--envirosense.controller.api.EventApiController
```

```
public class EventApiController
    extends java.lang.Object
```

Controller for managing events via HTTP calls

### Fields

#### eventService

```
envirosense.service.EventService eventService
```

### Constructors

#### EventApiController

```
public EventApiController()
```

### Methods

#### enableEvent

```
public <any> enableEvent(long id)
```

Enables the event with specified ID

**Parameters:**

id - The ID of the event that is to be enabled

**Returns:**

An HTTP response status

#### diableEvent

```
public <any> diableEvent(long id)
```

Disables the event with specified ID

**Parameters:**

id - The ID of the event that is to be disabled

**Returns:**

An HTTP response status



## newEvent

```
public <any> newEvent(Event event,  
    BindingResult binding)
```

Creates a new event

### Parameters:

event - The new event

binding - A binding result

### Returns:

An HTTP response status

---

## deleteEvent

```
public <any> deleteEvent(long id)
```

Deletes the event with the specified ID

### Parameters:

id - The ID of the event to delete

### Returns:

An HTTP response status

---

## getAll

```
public <any> getAll()
```

Retrieves and returns all events in the system

### Returns:

All events in the system and an HTTP status code

---

## getActive

```
public <any> getActive()
```

Retrieves and returns all active events in the system

### Returns:

All active events in the system and an HTTP status code

---

## getInactive

```
public <any> getInactive()
```

Retrieves and returns all inactive events in the system

### Returns:

All inactive events in the system and an HTTP status code

---

## envirosense.controller.api Class ReportController

```
java.lang.Object
├──envirosense.controller.api.ReportController
```

```
public class ReportController
extends java.lang.Object
```

REST Controllers for generating reports from the sensor data in the database

### Fields

#### sensorDataService

```
envirosense.service.SensorDataService sensorDataService
```

### Constructors

#### ReportController

```
public ReportController()
```

### Methods

#### getDataByRoomId

```
public <any> getDataByRoomId(long roomId,
    java.sql.Timestamp startDate,
    java.sql.Timestamp endDate)
```

REST Controller to return a dataset of all sensor data in a given room for the given time/date range

##### Parameters:

roomId - The ID of the room  
startDate - The start date of the data to return  
endDate - The end date of the data to return

##### Returns:

A list of SensorData objects for all sensors in the specified room for the given time/date range

#### getDataBySensorType

```
public <any> getDataBySensorType(SensorType sensorType,
    java.sql.Timestamp startDate,
    java.sql.Timestamp endDate)
```

REST Controller to return a dataset of all sensor data of a given sensor type for the given time/date range in all rooms

##### Parameters:

(continued from last page)

sensorType - The sensor type to retrieve data from  
startDate - the start date of the data to return  
endDate - the end date of the data to return

**Returns:**

A list of SensorData objects for all sensors of the specified type in all rooms for the given time/date range

---

## getDataBySensorId

```
public <any> getDataBySensorId(long sensorId,  
    java.sql.Timestamp startDate,  
    java.sql.Timestamp endDate)
```

REST Controller to return a dataset of sensor data from one specified sensor for the given time/date range

**Parameters:**

sensorId - The particular sensor to return data from  
startDate - the start date of the data to return  
endDate - the end date of the data to return

**Returns:**

A list of SensorData objects for the given sensor for the given time/date range

---

## getDataBySensorTypeAndRoomId

```
public <any> getDataBySensorTypeAndRoomId(long roomId,  
    SensorType sensorType,  
    java.sql.Timestamp startDate,  
    java.sql.Timestamp endDate)
```

REST Controller to return a dataset of the specified sensorType data in a given room for the given time/date range

**Parameters:**

roomId - The ID of the room  
sensorType - The sensorType to retrieve data from  
startDate - The start date of the data to return  
endDate - The end date of the data to return

**Returns:**

A list of SensorData objects for the specified sensor type in the specified room for the given time/date range

## envirosense.controller.api Class RoomApiController

```
java.lang.Object
└--envirosense.controller.api.RoomApiController
```

```
public class RoomApiController
    extends java.lang.Object
```

### Fields

#### roomService

```
envirosense.service.RoomService roomService
```

### Constructors

#### RoomApiController

```
public RoomApiController()
```

### Methods

#### getAllRooms

```
public <any> getAllRooms()
```

Retrieves and returns all rooms in the system

**Returns:**

All rooms and an HTTP status code

#### getOne

```
public <any> getOne(long id)
```

Retrieves and returns the room with the specified ID

**Parameters:**

id - The ID of the room to retrieve

**Returns:**

The retrieved room and an HTTP status code

## envirosense.controller.api Class SensorApiController

java.lang.Object

└─envirosense.controller.api.SensorApiController

public class **SensorApiController**  
extends java.lang.Object

### Fields

#### sensorService

envirosense.service.SensorService **sensorService**

### Constructors

#### SensorApiController

public **SensorApiController**()

### Methods

#### allSensors

public <any> **allSensors**()

Retrieves and returns all sensors in the system

**Returns:**

All sensors and an HTTP status code

#### sensor

public <any> **sensor**(long id)

Retrieves and returns the sensor with specified ID

**Parameters:**

id - The ID of the sensor to retrieve

**Returns:**

Retrieved sensor and an HTTP status code

#### sensor

public <any> **sensor**(java.lang.String type)

(continued from last page)

Retrieves and returns the sensor with specified type

**Parameters:**

`type` - The type of the sensor to retrieve

**Returns:**

Retrieved sensor and an HTTP status code

## envirosense.controller.api Class SensorDataController

```
java.lang.Object
└--envirosense.controller.api.SensorDataController
```

```
public class SensorDataController
    extends java.lang.Object
```

### Fields

#### dataService

```
envirosense.service.SensorDataService dataService
```

#### eventHandler

```
envirosense.service.EventHandler eventHandler
```

### Constructors

#### SensorDataController

```
public SensorDataController()
```

### Methods

#### newData

```
public <any> newData(java.util.List data,
    BindingResult result)
```

##### Parameters:

data - The body of this request (must be a list of sensor data)  
result - The result of binding the request body to the parameter type

##### Returns:

ResponseEntity with HTTP status of the request

#### getLatestByRoom

```
public <any> getLatestByRoom(long roomId)
```

Controller for retrieving data read from sensors in the specified room

(continued from last page)

**Parameters:**

`roomId` - The ID of the room the information from

**Returns:**

A `ResponseEntity` containing the retrieved data and an HTTP status code

---

## **getLatestBySensorType**

```
public <any> getLatestBySensorType(SensorType sensorType)
```

Controller for retrieving data read from sensors of the specified type

**Parameters:**

`sensorType` - The type of the sensor

**Returns:**

Data read by sensors with the specified type



---

## Package

# envirosense.model

Defines classes that model domain entities in the application

## envirosense.model

# Class BluetoothBeacon

java.lang.Object

└─envirosense.model.BluetoothBeacon

### All Implemented Interfaces:

java.io.Serializable

---

public class **BluetoothBeacon**  
extends java.lang.Object  
implements java.io.Serializable

---

## Fields

### serialVersionUID

private static final long **serialVersionUID**

Constant value: 1

---

### id

private java.lang.String **id**

---

### user

private java.lang.String **user**

---

## Constructors

### BluetoothBeacon

```
public BluetoothBeacon(java.lang.String id,  
                        java.lang.String user)
```

---

### BluetoothBeacon

```
public BluetoothBeacon()
```

---

## Methods

---

(continued from last page)

**getId**

```
public java.lang.String getId()
```

---

**getUser**

```
public java.lang.String getUser()
```

---

**setId**

```
public void setId(java.lang.String id)
```

---

**setUser**

```
public void setUser(java.lang.String user)
```

---

**hashCode**

```
public int hashCode()
```

---

**equals**

```
public boolean equals(java.lang.Object obj)
```

---

## envirosense.model Class Condition

java.lang.Object  
└─envirosense.model.Condition

### All Implemented Interfaces:

java.io.Serializable

---

```
public class Condition
  extends java.lang.Object
  implements java.io.Serializable
```

---

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

---

### id

```
private long id
```

---

### sensor

```
private envirosense.model.Sensor sensor
```

---

### value

```
private java.lang.String value
```

---

### modifier

```
private envirosense.model.ConditionModifier modifier
```

---

### conditionTime

```
private envirosense.model.ConditionTime conditionTime
```

(continued from last page)

## Constructors

### Condition

```
public Condition(long id,  
                 Sensor sensor,  
                 java.lang.String value,  
                 ConditionModifier modifier,  
                 ConditionTime conditionTime)
```

### Condition

```
public Condition()
```

## Methods

### getId

```
public long getId()
```

### getSensor

```
public Sensor getSensor()
```

### getValue

```
public java.lang.String getValue()
```

### getModifier

```
public ConditionModifier getModifier()
```

### getConditionTime

```
public ConditionTime getConditionTime()
```

### setId

```
public void setId(long id)
```

(continued from last page)

---

## setSensor

```
public void setSensor(Sensor sensor)
```

---

## setValue

```
public void setValue(java.lang.String value)
```

---

## setModifier

```
public void setModifier(ConditionModifier modifier)
```

---

## setWeeklyOccurrences

```
public void setWeeklyOccurrences(ConditionTime conditionTime)
```

---

## evaluate

```
public boolean evaluate(java.lang.Object data,  
    java.util.Calendar calendar)
```

This method takes an object that holds some sensor data and a calendar object, evaluates this condition with the provided arguments and returns a boolean.

### Parameters:

`data` - The sensor data to compare with this condition's value.  
`calendar` - The date that the data specified was read

### Returns:

`true` If a comparison of the data and the date (and time) matched, `false` if otherwise.

---

## dayInCondition

```
private boolean dayInCondition(int dayOfWeek)
```

Takes an integer representing the day of the week returned from `Calendar.get` and compares it with the day of the week in this condition's date.

### Parameters:

`dayOfWeek` - The day of the week to check

### Returns:

`true` If the specified integer represents the day of the week in this condition's date, `false` if otherwise.

---

## timeEqual

```
private boolean timeEqual(java.util.Calendar now)
```

Takes a calendar object and checks to see if it's greater than or equal to this condition's date, and if the hour and the minute of both times match.

---

(continued from last page)

**Parameters:**

`now` - The calendar object that is to be compared to this condition's date

**Returns:**

`true` if the condition described holds, `false` if otherwise.

## envirosense.model Class ConditionModifier

```
java.lang.Object
  |
  +- java.lang.Enum
        |
        +- envirosense.model.ConditionModifier
```

### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```
public final class ConditionModifier
extends java.lang.Enum
```

A ConditionModifier represents a logical operator used when evaluating conditions

## Fields

### GT

```
public static final envirosense.model.ConditionModifier GT
    Greater
```

### GE

```
public static final envirosense.model.ConditionModifier GE
    Greater or equal
```

### LE

```
public static final envirosense.model.ConditionModifier LE
    Less or equal
```

### LT

```
public static final envirosense.model.ConditionModifier LT
    Less
```

### EQ

```
public static final envirosense.model.ConditionModifier EQ
    Equal
```

### NE

```
public static final envirosense.model.ConditionModifier NE
    Not equal
```



---

## modifier

java.lang.String **modifier**

## Constructors

### ConditionModifier

private **ConditionModifier**(java.lang.String modifier)

## Methods

### values

public static [ConditionModifier\[\]](#) **values**()

---

### valueOf

public static [ConditionModifier](#) **valueOf**(java.lang.String name)

---

### getConditionModifier

public java.lang.String **getConditionModifier**()

Returns the string representation of modifier

**Returns:**

String representation of modifier

## envirosense.model

# Class ConditionTime

java.lang.Object

└--envirosense.model.ConditionTime

### All Implemented Interfaces:

java.io.Serializable

---

```
public class ConditionTime
    extends java.lang.Object
    implements java.io.Serializable
```

ConditionTime holds the date and time for which a condition becomes valid. It also holds the days in the week that the condition must be evaluated. There is also a boolean flag that specifies whether a condition must be evaluated every time from the time it becomes valid. The validity of the ConditionTime depends on the modifier.

