Package event

Class Summary

Event

A run-time representation of an Event persistent Object.

EventBuilder

A Builder class which creates new Events.

EventList

A class that aggregates Events.

EventListBuilder

A Builder which creates new EventList objects.

EventLoader

The Class EventLoader.

EventManager

EventManager, which is a Singleton which manages all the Event functions.

---

event

Class Event

```
java.lang.Object
    |
    +--event.Event
```

---

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

A run-time representation of an Event persistent Object. This class is used as an intermediary for creation, retrieval, and modification of Event data within the Java code (and the JVM). It is encodable (or serializable) to a database format (e.g., SQL Entry).

Fields

date

`protected java.lang.String` **`date`**
The date.

---

description

`protected java.lang.String` **`description`**
The description.

---

eventId

`protected int` **`eventId`**
The event id.

---

eventType

`protected int` **`eventType`**
The event type.

---

hostedBy

`protected int` **`hostedBy`**
The hosted by.

---

isCancelled

`protected boolean` **`isCancelled`**
The is cancelled.

---

latCoordinate

`protected double` **`latCoordinate`**
The lat coordinate.

---

longCoordinate

`protected double` **`longCoordinate`**
The long coordinate.

---

name

`protected java.lang.String` **`name`**
The name.

---

time

`protected java.lang.String` **`time`**
The time.

---

visibility

`protected boolean` **`visibility`**
The visibility.

Constructors

Event

`protected` **`Event`**`()`

Constructs a new Event class. Called through the EventBuilder class. Attribute assignations are done through protected scope.

Event

```
protected  Event(java.sql.ResultSet results)
           throws java.lang.Exception
```

Creates an Event from a given ResultSet.

Parameters:

results - the input ResultSet

Throws:

java.lang.Exception - Thrown if the ResultSet is missing any event-defining variable.

Methods

getDate

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

Gets the date.

Returns:

the date

---

getDescription

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

Gets the description.

Returns:

the description

---

getEventID

```
protected int getEventID()
```

Gets the event ID.

Returns:

the event id

getEventType

`public int` **`getEventType`**`()`

Gets the event type.

Returns:

the eventType

---

getHostedBy

`public int` **`getHostedBy`**`()`

Gets the hosted by.

Returns:

the hostedBy

---

getJSON

`protected org.json.JSONObject` **`getJSON`**`()`

Gets the json.

Returns:

the json

---

getJsonTranslation

`public org.json.JSONObject` **`getJsonTranslation`**`()`

Gets the json translation.

Returns:

the jsonTranslation

---

getLatCoordinate

`public double` **`getLatCoordinate`**`()`

Gets the lat coordinate.

Returns:

the latCoordinate

---

getLongCoordinate

`public double` **`getLongCoordinate`**`()`

Gets the long coordinate.

Returns:

the longCoordinate

---

getName

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

Gets the name.

Returns:

the name

---

getTime

`public java.lang.String` **`getTime`**`()`

Gets the time.

Returns:

the time

---

isVisibility

`public boolean` **`isVisibility`**`()`

Checks if is visibility.

Returns:

the visibility

---

event

Class EventBuilder

```
java.lang.Object
    |
    +--event.EventBuilder
```

---

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

A Builder class which creates new Events. It is used to decouple the parts of the process of creating a new Event from the actual Event class, which is intended to only be a data wrapper class. Namely, this class implements the checks and validations necessary to create a valid Event and will reject invalid ones.

Constructors

EventBuilder

`public` **`EventBuilder`**`()`

Creates a new EventBuilder to instantiate the new Event.

Methods

build

```
public event.Event build()
```

Builds the event.

Returns:

the event if the build is complete, throws an error otherwise.

---

isNotComplete

```
public boolean isNotComplete()
```

Checks if is not complete.

Returns:

true if the event is not complete, false otherwise.

---

setCoordinates

```
public event.EventBuilder setCoordinates(double lat,
                                         double logn)
```

Sets the coordinates of the event location.

Parameters:

lat - the latitute of the location
logn - the longitude of the location

Returns:

the event builder.

---

setDate

```
public event.EventBuilder setDate(java.lang.String date)
```

Sets the date of the event.

Parameters:

date - the date of the event.

Returns:

the EventBuilder.

setDescription

```
public event.EventBuilder
setDescription(java.lang.String description)
```

Sets the description of the event.

Parameters:

description - the description of the event.

Returns:

the EventBuilder.

setEventType

```
public event.EventBuilder setEventType(int eventType)
```

Sets the type of the event.

Parameters:

eventType - the type of the event.

Returns:

the EventBuilder.

setHostedBy

```
public event.EventBuilder setHostedBy(int
organization_id)
```

Sets the id of the host.

Parameters:

organization_id - the id of the host.

Returns:

the EventBuilder.

setName

`public event.EventBuilder **setName**(java.lang.String name)`

Sets the name of the event.

Parameters:

name - the name of the event.

Returns:

the EventBuilder.

---

setTime

`public event.EventBuilder **setTime**(java.lang.String time)`

Sets the time of the event.

Parameters:

time - the time of the event.

Returns:

the EventBuilder.

---

setVisibility

`public event.EventBuilder **setVisibility**(boolean visibility)`

Sets the visibility of the event.

Parameters:

visibility - the visibility of the event.

Returns:

the EventBuilder.

event

Class EventList

```
java.lang.Object
    |
    +--event.EventList
```

---

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

A class that aggregates Events.

Constructors

EventList

```
protected  EventList()
```

Constructs a new EventList class. Called through the EventListBuilder class. Attribute assignations are done through protected scope.

EventList

```
protected  EventList(java.sql.ResultSet results)
           throws java.lang.Exception
```

Contrusts a new EventList from the contents of the ResultSet.

Parameters:

results - the ResultSet containing a collection of Event SQL entries.

Throws:

java.lang.Exception - thrown if errors occur while parsing the ResultSet

Methods

returnJSONList

`public org.json.JSONArray` **`returnJSONList`**`()`

Return JSON list.

Returns:

the JSONArray containing the list of Events

---

event

Class EventListBuilder

```
java.lang.Object
    |
    +--event.EventListBuilder
```

---

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

A Builder which creates new EventList objects. As other builders, it is used to decouple the process of creating a new EventList from the actual EventList class, and also provides functions implementing attribute-base filtering (e.g., filter by location, or by hosting organization, etc.)

Constructors

EventListBuilder

`public` **`EventListBuilder`**`()`

Creates a new EventListBuilder to instantiate the new EventList.

Methods

getAllAvailableEvents

```
public event.EventList
getAllAvailableEvents(java.sql.ResultSet set)
                      throws java.lang.Exception
```

Creates an EventList with all the events in the given result set.

Parameters:

set - the result set

Returns:

the EventList

Throws:

java.lang.Exception - the exception

---

event

Class EventLoader

```
java.lang.Object
    |
    +--event.EventLoader
```

---

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

The Class EventLoader.

Constructors

EventLoader

```
public  EventLoader()
```

Methods

loadEventDetails

```
public event.Event loadEventDetails(java.sql.ResultSet
results)
                    throws java.lang.Exception
```

Loads an Event from the given ResultSet.

Parameters:

results - the ResultSet

Returns:

the Event

Throws:

java.lang.Exception - Happens when the ResultSet

---

event

Class EventManager

```
java.lang.Object
    |
    +--event.EventManager
```

---

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

EventManager, which is a Singleton which manages all the Event functions. This class receives dispatched actions from the SOS Dispatcher and completes that action using objects internal to its subsystem. It also is in charge of interacting with the SOS Data Store Façade directly. Part of the role of this class is to parse front-end format data (e.g., JSON-String description of new Events) and calling the appropriate functions on other classes according to that data. It is also in charge of encoding Event objects into database-format (e.g., SQL Table entries). Another role is to create EventLists based on filter requests through the EventListBuilder.

Constructors

EventManager

```
protected  EventManager()
```

A protected or private constructor ensures that no other class has access to the Singleton.

Methods

cancelEvent

```
public org.json.JSONObject
cancelEvent(org.json.JSONObject payload)
```

Cancels the given event.

Parameters:

payload - The event that is going to be cancelled.

Returns:

the JSON object

---

createEvent

```
public org.json.JSONObject
createEvent(org.json.JSONObject json)
```

Creates a new Event from a json Event description. Done by calling the EventBuilder class.

Parameters:

json - the json

Returns:

the JSON object

---

getEventOfOrganization

```
public org.json.JSONObject
getEventOfOrganization(org.json.JSONObject payload)
```

Returns all the Events hosted by a given Organization.

Parameters:

payload - A JSONObject which containts the following keys: organization a JSONObject with the following keys: organization_id which is the id of the target organization.

Returns:

All events hosted by the organization.

---

instance

```
public static event.EventManager instance()
```

Instance.

Returns:

The unique instance of this class.

---

loadEventDetails

```
public org.json.JSONObject
loadEventDetails(org.json.JSONObject payload)
```

Gets event information based on the ID that is provided.