### See Also:

[ConditionModifier](#)

---

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: 1

---

### id

```
private long id
```

---

### sunday

```
private boolean sunday
```

---

### monday

```
private boolean monday
```

---

### tuesday

```
private boolean tuesday
```

---

(continued from last page)

---

## wednesday

```
private boolean wednesday
```

---

## thursday

```
private boolean thursday
```

---

## friday

```
private boolean friday
```

---

## saturday

```
private boolean saturday
```

---

## dateTime

```
private java.sql.Timestamp dateTime
```

---

## allHours

```
private boolean allHours
```

---

## timeCheck

```
private envirosense.model.ConditionModifier timeCheck
```

---

# Constructors

## ConditionTime

```
public ConditionTime(boolean sunday,  
                    boolean monday,  
                    boolean tuesday,  
                    boolean wednesday,  
                    boolean thursday,  
                    boolean friday,  
                    boolean saturday,  
                    java.sql.Timestamp dateTime,  
                    boolean allHours,  
                    ConditionModifier timeCheck)
```

Creates a new condition time using the specified parameters

### Parameters:

(continued from last page)

sunday - Sunday check  
monday - Monday check  
tuesday - Tuesday check  
wednesday - Wednesday check  
thursday - Thursday check  
friday - Friday check  
saturday - Saturday check  
dateTime - When to check  
allHours - Check at all hours  
timeCheck - Time modifier

---

## ConditionTime

```
public ConditionTime()
```

## Methods

### getId

```
public long getId()
```

---

### isSunday

```
public boolean isSunday()
```

---

### isMonday

```
public boolean isMonday()
```

---

### isTuesday

```
public boolean isTuesday()
```

---

### isWednesday

```
public boolean isWednesday()
```

---

### isThursday

```
public boolean isThursday()
```

---

### isFriday

```
public boolean isFriday()
```

---

## isSaturday

```
public boolean isSaturday()
```

---

---

## getDateTime

```
public java.sql.Timestamp getDateTime()
```

---

---

## isAllHours

```
public boolean isAllHours()
```

---

---

## getTimeCheck

```
public ConditionModifier getTimeCheck()
```

---

---

## setId

```
public void setId(long id)
```

---

---

## setSunday

```
public void setSunday(boolean sunday)
```

---

---

## setMonday

```
public void setMonday(boolean monday)
```

---

---

## setTuesday

```
public void setTuesday(boolean tuesday)
```

---

---

## setWednesday

```
public void setWednesday(boolean wednesday)
```

---

(continued from last page)

## setThursday

```
public void setThursday(boolean thursday)
```

---

## setFriday

```
public void setFriday(boolean friday)
```

---

## setSaturday

```
public void setSaturday(boolean saturday)
```

---

## setDateTime

```
public void setDateTime(java.sql.Timestamp dateTime)
```

---

## setAllHours

```
public void setAllHours(boolean allHours)
```

---

## setTimeCheck

```
public void setTimeCheck(ConditionModifier timeCheck)
```

---

## envirosense.model Class Door

```
java.lang.Object
└--envirosense.model.Door
```

### All Implemented Interfaces:

```
java.io.Serializable
```

```
public class Door
  extends java.lang.Object
  implements java.io.Serializable
```

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

### sensorId

```
private long sensorId
```

### timestamp

```
private java.sql.Timestamp timestamp
```

### data

```
private boolean data
```

## Constructors

### Door

```
public Door(long sensorId,
            java.sql.Timestamp timestamp,
            boolean data)
```

(continued from last page)

## Door

```
public Door()
```

## Methods

### getSensorId

```
public long getSensorId()
```

### getTimestamp

```
public java.sql.Timestamp getTimestamp()
```

### getData

```
public java.lang.Boolean getData()
```

### setSensorId

```
public void setSensorId(long sensorId)
```

### setTimestamp

```
public void setTimestamp(java.sql.Timestamp timestamp)
```

### setData

```
public void setData(boolean data)
```



## envirosense.model Class Event

java.lang.Object  
└─envirosense.model.Event

### All Implemented Interfaces:

java.io.Serializable

---

public class **Event**  
extends java.lang.Object  
implements java.io.Serializable

---

## Fields

### serialVersionUID

private static final long **serialVersionUID**

Constant value: 1

---

### id

private long **id**

---

### name

private java.lang.String **name**

---

### message

private java.lang.String **message**

---

### useSlack

private boolean **useSlack**

---

### useEmail

private boolean **useEmail**

---

## usePhone

```
private boolean usePhone
```

---

## active

```
private boolean active
```

---

## owners

```
private java.util.Set owners
```

---

## conditions

```
private java.util.Set conditions
```

---

## Constructors

### Event

```
public Event(long id,  
             java.lang.String name,  
             java.lang.String message,  
             boolean useSlack,  
             boolean useEmail,  
             boolean usePhone,  
             boolean active,  
             java.util.Set owners,  
             java.util.Set conditions)
```

---

### Event

```
public Event()
```

---

## Methods

### getId

```
public long getId()
```

---

### getName

```
public java.lang.String getName()
```

---

(continued from last page)

---

## getMessage

```
public java.lang.String getMessage()
```

---

## isUseSlack

```
public boolean isUseSlack()
```

---

## isUseEmail

```
public boolean isUseEmail()
```

---

## isUsePhone

```
public boolean isUsePhone()
```

---

## isActive

```
public boolean isActive()
```

---

## getOwners

```
public java.util.Set getOwners()
```

---

## getConditions

```
public java.util.Set getConditions()
```

---

## setId

```
public void setId(long id)
```

---

## setName

```
public void setName(java.lang.String name)
```

---

(continued from last page)

## **setMessage**

```
public void setMessage(java.lang.String message)
```

---

## **setUseSlack**

```
public void setUseSlack(boolean useSlack)
```

---

## **setUseEmail**

```
public void setUseEmail(boolean useEmail)
```

---

## **setUsePhone**

```
public void setUsePhone(boolean usePhone)
```

---

## **setActive**

```
public void setActive(boolean active)
```

---

## **setOwners**

```
public void setOwners(java.util.Set owners)
```

---

## envirosense.model Class Humidity

```
java.lang.Object
└--envirosense.model.Humidity
```

### All Implemented Interfaces:

```
java.io.Serializable
```

```
public class Humidity
  extends java.lang.Object
  implements java.io.Serializable
```

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

### sensorId

```
private long sensorId
```

### timestamp

```
private java.sql.Timestamp timestamp
```

### data

```
private double data
```

## Constructors

### Humidity

```
public Humidity(long sensorId,
                java.sql.Timestamp timestamp,
                double data)
```

(continued from last page)

## Humidity

```
public Humidity()
```

## Methods

### getSensorId

```
public long getSensorId()
```

### getTimestamp

```
public java.sql.Timestamp getTimestamp()
```

### getData

```
public java.lang.Double getData()
```

### setSensorId

```
public void setSensorId(long sensorId)
```

### setTimestamp

```
public void setTimestamp(java.sql.Timestamp timestamp)
```

### setData

```
public void setData(double data)
```

## envirosense.model

### Class Motion

```
java.lang.Object
|
+--envirosense.model.Motion
```

#### All Implemented Interfaces:

```
java.io.Serializable
```

---

```
public class Motion
  extends java.lang.Object
  implements java.io.Serializable
```

---

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

---

### sensorId

```
private long sensorId
```

---

### timestamp

```
private java.sql.Timestamp timestamp
```

---

### data

```
private boolean data
```

---

## Constructors

### Motion

```
public Motion(long sensorId,
              java.sql.Timestamp timestamp,
              boolean data)
```

---

(continued from last page)

## Motion

```
public Motion()
```

## Methods

### getSensorId

```
public long getSensorId()
```

---

### getTimestamp

```
public java.sql.Timestamp getTimestamp()
```

---

### getData

```
public java.lang.Object getData()
```

---

### setSensorId

```
public void setSensorId(long sensorId)
```

---

### setTimestamp

```
public void setTimestamp(java.sql.Timestamp timestamp)
```

---

### setData

```
public void setData(boolean data)
```



## envirosense.model

### Class ReelyActiveBluetooth

```
java.lang.Object
└--envirosense.model.ReelyActiveBluetooth
```

#### All Implemented Interfaces:

```
java.io.Serializable
```

---

```
public final class ReelyActiveBluetooth
extends java.lang.Object
implements java.io.Serializable
```

---

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

---

### sensorId

```
private long sensorId
```

---

### timestamp

```
java.sql.Timestamp timestamp
```

---

### beacon

```
private envirosense.model.BluetoothBeacon beacon
```

---

### rsi

```
private int rsi
```

---

## Constructors

(continued from last page)

## ReelyActiveBluetooth

```
public ReelyActiveBluetooth(long sensorId,  
                             java.sql.Timestamp timestamp,  
                             BluetoothBeacon beacon,  
                             int rssi)
```

---

## ReelyActiveBluetooth

```
protected ReelyActiveBluetooth()
```

## Methods

### getSensorId

```
public long getSensorId()
```

---

### getTimestamp

```
public java.sql.Timestamp getTimestamp()
```

---

### getBeacon

```
public BluetoothBeacon getBeacon()
```

---

### getRssi

```
public int getRssi()
```

---

### setSensorId

```
public void setSensorId(long sensorId)
```

---

### setTimestamp

```
public void setTimestamp(java.sql.Timestamp timestamp)
```

---

### setBeacon

```
public void setBeacon(BluetoothBeacon beacon)
```

---

## setRssi

```
public void setRssi(int rssi)
```

## envirosense.model Class Report

java.lang.Object

└─**envirosense.model.Report**

public class **Report**  
extends java.lang.Object

### Fields

#### **name**

private final java.lang.String **name**

The name of the report

#### **dataSet**

private final java.util.List **dataSet**

The dataset that makes up this report

#### **starttime**

private final java.sql.Timestamp **starttime**

The earliest timestamp for the retrieved data set

#### **endtime**

private final java.sql.Timestamp **endtime**

The earliest latest for the retrieved data set

#### **filterType**

private final envirosense.model.ReportFilterType **filterType**

The type data in this report

#### **filterValue**

private final java.lang.Object **filterValue**

The filter value for this report

### Constructors

(continued from last page)

## Report

```
public Report(java.lang.String name,  
              java.util.List dataSet,  
              java.sql.Timestamp startDate,  
              java.sql.Timestamp endDate,  
              ReportFilterType filterType,  
              java.lang.Object filterValue)
```

Creates a new Report object using the specified arguments

### Parameters:

name - The name of the report  
dataSet - The dataset that makes up this report  
startDate - The earliest timestamp for the retrieved data set  
endDate - The latest timestamp for the retrieved data set  
filterType - The type data in this report  
filterValue - The filter value for this report

## Methods

### getName

```
public java.lang.String getName()
```

Returns the name of this report

### Returns:

The name of this report

### getDataSet

```
public java.util.List getDataSet()
```

Returns the dataset that makes up this report

### Returns:

The dataset that makes up this report

### getStartTime

```
public java.sql.Timestamp getStartTime()
```

Returns the earliest time for data in this report

### Returns:

The earliest time for data in this report

### getEndTime

```
public java.sql.Timestamp getEndTime()
```

Returns the latest time for data in this report

### Returns:

The latest time for data in this report

(continued from last page)

## **getFilterType**

```
public ReportFilterType getFilterType()
```

---

## **getFilterValue**

```
public java.lang.Object getFilterValue()
```

## envirosense.model

# Class ReportFilterType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- envirosense.model.ReportFilterType

```

### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```

public final class ReportFilterType
extends java.lang.Enum

```

A ReportFilterType determines what type of data will be used to filter data to be retrieved for a report.

## Fields

### Room

```
public static final envirosense.model.ReportFilterType Room
```

Filter data based on a room ID

### Sensor

```
public static final envirosense.model.ReportFilterType Sensor
```

Filter data based on a specific sensor

### SensorType

```
public static final envirosense.model.ReportFilterType SensorType
```

Filter data based on a specific sensor type

## Constructors

### ReportFilterType

```
private ReportFilterType()
```

## Methods

### values

```
public static ReportFilterType\[\] values()
```

(continued from last page)

**valueOf**

```
public static ReportFilterType valueOf(java.lang.String name)
```



## envirosense.model Class Role

java.lang.Object

└─envirosense.model.Role

### All Implemented Interfaces:

java.io.Serializable

---

public class **Role**  
extends java.lang.Object  
implements java.io.Serializable

## Fields

### serialVersionUID

private static final long **serialVersionUID**

Constant value: 1

### role

private java.lang.String **role**

## Constructors

### Role

public **Role**(java.lang.String role)

### Role

public **Role**()

## Methods

### getRole

public java.lang.String **getRole**()

(continued from last page)

## **setRole**

```
public void setRole(java.lang.String role)
```

---

## **hashCode**

```
public int hashCode()
```

---

## **equals**

```
public boolean equals(java.lang.Object obj)
```

## envirosense.model

### Class Room

```
java.lang.Object
└--envirosense.model.Room
```

```
public class Room
extends java.lang.Object
```

This class models a room in the system's environment

#### Fields

##### id

```
private long id
```

The ID of the room object

##### name

```
private java.lang.String name
```

The name of the room object

##### description

```
private java.lang.String description
```

The description of the room object

#### Constructors

##### Room

```
public Room(long id,
            java.lang.String name,
            java.lang.String description)
```

Creates a room object using the parameters passed in

##### Parameters:

id - The ID of the room object to be created

name - The name of the room object to be created

description - The description of the room object to be created

##### Room

```
protected Room()
```

#### Methods

(continued from last page)

## getId

```
public long getId()
```

Returns the ID of the room object

**Returns:**

The ID of the room object

---

## getName

```
public java.lang.String getName()
```

Returns the name of the room object

**Returns:**

The name of the room object

---

## getDescription

```
public java.lang.String getDescription()
```

Returns the description of the room object

**Returns:**

The description of the room object

---

## setId

```
public void setId(long id)
```

Sets the ID for the room object

**Parameters:**

id - The ID to set

---

## setName

```
public void setName(java.lang.String name)
```

Sets the name for the room object

**Parameters:**

name - The name to set

---

## setDescription

```
public void setDescription(java.lang.String description)
```

Sets the description for the room object

**Parameters:**

description - The description to set

---

## envirosense.model

### Class Sensor

```
java.lang.Object
└--envirosense.model.Sensor
```

---

```
public class Sensor
extends java.lang.Object
```

---

#### Fields

##### id

```
private long id
```

---

##### room

```
private envirosense.model.Room room
```

---

##### name

```
private java.lang.String name
```

---

##### sensorType

```
private envirosense.model.SensorType sensorType
```

---

#### Constructors

##### Sensor

```
public Sensor(long id,
               Room room,
               java.lang.String name,
               SensorType sensorType)
```

---

##### Sensor

```
protected Sensor()
```

---

(continued from last page)

## Methods

### **getId**

```
public long getId()
```

### **getRoom**

```
public Room getRoom()
```

### **getName**

```
public java.lang.String getName()
```

### **getSensorType**

```
public SensorType getSensorType()
```

### **setId**

```
public void setId(long id)
```

### **setRoom**

```
public void setRoom(Room room)
```

### **setName**

```
public void setName(java.lang.String name)
```

### **setSensorType**

```
public void setSensorType(SensorType sensorType)
```

## envirosense.model

### Class SensorDataPK

java.lang.Object

└─envirosense.model.SensorDataPK

#### All Implemented Interfaces:

java.io.Serializable

```
public class SensorDataPK
  extends java.lang.Object
  implements java.io.Serializable
```

This class is the `IdClass` for all sensor data classes

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

### sensorId

```
private long sensorId
```

### timestamp

```
private java.sql.Timestamp timestamp
```

## Constructors

### SensorDataPK

```
public SensorDataPK(long sensorId,
                    java.sql.Timestamp timestamp)
```

Creates a new `SensorData` object using the specified arguments

#### Parameters:

`sensorId` - The ID of the sensor data to be created

`timestamp` - The time and date of the sensor data to be created

### SensorDataPK

```
public SensorDataPK()
```

(continued from last page)

## Methods

### **getSensorId**

```
public long getSensorId()
```

### **getTimestamp**

```
public java.sql.Timestamp getTimestamp()
```

### **setTimestamp**

```
public void setTimestamp(java.sql.Timestamp timestamp)
```

### **setSensorId**

```
public void setSensorId(long sensorId)
```

### **hashCode**

```
public int hashCode()
```

### **equals**

```
public boolean equals(java.lang.Object obj)
```



## envirosense.model

### Class SensorType

```
java.lang.Object
  |
  +- java.lang.Enum
        +- envirosense.model.SensorType
```

#### All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

```
public final class SensorType
extends java.lang.Enum
```

A SensorType describes what type of data a particular sensor reads.

## Fields

### TE

```
public static final envirosense.model.SensorType TE

    Temperature sensor
```

### HU

```
public static final envirosense.model.SensorType HU

    Humidity sensor
```

### DR

```
public static final envirosense.model.SensorType DR

    Door sensor (for detecting opening and closing of doors)
```

### MO

```
public static final envirosense.model.SensorType MO

    Motion sensor
```

### PA

```
public static final envirosense.model.SensorType PA

    Particle or dust sensor
```

### RA

```
public static final envirosense.model.SensorType RA

    Indoor tracking sensor
```

## UK

```
public static final envirosense.model.SensorType UK
```

The sensor type is unknown

---

## sensorType

```
java.lang.String sensorType
```

## Constructors

### SensorType

```
private SensorType(java.lang.String sensorType)
```

## Methods

### values

```
public static SensorType\[\] values()
```

---

### valueOf

```
public static SensorType valueOf(java.lang.String name)
```

---

### getSensorType

```
public java.lang.String getSensorType()
```

Returns a string representation of the constant

**Returns:**

A string representation of the constant

## envirosense.model

# Class Temperature

java.lang.Object

└--envirosense.model.Temperature

### All Implemented Interfaces:

java.io.Serializable

---

public class **Temperature**  
extends java.lang.Object  
implements java.io.Serializable

---

## Fields

### serialVersionUID

private static final long **serialVersionUID**

Constant value: 1

---

### sensorId

private long **sensorId**

---

### timestamp

private java.sql.Timestamp **timestamp**

---

### data

private double **data**

---

## Constructors

### Temperature

```
public Temperature(long sensorId,  
                  java.sql.Timestamp timestamp,  
                  double data)
```

---

(continued from last page)

## Temperature

```
public Temperature()
```

## Methods

### **getSensorId**

```
public long getSensorId()
```

---

### **getTimestamp**

```
public java.sql.Timestamp getTimestamp()
```

---

### **getData**

```
public java.lang.Double getData()
```

---

### **setSensorId**

```
public void setSensorId(long sensorId)
```

---

### **setTimestamp**

```
public void setTimestamp(java.sql.Timestamp timestamp)
```

---

### **setData**

```
public void setData(double data)
```

## envirosense.model

### Class User

```
java.lang.Object
└--envirosense.model.User
```

```
public class User
extends java.lang.Object
```

Standard User Object

#### Fields

##### firstname

```
private java.lang.String firstname
```

The first name of the user

##### lastname

```
private java.lang.String lastname
```

The last name of the user

##### email

```
private java.lang.String email
```

The email address of the user. It is used for identifying the user and also for sending email notifications.

##### phone

```
private java.lang.String phone
```

The phone number of the user. It is used for sending SMS notifications

##### slackId

```
private java.lang.String slackId
```

The Slack ID (or username) of a user Used for sending notifications to the user through Slack

##### password

```
private java.lang.String password
```

The password of the user. It used for authenticating the user into the system

##### salt

```
private java.lang.String salt
```

(continued from last page)

Used for salting the password for storage

---

## enabled

```
private boolean enabled
```

A flag that indicates whether a user is active in the system

---

## roles

```
private java.util.Set roles
```

---

## events

```
private java.util.Set events
```

---

# Constructors

## User

```
public User(java.lang.String firstname,  
            java.lang.String lastname,  
            java.lang.String email,  
            java.lang.String phone,  
            java.lang.String slackId,  
            java.lang.String password,  
            java.lang.String salt,  
            boolean enabled,  
            java.util.Set roles)
```

Creates a user object

### Parameters:

firstname - The first name of the user object to be created  
lastname - The last name of the user object to be created  
email - The email address of the user object to be created  
phone - The phone number of the user object to be created  
slackId - The slack ID (or username) of the user object to be created  
password - The password of the user object to be created  
enabled - A boolean flag indicating whether the user object to be created is enabled  
roles - The roles associated with the user

---

## User

```
public User(User user)
```

---

## User

```
protected User()
```

---

# Methods

(continued from last page)

## getEmail

```
public java.lang.String getEmail()
```

Returns the email address of the user object

**Returns:**

The email address of a user object.

---

## getFirstname

```
public java.lang.String getFirstname()
```

Returns the first name of the user object.

**Returns:**

The first name of the user object.

---

## getLastName

```
public java.lang.String getLastName()
```

Returns the last name of the user object.

**Returns:**

The last name of the user object.

---

## getPhone

```
public java.lang.String getPhone()
```

Returns the phone number of the user object.

**Returns:**

The phone number of the user object.

---

## getSlackId

```
public java.lang.String getSlackId()
```

Returns the slack ID (or username) of the user object.