Parameters:

payload - The ID of the event that details are being requested.

Returns:

A JSON object with the event details.

---

markAttendance

```
public org.json.JSONObject
markAttendance(org.json.JSONObject payload)
```

Marks a User as attending an Event by creating an entry on the Attendance table.

Parameters:

payload - the payload

Returns:

the JSON object

---

retrieveListOfEvents

```
public org.json.JSONObject
retrieveListOfEvents(org.json.JSONObject payload)
```

Retrieves a list of events that are stored in the database.

Parameters:

payload - the payload

Returns:

A JSON array of events.

---

retrieveListOfEventsByLocation

```
public org.json.JSONObject
retrieveListOfEventsByLocation(org.json.JSONObject
payload)
```

Retrieves a list of Event given a location.

Parameters:

payload - A JSONObject with the following keys: lantitude the latitute of the search center longitude the longitude of the search center

Returns:

All events within 0.5 latitude/longitude range from the given center.

Package organization

Class Summary

Organization

A run-time representation of an Organization persistent object.

OrganizationBuilder

A Builder which creates new Organization objects.

OrganizationLoader

A class which creates an Organization object from an Organization database object.

OrganizationManager

A Singleton which manages all the Organization functions.

---

organization

Class Organization

```
java.lang.Object
    |
    +--organization.Organization
```

---

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

A run-time representation of an Organization persistent object. This class is used as an intermediary for creation, retrieval, and modification of Organization data within the Java code (and the JVM). It is encodable (or serializable) to a database format (e.g., SQL Entry)

Fields

description

`protected java.lang.String` **`description`**
The description.

---

name

`protected java.lang.String` **`name`**
The name.

---

privacy

`protected java.lang.String` **`privacy`**
The privacy.

---

requirements

`protected java.lang.String` **`requirements`**
The requirements.

Constructors

Organization

`protected` **`Organization`**`()`

Constructs a new Organization class. Called through the OrganizationBuilder class. Attribute assignations are done through protected scope.

Organization

`protected` **`Organization`**`(java.sql.ResultSet results) throws java.lang.Exception`

Creates an Organization from the target ResultSet.

Parameters:

results - the result set.

Throws:

java.lang.Exception - thrown if there's an error, like using the incorrect entry.

Methods

getDescription

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

Gets the description.

Returns:

the description

---

getJSONObject

`protected org.json.JSONObject` **`getJSONObject`**`()`

Gets the JSON object.

Returns:

the json translation of this Organization.

---

getJsonTranslation

`public org.json.JSONObject` **`getJsonTranslation`**`()`

Gets the json translation.

Returns:

the jsonTranslation

---

getName

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

Gets the name.

Returns:

the name

---

getPrivacy

`public java.lang.String` **`getPrivacy`**`()`

Gets the privacy.

Returns:

the privacy

---

getRequirements

`public java.lang.String` **`getRequirements`**`()`

Gets the requirements.

Returns:

the requirements

---

organization

Class OrganizationBuilder

```
java.lang.Object
    |
    +--organization.OrganizationBuilder
```

---

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

A Builder which creates new Organization objects. It is used to decouple the process, including validations and checks, of creating an Organization from the actual Organization class itself.

Fields

PRIVACY_PRIVATE

`public static final java.lang.String` **`PRIVACY_PRIVATE`**

The Constant PRIVACY_PRIVATE.

PRIVACY_PUBLIC

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

The Constant PRIVACY_PUBLIC.

Constructors

OrganizationBuilder

```
public  OrganizationBuilder()
```

Creates a new OrganizationBuilder to instantiate the new Event.

Methods

build

```
public organization.Organization build()
                                   throws
java.lang.IllegalArgumentException
```

Builds the Organization.

Returns:

an Organization object, or throws an error.

Throws:

java.lang.IllegalArgumentException - thrown if the organization being built is not complete.

---

isNotComplete

```
public boolean isNotComplete()
```

Checks if is not complete.

Returns:

true if the organization is not complete false otherwise.

---

setDescription

```
public organization.OrganizationBuilder
setDescription(java.lang.String description)
```

Sets the description.

Parameters:

description - the description.

Returns:

the OrganizationBuilder

---

setName

```
public organization.OrganizationBuilder
setName(java.lang.String name)
```

Sets the name.

Parameters:

name - the name.

Returns:

the OrganizationBuilder

---

setPrivacy

```
public organization.OrganizationBuilder
setPrivacy(java.lang.String privacy)
                                    throws
java.lang.IllegalArgumentException
```

Sets the privacy.

Parameters:

privacy - the privacy.

Returns:

the OrganizationBuilder

Throws:

java.lang.IllegalArgumentException - the illegal argument exception

---

setRequirements

```
public organization.OrganizationBuilder
setRequirements(java.lang.String requirements)
```

Sets the requirements.

Parameters:

requirements - the requirements.

Returns:

the OrganizationBuilder

---

organization

Class OrganizationLoader

```
java.lang.Object
    |
    +--organization.OrganizationLoader
```

---

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

A class which creates an Organization object from an Organization database object. This class decouples the parsing from the database to the system logic from the OrganizationManager class and can be extended to include internal checks for data integrity purposes.

Constructors

OrganizationLoader

```
public  OrganizationLoader()
```

Methods

LoadOrganization

```
public organization.Organization
LoadOrganization(java.sql.ResultSet set)
                                 throws
java.lang.Exception
```

Creates a Organization from a database-format entry.

Parameters:

set - the set

Returns:

a Organization object with the given attributes.

Throws:

java.lang.Exception - the exception

---

organization

Class OrganizationManager

```
java.lang.Object
    |
    +--organization.OrganizationManager
```

---

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

A Singleton which manages all the Organization functions. This class receives dispatched actions from the SOS Dispatcher and completes that action using objects internal to its subsystem. It also is in charge of interacting with the SOS Data Store Façade directly. Part of the role of this class is to parse front-end format data (e.g., JSON-String description of new Organization) and calling the appropriate functions on other classes according to that data. Another job of this class is to manage Role creation and assignment, as well as mediate the modification of data in an Organization, and of Event hosting.

Constructors

OrganizationManager

```
protected  OrganizationManager()
```

A protected or private constructor ensures that no other class has access to the Singleton.

Methods

createOrganization

```
public org.json.JSONObject
createOrganization(org.json.JSONObject json)
```

Creates an organization in the SOS System.

Parameters:

json - the json

Returns:

the JSON object

---

getAllOrganizations

```
public org.json.JSONObject getAllOrganizations()
```

Gets all the public organizations in the SOS.

Returns:

A JSONArray with all the public organizations stored in the SOS.

---

getAllOrganizations

```
public org.json.JSONObject
getAllOrganizations(org.json.JSONObject payload)
```

Gets all the organizations which a user currently belongs to.

Parameters:

payload - the payload

Returns:

A JSONObject with all the organizations the user is a part of.

---

grantRole

```
public org.json.JSONObject grantRole(org.json.JSONObject
payload)
```

Grants a number of privileges to a User for a given Organization.

Parameters:

payload - the payload

Returns:

the JSON object

---

instance

```
public static organization.OrganizationManager
instance()
```

Instance.

Returns:

The unique instance of this class.

---

joinOrganization

```
public org.json.JSONObject
joinOrganization(org.json.JSONObject payload)
```

Allows the user to join an organization that is part of SOS.

Parameters:

payload - the payload

Returns:

the JSON object

---

loadOrganizationDetails

```
public org.json.JSONObject
loadOrganizationDetails(org.json.JSONObject payload)
```

Loads details for the requested organization.

Parameters:

payload - the payload

Returns:

the JSON object

Package security

Class Summary

AccessManager

A Singleton dealing with access control actions.

PasswordManager

A Singleton which deals with password control actions.

TransferManager

A Singleton that handles secure data exchange between the front end and the back end.

---

security

Class AccessManager

```
java.lang.Object
    |
    +--security.AccessManager
```

---

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

A Singleton dealing with access control actions. It implements most of the back-end side of the access policy for SOS and host the relevant Enumerations for access permissions and other privileges. It also must be called to do checks on the relevant actions, such as creating events, deleting profiles, etc.

Constructors

AccessManager

`protected` **`AccessManager`**`()`

A protected or private constructor ensures that no other class has access to the Singleton.

Methods

CheckPrivileges

`public boolean` **`CheckPrivileges`**`()`

Check privileges.

Returns:

The result of privilege check for the current user class.

---

instance

`public static security.AccessManager` **`instance`**`()`

Instance.

Returns:

The unique instance of this class.

---

security

Class PasswordManager

```
java.lang.Object
    |
    +--security.PasswordManager
```

---

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

A Singleton which deals with password control actions. It implements most of the back-end side of the password policy for SOS, including resolving passwords and checking the input password against the database.

Constructors

PasswordManager

`protected` **`PasswordManager`**`()`

The constructor could be made private to prevent others from instantiating this class. But this would also make it impossible to create instances of PasswordManager subclasses.