**Returns:**

the slack ID (or username) of the user object.

---

## getPassword

```
public java.lang.String getPassword()
```

Returns the password of the user object.

**Returns:**

the password of the user object.

---

## getEnabled

```
public boolean getEnabled()
```

---

(continued from last page)

Returns the status flag of the user object.

**Returns:**

true if the user is active, false if otherwise.

---

## getEvents

```
public java.util.Set getEvents()
```

---

## getRoles

```
public java.util.Set getRoles()
```

---

## setEmail

```
public void setEmail(java.lang.String email)
```

Sets the email address of the user object.

**Parameters:**

email - The email address to set.

---

## setFirstname

```
public void setFirstname(java.lang.String firstname)
```

Sets the first name of the user object.

**Parameters:**

firstname - The first name to set

---

## setLastname

```
public void setLastname(java.lang.String lastname)
```

Sets the last name of the user object.

**Parameters:**

lastname - The last name to set

---

## setPhone

```
public void setPhone(java.lang.String phone)
```

Sets the phone number of the user object.

**Parameters:**

phone - The phone number to set

---

## setSlackId

```
public void setSlackId(java.lang.String slackId)
```

Sets the Slack ID (or username) of the user object.

---



(continued from last page)

**Parameters:**

slackId - The Slack ID (or username) to set.

---

**setPassword**

```
public void setPassword(java.lang.String password)
```

Sets the password of the user object.

**Parameters:**

password - The password to set

---

**setEnabled**

```
public void setEnabled(boolean state)
```

Sets the status flag of a user in the system.

**Parameters:**

state - A boolean indicating the status of a user in the system. true for active, false for inactive

---

**setSalt**

```
public void setSalt(java.lang.String salt)
```

---

**setEvents**

```
public void setEvents(java.util.Set events)
```

---

**setRoles**

```
public void setRoles(java.util.Set roles)
```

---

**hashCode**

```
public int hashCode()
```

---

**equals**

```
public boolean equals(java.lang.Object obj)
```

---

**getSalt**

```
public java.lang.String getSalt()
```

---

## envirosense.model

### Class UserRole

java.lang.Object

└─envirosense.model.UserRole

#### All Implemented Interfaces:

java.io.Serializable

```
public class UserRole
  extends java.lang.Object
  implements java.io.Serializable
```

A UserRole is used to determine which actions a user can perform on the system.

## Fields

### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

### role

```
private envirosense.model.Role role
```

### user

```
private envirosense.model.User user
```

## Constructors

### UserRole

```
public UserRole(User user,
                 Role role)
```

### UserRole

```
protected UserRole()
```

## Methods

(continued from last page)

## **getUserEmail**

```
public java.lang.String getUserEmail()
```

---

## **getRole**

```
public Role getRole()
```

---

## **setUser**

```
public void setUser(User user)
```

---

## **setRole**

```
public void setRole(Role role)
```

---

## **hashCode**

```
public int hashCode()
```

---

## **equals**

```
public boolean equals(java.lang.Object obj)
```

---

---

**Package**  
**envirosense.model.dto**

## envirosense.model.dto

### Class BluetoothDataDTO

java.lang.Object

└─envirosense.model.dto.BluetoothDataDTO

```
public class BluetoothDataDTO
    extends java.lang.Object
```

#### Fields

##### **rsi**

```
private int rsi
```

##### **beaconId**

```
private java.lang.String beaconId
```

##### **userEmail**

```
private java.lang.String userEmail
```

#### Constructors

##### **BluetoothDataDTO**

```
public BluetoothDataDTO(int rssi,
                        java.lang.String beaconId,
                        java.lang.String userEmail)
```

#### Methods

##### **getRssi**

```
public int getRssi()
```

##### **getBeaconId**

```
public java.lang.String getBeaconId()
```

---

## **getUserEmail**

```
public java.lang.String getUserEmail()
```

---

## **setRssi**

```
public void setRssi(int rssi)
```

---

## **setBeaconId**

```
public void setBeaconId(java.lang.String beaconId)
```

---

## **setUserEmail**

```
public void setUserEmail(java.lang.String userEmail)
```

---

## envirosense.model.dto

### Class SensorDataDTO

```
java.lang.Object
└--envirosense.model.dto.SensorDataDTO
```

```
public class SensorDataDTO
extends java.lang.Object
```

This class models a sensor data type

#### Fields

##### **sensorId**

```
private long sensorId
```

##### **roomName**

```
private java.lang.String roomName
```

##### **roomDescription**

```
private java.lang.String roomDescription
```

##### **data**

```
private java.lang.Object data
```

##### **sensorType**

```
private envirosense.model.SensorType sensorType
```

##### **timestamp**

```
private java.sql.Timestamp timestamp
```

#### Constructors

(continued from last page)

## SensorDataDTO

```
public SensorDataDTO(long id,
    java.lang.String roomName,
    java.lang.String roomDescription,
    java.lang.Object data,
    java.sql.Timestamp timestamp,
    SensorType type)
```

## SensorDataDTO

```
public SensorDataDTO()
```

## Methods

### getData

```
public java.lang.Object getData()
```

Returns the data contained in a SensorData object

**Returns:**

The data contained in a SensorData object

### getTimestamp

```
public java.sql.Timestamp getTimestamp()
```

Returns the date and time that data was read in a SensorData object

**Returns:**

the date and time that data was read in a SensorData object

### getSensorId

```
public long getSensorId()
```

Returns the ID of the sensor that read the data

**Returns:**

The ID of the sensor that read the data

### getRoomName

```
public java.lang.String getRoomName()
```

Returns the name of the room in which this data was read

**Returns:**

The name of the room in which this data was read

### getRoomDescription

```
public java.lang.String getRoomDescription()
```



---

(continued from last page)

Returns the description of the room in which this data was read

**Returns:**

The description of the room in which this data was read

---

## getSensorType

```
public SensorType getSensorType()
```

Returns the sensor type of the sensor that read the data

**Returns:**

The sensor type of the sensor that read the data

---

## setSensorId

```
public void setSensorId(long sensorId)
```

Sets the ID of the sensor that read the data

**Parameters:**

sensorId - The ID of the sensor that read the data

---

## setRoomName

```
public void setRoomName(java.lang.String roomName)
```

**Parameters:**

roomName

---

## setRoomDescription

```
public void setRoomDescription(java.lang.String roomDescription)
```

**Parameters:**

roomDescription

---

## setData

```
public void setData(java.lang.Object data)
```

Sets the data that was read

**Parameters:**

data - The data that was read

---

## setSensorType

```
public void setSensorType(SensorType sensorType)
```

Sets the type of sensor that read the data

**Parameters:**

sensorType - The type of sensor that read the data

---

---

## setTimestamp

```
public void setTimestamp(java.sql.Timestamp timestamp)
```

Sets the timestamp for when the data was read

### Parameters:

`timestamp` - The time and date when the data was read

## envirosense.model.dto

### Class UserDTO

```
java.lang.Object
└--envirosense.model.dto.UserDTO
```

---

```
public class UserDTO
extends java.lang.Object
```

---

#### Fields

##### **firstname**

```
private java.lang.String firstname
```

---

##### **lastname**

```
private java.lang.String lastname
```

---

##### **email**

```
private java.lang.String email
```

---

##### **phone**

```
private java.lang.String phone
```

---

##### **slackId**

```
private java.lang.String slackId
```

---

##### **password**

```
private java.lang.String password
```

---

##### **enabled**

```
private boolean enabled
```

---

---

## roles

```
private java.util.Set roles
```

---

## events

```
private java.util.Set events
```

---

## Constructors

### UserDTO

```
public UserDTO()
```

---

### UserDTO

```
public UserDTO(java.lang.String firstname,  
               java.lang.String lastname,  
               java.lang.String email,  
               java.lang.String phone,  
               java.lang.String slackId,  
               java.lang.String password,  
               boolean enabled,  
               java.util.Set events,  
               java.util.Set roles)
```

---

### UserDTO

```
public UserDTO(User user)
```

---

## Methods

### getFirstname

```
public java.lang.String getFirstname()
```

---

### getLastname

```
public java.lang.String getLastname()
```

---

### getEmail

```
public java.lang.String getEmail()
```

---

(continued from last page)

---

## getPhone

```
public java.lang.String getPhone()
```

---

## getSlackId

```
public java.lang.String getSlackId()
```

---

## getPassword

```
public java.lang.String getPassword()
```

---

## isEnabled

```
public boolean isEnabled()
```

---

## getRoles

```
public java.util.Set getRoles()
```

---

## getEvents

```
public java.util.Set getEvents()
```

---

## setFirstname

```
public void setFirstname(java.lang.String firstname)
```

---

## setLastname

```
public void setLastname(java.lang.String lastname)
```

---

## setEmail

```
public void setEmail(java.lang.String email)
```

---

(continued from last page)

## setPhone

```
public void setPhone(java.lang.String phone)
```

---

## setSlackId

```
public void setSlackId(java.lang.String slackId)
```

---

## setPassword

```
public void setPassword(java.lang.String password)
```

---

## setEnabled

```
public void setEnabled(boolean enabled)
```

---

## setRoles

```
public void setRoles(java.util.Set roles)
```

---

## setEvents

```
public void setEvents(java.util.Set events)
```

---

## mapNoPassword

```
public static UserDTO mapNoPassword(User user)
```

---

## mapNewUser

```
public static User mapNewUser(UserDTO userDTO)
```

---

---

# Package

# **envirosense.repository**

This package holds interfaces and classes DAO

## envirosense.repository

# Interface BluetoothBeaconRepository

---

public interface **BluetoothBeaconRepository**  
extends

Repository interface for getting bluetooth beacon information

---

## Methods

### **findByUser**

```
public abstract java.util.List findByUser(java.lang.String user)
```

Returns all beacons owned by the specified user

**Parameters:**

`user` - The user whose beacons are to be retrieved

**Returns:**

Beacons owned by the specified user



## **envirosense.repository**

# **Interface ConditionRepository**

---

public interface **ConditionRepository**  
extends

---

## envirosense.repository

# Interface DoorRepository

---

public interface **DoorRepository**  
extends

---

## Methods

### findByTimestampBetween

```
public abstract java.util.List findByTimestampBetween(java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all door data that was read between the specified timestamps

**Parameters:**

start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of door data that satisfy the conditions outlined above.

---

### findByRoomIdAndTimestampBetween

```
public abstract java.util.List findByRoomIdAndTimestampBetween(long roomId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all door data that was read from the specified room with the specified time range.

**Parameters:**

roomId - The room in which the data was read  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of door data that satisfy the conditions explained above.