Methods

HashPassword

```
public static java.lang.String
HashPassword(java.lang.String username,
java.lang.String password)
```

Hash password.

Parameters:

username - the username
password - is a String to be validated

Returns:

will return an encrypted version of the password as a String

ValidateLogInCredentials

```
public static boolean ValidateLogInCredentials(user.User
user,
java.lang.String pwd)
```

Validate log in credentials.

Parameters:

user - the user
pwd - is the user's password for log in

Returns:

is the validation of the login credentials

ValidatePassword

```
public static boolean ValidatePassword(java.lang.String
password)
```

Validate password.

Parameters:

password - as a String to be validated

Returns:

is true if password successfully validates

instance

```
public static security.PasswordManager instance()
```

Instance.

Returns:

The unique instance of this class.

---

security

Class TransferManager

```
java.lang.Object
    |
    +--security.TransferManager
```

---

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

A Singleton that handles secure data exchange between the front end and the back end.

Methods

decryptMessage

```
public java.lang.String
decryptMessage(org.json.JSONObject msg)
```

Decrypts the message in the given JSONObject.

Parameters:

msg - a json object which must have: key a symmetric key encrypted with this TransferManager's public certificate. iv an iv value encrypted with this TransferManager's public certificate. text a cyphertext encrypted using AES/CBC/PKCS5Padding and the given key and iv.

Returns:

the plaintext form of text.

---

encryptMessage

```
public org.json.JSONObject
encryptMessage(java.lang.String msg,
java.lang.String alias)
```

Produces an Base64 version of the encrypted ciphertext for the input given in msg.

Parameters:

msg - the input to be encrypted and encoded.
alias - the alias

Returns:

the JSONObject containing: 'key' the encrypted key parameter, decryptable with the target's private key. 'iv' the encrypted iv parameter, decryptable with the target's private key. 'text' the encrypted text, decrypateble with a AES/CBC/PKCS7Padding using the given key and iv.

---

getSharableCertificate

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

Gets the sharable (PEM) string version of this object's certificate.

Returns:

a String containing the PEM version of the certificate.

---

instance

```
public static security.TransferManager instance()
```

Instance.

Returns:

The unique instance of this class.

---

setCertificateEntry

```
public void setCertificateEntry(java.lang.String certS,
                                java.lang.String alias)
```

Adds an external certificate to the KeyStore with the given alias.

Parameters:

certS - the certificate, usually in a PEM format.
alias - the alias for the certificate.

Package sosInterface

Class Summary

SOSCommand

The Class SOSCommand.

SOSDispatcher

The Class SOSDispatcher.

SOSDispatcher.REQUEST_TYPES

The Enum REQUEST_TYPES.

SOSServer

SOSServer communicates with the front-end for creation of events.

SOSServer_Driver

The Class SOSServer_Driver.

---

sosInterface

Class SOSCommand

```
java.lang.Object
    |
    +--sosInterface.SOSCommand
```

---

public abstract class **SOSCommand**
extends java.lang.Object

The Class SOSCommand.

Fields

client

```
protected com.corundumstudio.socketio.SocketIOClient
client
```

The client.

---

errorPayload

`protected org.json.JSONObject` **`errorPayload`**

The error payload.

---

errorStatus

`protected java.lang.String` **`errorStatus`**

The error status.

Constructors

SOSCommand

`protected` **`SOSCommand`**`(com.corundumstudio.socketio.SocketIOClient client)`

Creates an SOSCommand Object which will report to the given client.

Parameters:

client - the client for this SOSCommand.

Methods

createCommand

```
public static sosInterface.SOSCommand
createCommand(sosInterface.SOSDispatcher.REQUEST_TYPES
request,
com.corundumstudio.socketio.SocketIOClient client,
org.json.JSONObject payload)
```

Creates a Command subclass which implements one of the commands of the server. The list of commands can be seen in the SOSDispatcher.REQUEST_TYPES enumeration. Each subclass implements the execute function which instantiates and calls the relevant action on the managers of the relevant classes.

Parameters:

request - the request type.
client - the client to be passed.
payload - the payload of the request.

Returns:

the SOSCommand object implementing the dispatchable action.

---

errorStatus

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

Returns the stored error status, which is set by the execute function in case of errors.

Returns:

the string

---

execute

```
public abstract boolean execute()
                          throws
java.lang.RuntimeException
```

Executes the command. Must be implemeted by subclasses.

Returns:

true if the command executed successfully, false otherwise.

Throws:

java.lang.RuntimeException - the runtime exception

failWith

```
protected void failWith(org.json.JSONObject
errorPayload)
```

Reports a failure to the client, with the given payload.

Parameters:

errorPayload - the failure body.

succeedWith

```
protected void succeedWith(org.json.JSONObject
successPayload)
```

Reports a success to the client, with the given payload.

Parameters:

successPayload - the payload of the success.

sosInterface

Class SOSDispatcher

```
java.lang.Object
    |
    +--sosInterface.SOSDispatcher
```

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

The Class SOSDispatcher.

Constructors

SOSDispatcher

```
protected  SOSDispatcher()
```

Creates new dispatcher.

Methods

dispatch

```
public void
dispatch(sosInterface.SOSDispatcher.REQUEST_TYPES
request,
com.corundumstudio.socketio.SocketIOClient client,
                    org.json.JSONObject payload)
```

Dispatches an action by creating and executing an SOSCommand.

Parameters:

request - the request.
client - the client to return the request action.
payload - the payload of the request.

---

getInstance

```
public static sosInterface.SOSDispatcher getInstance()
```

Returns the Singleton dispatcher instance.

Returns:

the unique SOSDispatcher.

---

sosInterface

Class SOSDispatcher.REQUEST_TYPES

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--sosInterface.SOSDispatcher.REQUEST_TYPES
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

---

public static final class **SOSDispatcher.REQUEST_TYPES**
extends java.lang.Enum

The Enum REQUEST_TYPES.

Fields

ATTEND_EVENT

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES ATTEND_EVENT
```

The attend event.

CREATE_EVENT

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES CREATE_EVENT
```

The create event.

CREATE_ORG

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES CREATE_ORG
```

The create org.

CREATE_USER

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES CREATE_USER
```

The create user.

EVENT_CANCEL

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES EVENT_CANCEL
```

The event cancel.

JOIN_ORG

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES JOIN_ORG
```

The join org.

LOAD_USER

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES LOAD_USER
```

The load user.

LOGIN

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES LOGIN
```

The login.

RETR_ALL_EVENTS

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES RETR_ALL_EVENTS
```

The retr all events.

RETR_EVENT

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES RETR_EVENT
```

The retr event.

RETR_EVENTS_BY_LOCATION

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES
RETR_EVENTS_BY_LOCATION
```

The retr events by location.

RETR_EVENTS_FOR_ORG

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES
RETR_EVENTS_FOR_ORG
```

The retr events for org.

RETR_EVENT_FOR_ORG

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES
RETR_EVENT_FOR_ORG
```

The retr event for org.

RETR_MEMBER_FOR_ORG

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES
RETR_MEMBER_FOR_ORG
```

The retr member for org.

RETR_ORG

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES RETR_ORG
```

The retr org.

RETR_ORGS

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES RETR_ORGS
```

The retr orgs.

RETR_ORGS_FOR_USER

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES
RETR_ORGS_FOR_USER
```

The retr orgs for user.

SET_ROLE

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES SET_ROLE
```

The set role.

UPDATE_USER

```
public static final
sosInterface.SOSDispatcher.REQUEST_TYPES UPDATE_USER
```
The update user.

Methods

valueOf

```
public static sosInterface.SOSDispatcher.REQUEST_TYPES
valueOf(java.lang.String name)
```

values

```
public static sosInterface.SOSDispatcher.REQUEST_TYPES[]
values()
```

sosInterface

Class SOSServer

```
java.lang.Object
    |
    +--sosInterface.SOSServer
```

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

SOSServer communicates with the front-end for creation of events. Also it is held responsible for managing user sessions and keeping track of them, as well as dispatching messages through the system.

Methods

ListenForEvents

```
public void ListenForEvents()
```

Starts the server and sets it to listen for events from a client socket.io front-end.

---

instance

```
public static sosInterface.SOSServer instance()
                                     throws
java.lang.Exception
```

Instance.

Returns:

The unique instance of this class.

Throws:

java.lang.Exception - the exception

---

sosInterface

Class SOSServer_Driver

```
java.lang.Object
    |
    +--sosInterface.SOSServer_Driver
```

---

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

The Class SOSServer_Driver.

Constructors

SOSServer_Driver

```
public  SOSServer_Driver()
```

Methods

main

```
public static void main(java.lang.String[] args)
```

The main method.

Parameters:

args - the arguments

Package sosInterface.socket

Class Summary

SOSConnectListener

The listener interface for receiving SOSConnect events.