---

### findBySensorIdAndTimestampBetween

```
public abstract java.util.List findBySensorIdAndTimestampBetween(long sensorId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all door data that was read by the specified sensor in the specified time range

**Parameters:**

sensorId - The ID of the sensor that read the data being retrieved  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of door data that satisfy the conditions explained above.

---

## findBySensorId

```
public abstract java.util.List findBySensorId(long sensorId)
```

Retrieves all door data that was read by the sensor with the ID specified

**Parameters:**

sensorId - The ID of the sensor that read the data to be returned

**Returns:**

A list of door data that satisfy the conditions outlined above

---

## findByRoomId

```
public abstract java.util.List findByRoomId(long roomId)
```

Retrieves all motion data that was read in the room with the ID specified

**Parameters:**

roomId - The ID of the room in which the data was read

**Returns:**

A list of motion data that satisfy the conditions outlined above

---

## findLatestByRoomId

```
public abstract java.util.List findLatestByRoomId(long roomId)
```

Retrieves the latest door data from the room the specified ID.

**Parameters:**

roomId - The ID of the room in which the data was read

**Returns:**

A List of door data that satisfy the conditions given above.

---

## findLatestBySensorId

```
public abstract java.util.List findLatestBySensorId(long sensorId)
```

Retrieves the latest door data stored in the database.

**Parameters:**

sensorId - The ID of the room in which the data was read

**Returns:**

A List of door data that satisfy the conditions given above.

---

## findLatest

```
public abstract java.util.List findLatest()
```

Retrieves the latest door data from all rooms stored in the database.

**Returns:**

A List of door data that satisfy the conditions given above.

---

## envirosense.repository Interface EventRepository

---

public interface **EventRepository**  
extends

---

### Methods

#### **findByName**

```
public abstract java.util.List findByName(java.lang.String name)
```

---

#### **findByActiveTrue**

```
public abstract java.util.List findByActiveTrue()
```

---

#### **findByActiveFalse**

```
public abstract java.util.List findByActiveFalse()
```

---

#### **findByUserEmail**

```
public abstract java.util.List findByUserEmail(java.lang.String email)
```

---

#### **findByUserEmailActive**

```
public abstract java.util.List findByUserEmailActive(java.lang.String email)
```

---

#### **findByUserEmailInactive**

```
public abstract java.util.List findByUserEmailInactive(java.lang.String email)
```

## envirosense.repository

# Interface HumidityRepository

---

public interface **HumidityRepository**  
extends

---

## Methods

### findByTimestampBetween

```
public abstract java.util.List findByTimestampBetween(java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves and returns all humidity data that was recorded between the specified time range

**Parameters:**

start - The earliest date and time for the data to be retrieved  
end - The latest date and time for the data to be retrieved

**Returns:**

A list containing data retrieved

---

### findByRoomIdAndTimestampBetween

```
public abstract java.util.List findByRoomIdAndTimestampBetween(long roomId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all humidity data that was read from the specified room with the specified time range.

**Parameters:**

roomId - The room in which the data was read  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of humidity data that satisfy the conditions explained above.

---

### findBySensorIdAndTimestampBetween

```
public abstract java.util.List findBySensorIdAndTimestampBetween(long sensorId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all humidity data that was read by the specified sensor in the specified time range

**Parameters:**

sensorId - The ID of the sensor that read the data being retrieved  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of humidity data that satisfy the conditions explained above.

---

---

## findBySensorId

```
public abstract java.util.List findBySensorId(long sensorId)
```

Retrieves all humidity data that was read by the sensor with the ID specified

**Parameters:**

sensorId - The ID of the sensor that read the data to be returned

**Returns:**

A list of humidity data that satisfy the conditions outlined above

---

## findByRoomId

```
public abstract java.util.List findByRoomId(long roomId)
```

Retrieves all temperature data that was read in the room with the ID specified

**Parameters:**

roomId - The ID of the room in which the data was read

**Returns:**

A list of temperature data that satisfy the conditions outlined above

---

## findLatestByRoomId

```
public abstract java.util.List findLatestByRoomId(long roomId)
```

Retrieves the latest humidity data from the room the specified ID.

**Parameters:**

roomId - The ID of the room in which the data was read

**Returns:**

A List of humidity data that satisfy the conditions given above.

---

## findLatestBySensorId

```
public abstract java.util.List findLatestBySensorId(long sensorId)
```

Retrieves the latest humidity data stored in the database.

**Parameters:**

sensorId - The ID of the room in which the data was read

**Returns:**

A List of humidity data that satisfy the conditions given above.

---

## findLatest

```
public abstract java.util.List findLatest()
```

Retrieves the latest humidity data stored in the database.

**Returns:**

A List of humidity data that satisfy the conditions given above.

---

## envirosense.repository

# Interface MotionRepository

---

public interface **MotionRepository**  
extends

---

## Methods

### findByTimestampBetween

```
public abstract java.util.List findByTimestampBetween(java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all motion data that was read between the specified timestamps

**Parameters:**

start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of motion data that satisfy the conditions outlined above.

---

### findByRoomIdAndTimestampBetween

```
public abstract java.util.List findByRoomIdAndTimestampBetween(long roomId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all motion data that was read from the specified room with the specified time range.

**Parameters:**

roomId - The room in which the data was read  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of motion data that satisfy the conditions explained above.

---

### findBySensorIdAndTimestampBetween

```
public abstract java.util.List findBySensorIdAndTimestampBetween(long sensorId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all motion data that was read by the specified sensor in the specified time range

**Parameters:**

sensorId - The ID of the sensor that read the data being retrieved  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of motion data that satisfy the conditions explained above.

---

## findBySensorId

```
public abstract java.util.List findBySensorId(long sensorId)
```

Retrieves all motion data that was read by the sensor with the ID specified

**Parameters:**

`sensorId` - The ID of the sensor that read the data to be returned

**Returns:**

A list of motion data that satisfy the conditions outlined above

---

## findByRoomId

```
public abstract java.util.List findByRoomId(long roomId)
```

Retrieves all motion data that was read in the room with the ID specified

**Parameters:**

`roomId` - The ID of the room in which the data was read

**Returns:**

A list of motion data that satisfy the conditions outlined above

---

## findLatestByRoomId

```
public abstract java.util.List findLatestByRoomId(long roomId)
```

Retrieves the latest motion data from the room the specified ID.

**Parameters:**

`roomId` - The ID of the room in which the data was read

**Returns:**

A List of motion data that satisfy the conditions given above.

---

## findLatestBySensorId

```
public abstract java.util.List findLatestBySensorId(long sensorId)
```

Retrieves the latest motion data stored in the database.

**Parameters:**

`sensorId` - The ID of the room in which the data was read

**Returns:**

A List of motion data that satisfy the conditions given above.

---

## findLatest

```
public abstract java.util.List findLatest()
```

Retrieves the latest motion data stored in the database.

**Returns:**

A List of motion data that satisfy the conditions given above.

---



## envirosense.repository

# Interface RaBluetoothRepository

---

public interface **RaBluetoothRepository**  
extends

---

## Methods

### findByTimestampBetween

```
public abstract java.util.List findByTimestampBetween(java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves and returns all presence data that was recorded between the specified time range

**Parameters:**

start - The earliest date and time for the data to be retrieved  
end - The latest date and time for the data to be retrieved

**Returns:**

A list containing data retrieved

---

### findByRoomIdAndTimestampBetween

```
public abstract java.util.List findByRoomIdAndTimestampBetween(long roomId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all presence data that was read from the specified room with the specified time range.

**Parameters:**

roomId - The room in which the data was read  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of presence data that satisfy the conditions explained above.

---

### findBySensorIdAndTimestampBetween

```
public abstract java.util.List findBySensorIdAndTimestampBetween(long sensorId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all presence data that was read by the specified sensor in the specified time range

**Parameters:**

sensorId - The ID of the sensor that read the data being retrieved  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of presence data that satisfy the conditions explained above.

---

---

## findBySensorId

```
public abstract java.util.List findBySensorId(long sensorId)
```

Retrieves all presence data that was read by the sensor with the ID specified

**Parameters:**

sensorId - The ID of the sensor that read the data to be returned

**Returns:**

A list of presence data that satisfy the conditions outlined above

---

## findByRoomId

```
public abstract java.util.List findByRoomId(long roomId)
```

Retrieves all presence data that was read in the room with the ID specified

**Parameters:**

sensorId - The ID of the room in which the data was read

**Returns:**

A list of presence data that satisfy the conditions outlined above

---

## findLatestByRoomId

```
public abstract java.util.List findLatestByRoomId(long roomId)
```

Retrieves the latest presence data from the room the specified ID.

**Parameters:**

roomId - The ID of the room in which the data was read

**Returns:**

A List of presence data that satisfy the conditions given above.

---

## findLatestBySensorId

```
public abstract java.util.List findLatestBySensorId(long sensorId)
```

Retrieves the latest presence data stored in the database.

**Returns:**

A List of presence data that satisfy the conditions given above.

---

## findLatest

```
public abstract java.util.List findLatest()
```

Retrieves the latest presence data stored in the database.

**Returns:**

A List of presence data that satisfy the conditions given above.

---

## **envirosense.repository** **Interface RoleRepository**

---

public interface **RoleRepository**  
extends

---

## envirosense.repository Interface RoomRepository

---

public interface **RoomRepository**  
extends

---

## envirosense.repository Interface SensorRepsitory

---

public interface **SensorRepsitory**  
extends

---

### Methods

#### **findBySensorType**

public abstract java.util.List **findBySensorType**([SensorType](#) type)

Retrieves and returns all sensors of the specified specified type

**Parameters:**

type - The type of the sensors to retrieve

**Returns:**

The retrieved sensors

## envirosense.repository

# Interface TemperatureRepository

---

public interface **TemperatureRepository**  
extends

---

## Methods

### findByTimestampBetween

```
public abstract java.util.List findByTimestampBetween(java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all temperature data that was read between the specified timestamps

**Parameters:**

start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of temperature data that satisfy the conditions outlined above.

---

### findByRoomIdAndTimestampBetween

```
public abstract java.util.List findByRoomIdAndTimestampBetween(long roomId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all temperature data that was read from the specified room with the specified time range.

**Parameters:**

roomId - The room in which the data was read  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of temperature that satisfies the conditions explained above.

---

### findBySensorIdAndTimestampBetween

```
public abstract java.util.List findBySensorIdAndTimestampBetween(long sensorId,  
java.sql.Timestamp start,  
java.sql.Timestamp end)
```

Retrieves all temperature data that was read by the specified sensor in the specified time range

**Parameters:**

sensorId - The ID of the sensor that read the data being retrieved  
start - The time and date to start checking (inclusive)  
end - The time and date to end checking (inclusive)

**Returns:**

A list of temperature data that satisfy the conditions explained above.

---

## findBySensorId

```
public abstract java.util.List findBySensorId(long sensorId)
```

Retrieves all temperature data that was read by the sensor with the ID specified

**Parameters:**

`sensorId` - The ID of the sensor that read the data to be returned

**Returns:**

A list of temperature data that satisfy the conditions outlined above

---

## findByRoomId

```
public abstract java.util.List findByRoomId(long roomId)
```

Retrieves all temperature data that was read in the room with the ID specified

**Parameters:**

`roomId` - The ID of the room in which the data was read

**Returns:**

A list of temperature data that satisfy the conditions outlined above

---

## findLatestByRoomId

```
public abstract java.util.List findLatestByRoomId(long roomId)
```

Retrieves the latest temperature data from the room the specified ID.

**Parameters:**

`roomId` - The ID of the room in which the data was read

**Returns:**

A List of temperature data that satisfy the conditions given above.

---

## findLatestBySensorId

```
public abstract java.util.List findLatestBySensorId(long sensorId)
```

Retrieves the latest temperature data stored in the database.