SOSEventListener

Internal class extending a Socket.IO class to wrap the whole connection within an encryption mechanism.

---

sosInterface.socket

Class SOSConnectListener

```
java.lang.Object
    |
    +--sosInterface.socket.SOSConnectListener
```

All Implemented Interfaces:

com.corundumstudio.socketio.listener.ConnectListener

---

public class **SOSConnectListener**
extends java.lang.Object
implements com.corundumstudio.socketio.listener.ConnectListener

The listener interface for receiving SOSConnect events. The class that is interested in processing a SOSConnect event implements this interface, and the object created with that class is registered with a component using the component's `addSOSConnectListener` method. When the SOSConnect event occurs, that object's appropriate method is invoked.

SOSConnectEvent

Constructors

SOSConnectListener

`public` **`SOSConnectListener`**`()`

Methods

onConnect

```
public void
onConnect(com.corundumstudio.socketio.SocketIOClient
client)
```

On connect.

Parameters:

client - the client

---

sosInterface.socket

Class SOSEventListener

```
java.lang.Object
    |
    +--sosInterface.socket.SOSEventListener
```

All Implemented Interfaces:
com.corundumstudio.socketio.listener.DataListener

---

public abstract class **SOSEventListener**
extends java.lang.Object
implements com.corundumstudio.socketio.listener.DataListener

Internal class extending a Socket.IO class to wrap the whole connection within an encryption mechanism.

SOSEventEvent

Constructors

SOSEventListener

```
public  SOSEventListener()
```

Methods

doOnData

```
public abstract void
doOnData(com.corundumstudio.socketio.SocketIOClient
client,
                                org.json.JSONObject json,
com.corundumstudio.socketio.AckRequest ackRequest)
```

Implemented by the event-specific data listener so it can work afte decrypting. For more information, read the javadoc for the netty-socket.io DataListener<>() class.

Parameters:

client - the client
json - the json
ackRequest - the ack request

---

doSendEvent

```
public static void
doSendEvent(com.corundumstudio.socketio.SocketIOClient
client,
                                 org.json.JSONObject data)
```

Encrypts the given data and sends it to the client.

Parameters:

client - the client target.
data - the plaintext JSON data.

---

onData

```
public void
onData(com.corundumstudio.socketio.SocketIOClient
client,
                    java.lang.String cipher,
com.corundumstudio.socketio.AckRequest ackRequest)
```

Decrypts on data.

Parameters:

client - the client
cipher - the cipher
ackRequest - the ack request

Package storage

Class Summary

DataStoreFacade

The Class DataStoreFacade.

DataStoreFacade_Driver

The Class DataStoreFacade_Driver.

---

storage

Class DataStoreFacade

```
java.lang.Object
    |
    +--storage.DataStoreFacade
```

---

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

The Class DataStoreFacade.

Constructors

DataStoreFacade

```
public  DataStoreFacade()
        throws java.lang.Exception
```

Attempts to connect to the Database.

Throws:

java.lang.Exception - Throws an exception if the database connection fails.

Methods

addNewRoleToOrganization

```
public void addNewRoleToOrganization(java.lang.String
roleName,
                                     int organizationID,
                                     int userID,
                                     boolean[]
privileges)
             throws java.lang.Exception
```

Adds a new role to the organization provided and assigns the role to the user provided.

Parameters:

roleName - The name that is given to the role.
organizationID - The organization ID of the organization that the role belongs to.
userID - The user ID of the user that will own the role.
privileges - The list of privileges granted to the user for their particular role.

Throws:

java.lang.Exception - An exception is thrown when the new role fails to be stored in the SOS database.

---

cancelEvent

```
public void cancelEvent(int eventID)
             throws java.lang.Exception
```

Requests to cancel the event in the database.

Parameters:

eventID - The ID of the event that needs to be cancelled.

Throws:

java.lang.Exception - Throws an exception if the event could not be cancelled.

---

createNewEvent

```
public void createNewEvent(event.Event event)
            throws java.lang.Exception
```

Creates a new event in the SOS system.

Parameters:

event - the event

Throws:

java.lang.Exception - Throws an exception if the parameters are not in the expected format and if the organization hosting the event no longer exists.

---

createNewOrganization

```
public void
createNewOrganization(organization.Organization org,
                                int userID)
             throws java.lang.Exception
```

Creates a new organization on the SOS system.

Parameters:

org - the org
userID - The user ID of the user creating the organization.

Throws:

java.lang.Exception - Throws an exception if there is an issue storing the organization into the database.