**Parameters:**

`sensorId` - The ID of the room in which the data was read

**Returns:**

A List of temperature data that satisfy the conditions given above.

---

## findLatest

```
public abstract java.util.List findLatest()
```

Retrieves the latest temperature data stored in the database.

**Returns:**

A List of temperature data that satisfy the conditions given above.

---

## envirosense.repository

# Interface UserRepository

---

public interface **UserRepository**  
extends

Repository for retrieving users from the database

---

## Methods

### **findByEnabledTrue**

```
public abstract java.util.Set findByEnabledTrue()
```

Returns all active users

**Returns:**

All active users

---

### **findByEnabledFalse**

```
public abstract java.util.Set findByEnabledFalse()
```

Returns all inactive users

**Returns:**

All inactive users

---

### **findByFirstname**

```
public abstract java.util.Set findByFirstname(java.lang.String firstname)
```

Returns all users whose first name match the first name provided

**Parameters:**

firstname - The first name to check form

**Returns:**

All users whose first name match the first name provided

---

### **findByLastname**

```
public abstract java.util.Set findByLastname(java.lang.String lastname)
```

Returns all users whose last name match the last name provided

**Parameters:**

firstname - The last name to check form

**Returns:**

All users whose last name match the last name provided

---



(continued from last page)

## findByEmailIgnoreCase

```
public abstract User findByEmailIgnoreCase(java.lang.String email)
```

Returns the user with the specified email.

**Parameters:**

email - The email of the user to return

**Returns:**

A user object whose email matches (case insensitive) the one specified

---

## findByLastnameIgnoreCase

```
public abstract java.util.Set findByLastnameIgnoreCase(java.lang.String lastname)
```

Returns all users whose last name match the last name provided

**Parameters:**

lastname - The last name to check form

**Returns:**

All users whose last name match (case insensitive) the last name provided

---

## findByFirstnameIgnoreCase

```
public abstract java.util.Set findByFirstnameIgnoreCase(java.lang.String firstname)
```

Returns all users whose first name match the first name provided

**Parameters:**

firstname - The first name to check form

**Returns:**

All users whose first name match (case insensitive) the first name provided

---

---

## Package

# envirosense.service

Defines service classes, which are implementations for the applicaiton logic

## envirosense.service Interface CommunicationMethod

All Known Implementing Classes:

[SlackNotifier](#), [EmailNotifier](#)

---

public interface **CommunicationMethod**  
extends

A service for sending messages to users. Information for the message destination is can be found in a user object.

---

### Methods

#### sendMessage

```
public abstract boolean sendMessage(User receivingUser,  
    java.lang.String message)
```

Sends the a message to the specified user with message as the content

**Parameters:**

`receivingUser` - The user that the message will be sent to  
`message` - The content of the message

**Returns:**

True if the send was successful, false if otherwise

## envirosense.service Class CustomUserDetails

```
java.lang.Object
├--envirosense.service.CustomUserDetails
```

```
public class CustomUserDetails
    extends java.lang.Object
```

### Fields

#### serialVersionUID

```
private static final long serialVersionUID
```

Constant value: **1**

#### username

```
private final java.lang.String username
```

#### password

```
private final java.lang.String password
```

#### salt

```
private final java.lang.String salt
```

#### enabled

```
private final boolean enabled
```

#### authorities

```
private final java.util.Collection authorities
```

### Constructors

(continued from last page)

## CustomUserDetails

```
public CustomUserDetails(java.lang.String username,  
                          java.lang.String password,  
                          java.lang.String salt,  
                          boolean enabled,  
                          java.util.Collection authorities)
```

## Methods

### getAuthorities

```
public java.util.Collection getAuthorities()
```

### getPassword

```
public java.lang.String getPassword()
```

### getUsername

```
public java.lang.String getUsername()
```

### getSalt

```
public java.lang.String getSalt()
```

### isAccountNonExpired

```
public boolean isAccountNonExpired()
```

### isAccountNonLocked

```
public boolean isAccountNonLocked()
```

### isCredentialsNonExpired

```
public boolean isCredentialsNonExpired()
```

### isEnabled

```
public boolean isEnabled()
```

(continued from last page)

## envirosense.service Class EmailNotifier

java.lang.Object

└─envirosense.service.EmailNotifier

All Implemented Interfaces:

[CommunicationMethod](#)

```
public class EmailNotifier
  extends java.lang.Object
  implements CommunicationMethod
```

This service class is used for sending messages to users via email

### Fields

#### logger

```
private static final Logger logger
```

#### mailSender

```
JavaMailSender mailSender
```

#### from

```
java.lang.String from
```

### Constructors

#### EmailNotifier

```
public EmailNotifier()
```

### Methods

#### sendMessage

```
public boolean sendMessage(User receivingUser,
  java.lang.String message)
```

## envirosense.service Class EventHandler

```
java.lang.Object
└--envirosense.service.EventHandler
```

```
public class EventHandler
extends java.lang.Object
```

Filter incoming and send notifications for the ones that satisfy an events conditions.

### Fields

#### eventRepository

```
envirosense.repository.EventRepository eventRepository
```

#### emailNotifier

```
envirosense.service.EmailNotifier emailNotifier
```

#### slackNotifier

```
envirosense.service.SlackNotifier slackNotifier
```

### Constructors

#### EventHandler

```
public EventHandler()
```

Creates an EventHandler object

### Methods

#### run

```
public void run(java.util.List data)
```

Runs the handler on the specified data

##### Parameters:

data - The data on which the handler will run



(continued from last page)

## sendNotification

```
private boolean sendNotification(User user,  
    Event event)
```

Sends the message contained in the event to the specified user. The channel used for sending the message depends on which ones are set to be used.

**Parameters:**

`user` - The user to send the message to.  
`event` - The event whose message is to be sent

**Returns:**

`true` if the message was sent successfully on all set channels, `false` if otherwise.

## envirosense.service

### Class EventService

```
java.lang.Object
  |
  +--envirosense.service.EventService
```

---

```
public class EventService
  extends java.lang.Object
```

---

#### Fields

##### eventRepository

```
envirosense.repository.EventRepository eventRepository
```

#### Constructors

##### EventService

```
public EventService()
```

#### Methods

##### enableEvent

```
public boolean enableEvent(long id)
```

Enables the event with specified ID

**Parameters:**

`id` - The ID of the event to be enabled

**Returns:**

`true` if the operation executed successfully, return `false` otherwise

---

##### disableEvent

```
public boolean disableEvent(long id)
```

Disables the event with specified ID

**Parameters:**

`id` - The ID of the event to be disabled

**Returns:**

`true` if the operation executed successfully, return `false` otherwise

---

(continued from last page)

---

## addEvent

```
public Event addEvent(Event event)
```

Adds a new event

**Parameters:**

event - The event to be added

**Returns:**

The added event

---

## removeEvent

```
public boolean removeEvent(long id)
```

Retrieves and returns the event with specified ID

**Parameters:**

id - The ID of the event to retrieve

**Returns:**

The retrieved event

---

## findAll

```
public java.util.List findAll()
```

Retrieves and returns all events

**Returns:**

All events

---

## findByName

```
public java.util.List findByName(java.lang.String name)
```

Retrieves all events with the specified name

**Parameters:**

name - The name of the events to retrieve

**Returns:**

The retrieved events

---

## findByActiveTrue

```
public java.util.List findByActiveTrue()
```

Retrieves and returns all active events

**Returns:**

All active events

---

## findByActiveFalse

```
public java.util.List findByActiveFalse()
```

(continued from last page)

Retrieves and returns all inactive events

**Returns:**

All inactive events

---

## **findByUserEmail**

```
public java.util.List findByUserEmail(java.lang.String email)
```

Retrieves all events that this user has

**Parameters:**

email - The email of the user whose events are to be retrieved

**Returns:**

The retrieved events

---

## **findByUserEmailActive**

```
public java.util.List findByUserEmailActive(java.lang.String email)
```

Retrieves all active events that this user has

**Parameters:**

email - The email of the user whose events are to be retrieved

**Returns:**

The retrieved events

---

## **findByUserEmailInactive**

```
public java.util.List findByUserEmailInactive(java.lang.String email)
```

Retrieves all inactive events that this user has

**Parameters:**

email - The email of the user whose events are to be retrieved

**Returns:**

The retrieved events

---

## envirosense.service

### Class RoomService

java.lang.Object

└--envirosense.service.RoomService

```
public class RoomService
    extends java.lang.Object
```

## Fields

### roomRepository

envirosense.repository.RoomRepository **roomRepository**

## Constructors

### RoomService

```
public RoomService()
```

## Methods

### save

```
public Room save(Room room)
```

Saves the specified room back into the database

#### Parameters:

room - The room to save

#### Returns:

The saved room

### delete

```
public void delete(long id)
```

Deletes the room with specified ID

#### Parameters:

id - The ID of the room to delete

### findAll

```
public java.util.List findAll()
```

(continued from last page)

Retrieves and returns all rooms stored in the system

**Returns:**

All rooms stored in the system

---

## findOne

```
public Room findOne(long id)
```

Retrieves and returns the room with specified ID

**Parameters:**

id - The ID of the room to retrieve

**Returns:**

The retrieved room

## envirosense.service

### Class SensorDataService

java.lang.Object

└--envirosense.service.SensorDataService

---

public class **SensorDataService**  
extends java.lang.Object

---

## Fields

### temperatureRepository

private envirosense.repository.TemperatureRepository **temperatureRepository**

The temperature data repository: Used for retrieving temperature data from the DB

---

### humidityRepository

private envirosense.repository.HumidityRepository **humidityRepository**

The humidity data repository: Used for retrieving humidity data from the DB

---

### doorRepository

envirosense.repository.DoorRepository **doorRepository**

The door data repository: Used for retrieving door state data from the DB

---

### motionRepository

envirosense.repository.MotionRepository **motionRepository**

The motion data repository: Used for retrieving motion data from the DB

---

### sensorService

envirosense.service.SensorService **sensorService**

The sensor service: Used for information about sensors from the DB

---

### roomRepository

envirosense.repository.RoomRepository **roomRepository**

---

### sensorRepository

envirosense.repository.SensorRepository **sensorRepository**

---

## raRepository

envirosense.repository.RaBluetoothRepository **raRepository**

## Constructors

### SensorDataService

```
public SensorDataService()
```

## Methods

### findByRoomId

```
public java.util.List findByRoomId(long roomId)
```

Retrieves all data read in the room with the specified ID

**Parameters:**

roomId - The ID of the room in which the data to be retrieved was read

**Returns:**

Data read in the room with the specified ID

---

### findBySensorId

```
public java.util.List findBySensorId(long sensorId)
```

Retrieves all data read by the sensor with specified ID

**Parameters:**

sensorId - The ID of the sensor that read the data to be retrieved

**Returns:**

A list of sensor data read by the sensor with the specified ID

---

### findBySensorType

```
public java.util.List findBySensorType(SensorType sensorType)
```

Retrieves all data read by the sensors with specified type

**Parameters:**

sensorType - The type of the sensor that read the data to be retrieved

**Returns:**

A list of sensor data read by the sensor with the specified type

---

### findLastestByRoomId

```
public java.util.List findLastestByRoomId(long roomId)
```

Retrieves the latest data read by all sensors in the room with the specified ID

---



---

(continued from last page)

**Parameters:**

roomId - The ID of the room in which the data to be retrieved was read

**Returns:**

The latest data read by all sensors in the room with the specified ID

---

## findLatestBySensorId

```
public java.util.List findLatestBySensorId(long sensorId)
```

Retrieves the latest data read from the sensor with the ID specified

**Parameters:**

sensorId - The ID sensor whose latest data is to be retrieved

**Returns:**

The latest data read from the sensor with the ID specified

---

## findLatestBySensorType

```
public java.util.List findLatestBySensorType(SensorType sensorType)
```

Retrieves the latest data read by sensors with specified type in all rooms

**Parameters:**

sensorType - The type of the sensor whose latest data is to be retrieved

**Returns:**

The latest data read by sensors with specified type in all rooms

---

## findByRoomIdAndTimestampBetween

```
public java.util.List findByRoomIdAndTimestampBetween(long roomId,  
    java.sql.Timestamp startTime,  
    java.sql.Timestamp endTime)
```

Retrieves all sensor data that was read from the specified room within the specified time range.

**Parameters:**

roomId - The room in which the data was read  
startTime - The time and date to start checking (inclusive)  
endTime - The time and date to end checking (inclusive)

**Returns:**

A list of sensor data that satisfy the conditions explained above.

---

## findBySensorIdAndTimestampBetween

```
public java.util.List findBySensorIdAndTimestampBetween(long sensorId,  
    java.sql.Timestamp startTime,  
    java.sql.Timestamp endtime)
```

Retrieves data read by the sensor with specified ID within the specified time range

**Parameters:**

sensorId - The ID of the sensor that read the data to be retrieved  
startTime - The time and date to start checking (inclusive)  
endtime - The time and date to end checking (inclusive)

---

(continued from last page)

**Returns:**

A list of sensor data that meet the criteria described above

---

## findBySensorTypeAndTimestampBetween

```
public java.util.List findBySensorTypeAndTimestampBetween(SensorType sensorType,  
    java.sql.Timestamp startTime,  
    java.sql.Timestamp endTime)
```

Retrieves all sensor data from all sensors with type specified that was read between the specified time range

**Parameters:**

sensorType - The type of the sensor  
startTime - The time and date to start checking (inclusive)  
endTime - The time and date to end checking (inclusive)

**Returns:**

A list of sensor data that satisfy the conditions outlined above.

---

## findByRoomIdSensorTypeAndTimestamp

```
public java.util.List findByRoomIdSensorTypeAndTimestamp(long roomId,  
    SensorType sensorType,  
    java.sql.Timestamp startTime,  
    java.sql.Timestamp endTime)
```

Retrieves data read by sensors of the type specified in the room with the ID specified

**Parameters:**

roomId - The ID of the room in which the data to be retrieved was read  
sensorType - The type of sensors that read the data to be retrieved  
startTime - The time to start checking (inclusive)  
endTime - The time to end checking (inclusive)

**Returns:**

A list of sensor data matching the criteria explained above

---

## save

```
public void save(java.util.List data)
```

Saves sensor data contained in the data parameter

**Parameters:**

data - The data to be saved.

---

## mapTemperatureData

```
private java.util.List mapTemperatureData(java.util.List data)
```

Maps a list of temperature data to a list of SensorDataDTO

**Parameters:**

data - The temperature data to map

**Returns:**

A list of SensorDataDTO

## mapHumidityData

```
private java.util.List mapHumidityData(java.util.List data)
```

Maps a list of humidity data to a list of SensorDataDTO

**Parameters:**

data - The humidity data to map

**Returns:**

A list of SensorDataDTO

---

## mapDoorData

```
private java.util.List mapDoorData(java.util.List data)
```

Maps a list of door data to a list of SensorDataDTO

**Parameters:**

data - The door data to map

**Returns:**

A list of SensorDataDTO

---

## mapMotionData

```
private java.util.List mapMotionData(java.util.List data)
```

Maps a list of motion data to a list of SensorDataDTO

**Parameters:**

data - The motion data to map

**Returns:**

A list of SensorDataDTO

---

## mapRABleData

```
private java.util.List mapRABleData(java.util.List data)
```

Maps a list of ReelyActive data to a list of SensorDataDTO

**Parameters:**

data - The ReelyActive data to map

**Returns:**

A list of SensorDataDTO

---

## getRoomInfo

```
private java.lang.String[] getRoomInfo(long sensorId)
```

Returns the name and description of the room in which the sensor with specified ID is located

**Parameters:**

sensorId - The sensor whose room information is to be retrieved.

---

(continued from last page)

**Returns:**

A two element array for the name and description of the room if found, or empty if not.

## envirosense.service Class SensorService

java.lang.Object

└─envirosense.service.SensorService

```
public class SensorService
    extends java.lang.Object
```

### Fields

#### sensorRepository

envirosense.repository.SensorRepository **sensorRepository**

### Constructors

#### SensorService

```
public SensorService()
```

### Methods

#### findAll

```
public java.util.List findAll()
```

Returns all sensors in the database

**Returns:**

All sensors in the database

#### findByType

```
public java.util.List findByType(java.lang.String type)
```

Returns all sensors of the type specified

**Parameters:**

type - The sensor type to search for

**Returns:**

All sensors of the type specified

#### findOne

```
public Sensor findOne(long id)
```

(continued from last page)

Returns the sensor with the specified ID

**Parameters:**

`id` - The ID of the sensor to return

**Returns:**

The sensor with the specified ID

## envirosense.service Class SlackNotifier

java.lang.Object

└─envirosense.service.SlackNotifier

All Implemented Interfaces:

[CommunicationMethod](#)

---

```
public class SlackNotifier
  extends java.lang.Object
  implements CommunicationMethod
```

This service class is used for sending messages to users via Slack

### Fields

#### logger

```
private static final Logger logger
```

#### slackApi

```
private SlackApi slackApi
```

#### commonRoom

```
private java.lang.String commonRoom
```

#### botName

```
private java.lang.String botName
```

### Constructors

#### SlackNotifier

```
public SlackNotifier()
```

### Methods

(continued from last page)

## sendMessage

```
public boolean sendMessage(User receivingUser,  
    java.lang.String message)
```



## envirosense.service Class UserDetailsAuthService

```
java.lang.Object
└--envirosense.service.UserDetailsAuthService
```

```
public class UserDetailsAuthService
extends java.lang.Object
```

### Fields

#### userRepository

```
envirosense.repository.UserRepository userRepository
```

#### roleRepository

```
envirosense.repository.RoleRepository roleRepository
```

### Constructors

#### UserDetailsAuthService

```
public UserDetailsAuthService()
```

### Methods

#### loadUserByUsername

```
public UserDetails loadUserByUsername(java.lang.String email)
throws UsernameNotFoundException
```

#### getGrantedAuthorities

```
public java.util.List getGrantedAuthorities(User user)
```

Returns all roles for the specified user as a list of GrantedAuthorities (roles)

##### Parameters:

user - The user whose authorities are to be returned

##### Returns:

A list of authorities for the specified user

## envirosense.service Interface UserService

All Known Implementing Classes:  
[UserServiceImpl](#)

---

public interface **UserService**  
extends

Interface for the Services of a User

---

### Methods

#### save

```
public abstract User save(User user)
```

Saves the provided user to the database

**Parameters:**

`user` - The user that is to be saved

**Returns:**

The saved user

---

#### update

```
public abstract User update(UserDTO userDTO)
```

Updates an existing user using information in userDTO

**Parameters:**

`userDTO` - The DTO containing the updated information

**Returns:**

The update user object

---

#### save

```
public abstract java.util.List save(java.util.List users)
```

Save the list of users provided to the database

**Parameters:**

`users` - The users to save

**Returns:**

The saved users

---

#### delete

```
public abstract void delete(User user)
```

Deletes the specified user from the database

---

(continued from last page)

**Parameters:**

user - The user to delete

---

**resetPassword**

```
public abstract User resetPassword(User user,  
    java.lang.String newPassword)
```

Reset the password of the specified user to the specified new password

**Parameters:**

user - The user whose password is to be reset

newPassword - The new password to set

**Returns:**

---

**findAllActive**

```
public abstract java.util.Set findAllActive()
```

Returns all active users

**Returns:**

All active users

---

**findAllInactive**

```
public abstract java.util.Set findAllInactive()
```

Returns all inactive users

**Returns:**

All inactive users

---

**findByFirstname**

```
public abstract java.util.Set findByFirstname(java.lang.String firstname)
```

Returns all users whose first name match the first name provided

**Parameters:**

firstname - The first name to check form

**Returns:**

All users whose first name match the first name provided

---

**findByLastname**

```
public abstract java.util.Set findByLastname(java.lang.String lastname)
```

Returns all users whose last name match the last name provided

**Parameters:**

lastname - The last name to check form

**Returns:**

All users whose last name match the last name provided

---

---

## findByEmail

```
public abstract User findByEmail(java.lang.String email)
```

Returns the user whose email address matches the email specified

### Parameters:

email - The email to use for search

### Returns:

The user with email address specified

---

## findAll

```
public abstract java.util.List findAll()
```

Returns all users stored in the database

### Returns:

All the users in the database

## envirosense.service

# Class UserServiceImpl

java.lang.Object

└─envirosense.service.UserServiceImpl

All Implemented Interfaces:

[UserService](#)

---

public class **UserServiceImpl**  
extends java.lang.Object  
implements [UserService](#)

Implementation of the Services for users

---

## Fields

### userRepository

envirosense.repository.UserRepository **userRepository**

---

### passwordEncoder

Md5PasswordEncoder **passwordEncoder**

---

## Constructors

### UserServiceImpl

public **UserServiceImpl**()

---

## Methods

### save

public [User](#) **save**([User](#) user)

---

### save

public java.util.List **save**(java.util.List users)

---

(continued from last page)

---

## update

```
public User update(UserDTO userDTO)
```

---

## resetPassword

```
public User resetPassword(User user,  
                           java.lang.String newPassword)
```

---

## delete

```
public void delete(User user)
```

---

## findAllActive

```
public java.util.Set findAllActive()
```

---

## finalAllInactive

```
public java.util.Set finalAllInactive()
```

---

## findByFirstname

```
public java.util.Set findByFirstname(java.lang.String firstname)
```

---

## findByLastname

```
public java.util.Set findByLastname(java.lang.String lastname)
```

---

## findByEmail

```
public User findByEmail(java.lang.String email)
```

---

## findAll

```
public java.util.List findAll()
```

---

## generateRandom

```
private static byte[] generateRandom(int n)
```

---

(continued from last page)

Returns n randomly generated bytes

**Parameters:**

n - The number of random bytes to return

**Returns:**

n randomly generated bytes

---

## byteToHex

```
private java.lang.String byteToHex(byte[] input)
```

Converts the input byte array to a hex sequence

**Parameters:**

input - The byte array to convert to hex sequence

**Returns:**

A hex sequence of the input byte array

---

## toHex

```
private char toHex(byte input)  
throws java.lang.IllegalArgumentException
```

Converts the lower 4 bits of input to a hex character

**Parameters:**

input - the byte whose lower 4 bits is to be converted

**Returns:**

A hex character mapping to the lower 4 bits of the input byte

**Throws:**

`IllegalArgumentException` - If the input byte is greater than 15 or less than 0

# Index

## A

active 50  
addEvent 122  
addResourceHandlers 4  
addViewControllers 4  
adminAllEvents 14  
all 22  
allHours 43  
allSensors 29  
ApplicationConfiguration 4  
authenticationProvider 9  
authorities 116

## B

beacon 57  
beaconId 85  
beaconRepository 22  
BluetoothBeacon 34  
BluetoothBeaconController 22  
BluetoothDataDTO 85  
botName 135  
byteToHex 143

## C

commonRoom 135  
Condition 36, 37  
ConditionModifier 41  
conditions 50  
ConditionTime 43, 44  
conditionTime 36  
configure 9, 10  
configureDefaultServletHandling 4  
configureGlobalSecurity 9  
CustomUserDetails 116

## D

data 47, 53, 55, 75, 87  
dataService 31  
dataSet 60

dataSource 4  
dateTime 43  
dayInCondition 38  
delete 125, 138, 142  
deleteEvent 25  
deleteUser 19  
description 67  
disableEvent 24  
disableEvent 122  
DISPATCHER\_SERVLET\_NAME 8  
Door 47  
doorRepository 127  
DR 73

## E

email 77, 91  
EmailConfiguration 7  
EmailNotifier 119  
emailNotifier 120  
enabled 78, 91, 116  
enableEvent 24, 122  
endtime 60  
entityManagerFactory 4  
env 4, 7, 12  
EQ 40  
equals 35, 66, 72, 81, 83  
evaluate 38  
Event 50  
EventApiController 24  
EventController 14  
EventHandler 120  
eventHandler 31  
eventRepository 120, 122  
events 78, 92  
EventService 122  
eventService 14, 24

## F

filterType 60  
filterValue 60  
finalAllInactive 139, 142  
findAll 123, 125, 133, 140, 142  
findAllActive 139, 142



findByActiveFalse 100, 123  
findByActiveTrue 100, 123  
findByEmail 140, 142  
findByEmailIgnoreCase 112  
findByEnabledFalse 112  
findByEnabledTrue 112  
findByFirstname 112, 139, 142  
findByFirstnameIgnoreCase 113  
findById 22  
findByLastname 112, 139, 142  
findByLastnameIgnoreCase 113  
findByName 100, 123  
findByRoomId 99, 102, 104, 106, 111, 128  
findByRoomIdAndTimestampBetween 98, 101, 103, 105, 110, 129  
findByRoomIdSensorTypeAndTimestamp 130  
findBySensorId 99, 102, 104, 106, 111, 128  
findBySensorIdAndTimestampBetween 98, 101, 103, 105, 110, 129  
findBySensorType 109, 128  
findBySensorTypeAndTimestampBetween 130  
findByTimestampBetween 98, 101, 103, 105, 110  
findByType 133  
findByUser 22, 96  
findByUserEmail 100, 124  
findByUserEmailActive 100, 124  
findByUserEmailInactive 100, 124  
findLastestByRoomId 128  
findLatest 99, 102, 104, 106, 111  
findLatestByRoomId 99, 102, 104, 106, 111  
findLatestBySensorId 99, 102, 104, 106, 111, 129  
findLatestBySensorType 129  
findOne 126, 133  
firstname 77, 91  
friday 43  
from 119

**G**

GE 40  
generateRandom 142  
getActive 25  
getActiveUsers 20  
getAll 25  
getAllRooms 28  
getAllUsers 20  
getAuthorities 117  
getBeacon 58  
getBeaconId 85  
getConditionModifier 41  
getConditions 51  
getConditionTime 37  
getData 48, 54, 56, 76, 88  
getDataByRoomId 26  
getDataBySensorId 27  
getDataBySensorType 26  
getDataBySensorTypeAndRoomId 27  
getDataSet 61  
getDateTime 45  
getDescription 68  
getEmail 78, 92  
getEnabled 79  
getEndTime 61  
getEvents 80, 93  
getFilterType 61  
getFilterValue 62  
getFirstname 79, 92  
getGrantedAuthorities 137  
getId 34, 37, 44, 50, 67, 69  
getInactive 25  
getInactiveUsers 20  
getLastname 79, 92  
getLatestByRoom 31  
getLatestBySensorType 32  
getMessage 51  
getModifier 37  
getName 50, 61, 68, 70  
getOne 28  
getOwners 51  
getPassword 79, 93, 117  
getPhone 79, 93  
getRole 65, 83  
getRoles 80, 93  
getRoom 70  
getRoomDescription 88  
getRoomInfo 131  
getRoomName 88  
getRssi 58, 85  
getSalt 81, 117  
getSensor 37

getSensorId 48, 54, 56, 58, 72, 76, 88  
getSensorType 70, 74, 89  
getSlackId 79, 93  
getStartTime 61  
getTimeCheck 45  
getTimestamp 48, 54, 56, 58, 72, 76, 88  
getUser 35  
getUserByEmail 19  
getUserByFirstName 19  
getUserByLastName 20  
getUserEmail 82, 86  
getUsername 117  
getValue 37  
GT 40

## H

hashCode 35, 66, 72, 81, 83  
hibernateProperties 5  
HomeController 15  
homePage 15  
HU 73  
Humidity 53  
humidityRepository 127

## I

id 34, 36, 42, 49, 67, 69  
Initializer 8  
isAccountNonExpired 117  
isAccountNonLocked 117  
isActive 51  
isAllHours 45  
isCredentialsNonExpired 117  
isEnabled 93, 117  
isFriday 44  
isMonday 44  
isSaturday 45  
isSunday 44  
isThursday 44  
isTuesday 44  
isUseEmail 51  
isUsePhone 51  
isUseSlack 51  
isWednesday 44

## J

javaMailService 7

## L

lastname 77, 91  
LE 40  
loadUserByUsername 137  
logger 119, 135  
login 16  
LoginController 16  
LT 40

## M

mailSender 119  
mapDoorData 131  
mapHumidityData 131  
mapMotionData 131  
mapNewUser 94  
mapNoPassword 94  
mapRABleData 131  
mapTemperatureData 130  
md5PasswordEncoder 9  
message 49  
messageSource 5  
MO 73  
modifier 36, 41  
monday 42  
Motion 55  
motionRepository 127

## N

name 49, 60, 67, 69  
NE 40  
newData 31  
newEvent 25

## O

onStartup 8  
owners 50

**P**

PA 73

password 77, 91, 116

passwordEncoder 141

phone 77, 91

PROPERTY\_NAME\_DATABASE\_DRIVER 3

PROPERTY\_NAME\_DATABASE\_PASSWORD 3

PROPERTY\_NAME\_DATABASE\_URL 3

PROPERTY\_NAME\_DATABASE\_USERNAME 3

PROPERTY\_NAME\_ENTITYMANAGER\_PACKAGES\_TO\_SCAN 4

PROPERTY\_NAME\_HIBERNATE\_DIALECT 3

PROPERTY\_NAME\_HIBERNATE\_HBML2DDL 4

PROPERTY\_NAME\_HIBERNATE\_SHOW\_SQL 3

PROPERTY\_NAME\_MAIL\_AUTH 6

PROPERTY\_NAME\_MAIL\_FROM 7

PROPERTY\_NAME\_MAIL\_HOST 6

PROPERTY\_NAME\_MAIL\_PASSWORD 6

PROPERTY\_NAME\_MAIL\_PORT 6

PROPERTY\_NAME\_MAIL\_PROTOCOL 6

PROPERTY\_NAME\_MAIL\_USERNAME 6

PROPERTY\_NAME\_SLACK\_WEBHOOK\_URL 12

propertyPlaceholderConfigurer 5

**R**

RA 73

raRepository 128

ReelyActiveBluetooth 57, 58

removeEvent 123

Report 60

ReportController 26

ReportFilterType 63

reportPage 17

ReportViewController 17

resetPassword 19, 139, 142

Role 65

role 65, 82

roleRepository 137

roles 78, 92

Room 63, 67

room 69

RoomApiController 28

roomDescription 87

roomName 87

roomRepository 125, 127

RoomService 125

roomService 28

rsi 57, 85

run 120

**S**

salt 77, 116

saturday 43

save 125, 130, 138, 141

saveUser 18

saveUsers 19

SecurityConfiguration 9

SecurityWebApplicationInitializer 11

sendMessage 115, 119, 135

sendNotification 120

Sensor 63, 69

sensor 29, 36

SensorApiController 29

SensorDataController 31

SensorDataDTO 87, 88

SensorDataPK 71

SensorDataService 128

sensorDataService 26

sensorId 47, 53, 55, 57, 71, 75, 87

sensorRepository 127, 133

SensorService 133

sensorService 29, 127

SensorType 63, 74

sensorType 69, 74, 87

serialVersionUID 34, 36, 42, 47, 49, 53, 55, 57, 65, 71, 75, 82, 116

setActive 52

setAllHours 46

setBeacon 58

setBeaconId 86

setData 48, 54, 56, 76, 89

setDateTime 46

setDescription 68

setEmail 80, 93

setEnabled 81, 94

setEvents 81, 94

setFirstname 80, 93  
setFriday 46  
setId 35, 37, 45, 51, 68, 70  
setLastname 80, 93  
setMessage 51  
setModifier 38  
setMonday 45  
setName 51, 68, 70  
setOwners 52  
setPassword 81, 94  
setPhone 80, 93  
setRole 65, 83  
setRoles 81, 94  
setRoom 70  
setRoomDescription 89  
setRoomName 89  
setRssi 59, 86  
setSalt 81  
setSaturday 46  
setSensor 37  
setSensorId 48, 54, 56, 58, 72, 76, 89  
setSensorType 70, 89  
setSlackId 80, 94  
setSunday 45  
setThursday 45  
setTimeCheck 46  
setTimestamp 48, 54, 56, 58, 72, 76, 90  
setTuesday 45  
setUseEmail 52  
setUsePhone 52  
setUser 35, 83  
setUserEmail 86  
setUseSlack 52  
setValue 38  
setWednesday 45  
setWeeklyOccurrences 38  
slackApi 12, 135  
SlackConfiguration 12  
slackId 77, 91  
SlackNotifier 135  
slackNotifier 120  
starttime 60  
sunday 42

## T

TE 73  
Temperature 75  
temperatureRepository 127  
thursday 43  
timeCheck 43  
timeEqual 38  
timestamp 47, 53, 55, 57, 71, 75, 87  
toHex 143  
transactionManager 5  
tuesday 42

## U

UK 74  
update 138, 141  
updateUser 18  
useEmail 49  
usePhone 50  
User 78  
user 34, 82  
userAllEvents 14  
UserController 18  
userDetailsAuth 9  
UserDetailsAuthService 137  
UserDTO 92  
userEmail 85  
username 116  
userRepository 137, 141  
UserRole 82  
userService 18  
UserServiceImpl 141  
useSlack 49

## V

value 36  
valueOf 41, 63, 74  
values 41, 63, 74  
viewResolver 5

## W

wednesday 42