---

filterEventsByLocation

```
public java.sql.ResultSet filterEventsByLocation(double
lat_coordinate,
                                                 double
long_coordinate)
                          throws java.lang.Exception
```

Returns a list of JSON objects .

Parameters:

lat_coordinate - The latitude of the location of interest.
long_coordinate - The longitude of the location of interest.

Returns:

The results from the database of the closest events.

Throws:

java.lang.Exception - An exception is thrown if their is a problem retrieving nearby events.

---

getEvents

```
public java.sql.ResultSet getEvents()
                          throws java.lang.Exception
```

Gets all the events in the database that are not cancelled.

Returns:

A result set with the events found in the database.

Throws:

java.lang.Exception - An exception is thrown when the procedure fails to retrieve the results from the database.

---

getEventsByOrganization

```
public java.sql.ResultSet getEventsByOrganization(int
orgID)
                          throws java.lang.Exception
```

Returns the Events of the given Organization.

Parameters:

orgID - the id of the organization.

Returns:

the ResultSet containing the event entries.

Throws:

java.lang.Exception - the exception

---

getEventsByUser

```
public java.sql.ResultSet getEventsByUser(int userID)
                          throws java.lang.Exception
```

Returns the Events attended by the given user.

Parameters:

userID - the user id

Returns:

a ResultSet containing the target events

Throws:

java.lang.Exception - the exception

---

joinOrganization

```
public void joinOrganization(int userID,
                             int organizationID)
              throws java.lang.Exception
```

Allows the user to join an organization.

Parameters:

userID - The ID of the user that wants to join an organization.
organizationID - The ID of the organization that the user wants to join.

Throws:

java.lang.Exception - Throws an exception if the user tries to join an organization they already belong to or does not exist.

---

registerNewUser

```
public void registerNewUser(user.User user)
```

Registers a new user for the SOS system.

Parameters:

user - the user

---

retrieveEventDetails

```
public java.sql.ResultSet retrieveEventDetails(int
eventID)
                           throws java.lang.Exception
```

Retrieves all the details for a certain event.

Parameters:

eventID - The ID of the event that we want the details for.

Returns:

The details of the events in the form of a result set.

Throws:

java.lang.Exception - Throws an exception if the event details were not found.

---

retrieveMembersOfOrganization

```
public java.sql.ResultSet
retrieveMembersOfOrganization(int organization_id)
                              throws java.lang.Exception
```

Retrives all the users which are members of a given organization.

Parameters:

organization_id - the id of the organization.

Returns:

the ResultSet containing the user entries.

Throws:

java.lang.Exception - the exception

---

retrieveOrganizationDetails

```
public java.sql.ResultSet
retrieveOrganizationDetails(int organizationID)
                              throws java.lang.Exception
```

Retrieves the information of the organization that is stored in the database.

Parameters:

organizationID - The ID of the organization that the details were requested for.

Returns:

The set of details found in the database.

Throws:

java.lang.Exception - Throws an exception if there is a problem retrieving the details for the specified information.

---

retrieveOrganizationsForUser

```
public java.sql.ResultSet
retrieveOrganizationsForUser(int userID)
                          throws java.lang.Exception
```

Retrieves all the organizations which the user belongs to,.

Parameters:

userID - The ID of the user that we want all the organizations for.

Returns:

A set of organizations which the user belongs to within the SOS.

Throws:

java.lang.Exception - Throws an exception if their is an error with the connectivity to the storage of the system.

---

retrievePublicOrganizations

```
public java.sql.ResultSet retrievePublicOrganizations()
                          throws java.lang.Exception
```

Retrieves all of the public organizations stored in the storage.

Returns:

The set of all the public organizations located in the storage.

Throws:

java.lang.Exception - Throws an exception if their was an issue retrieving all of the organizations from the storage.

---

retrieveUserByUsername

```
public java.sql.ResultSet
retrieveUserByUsername(java.lang.String username)
                       throws java.lang.Exception
```

Retrieves an user entry from its unique username. User for login in moslty.

Parameters:

username - the username of the user

Returns:

the result set containing the user entry.

Throws:

java.lang.Exception - the exception

---

retrieveUserDetails

```
public java.sql.ResultSet retrieveUserDetails(int
userID)
                          throws java.lang.Exception
```

Retrieves the details from a given user stored in the DB.

Parameters:

userID - The ID for the user that the details are requested for.

Returns:

A set of details found in the storage.

Throws:

java.lang.Exception - Throws an exception if their is an issue connecting to the database.

---

saveUserAttendance

```
public void saveUserAttendance(long userID,
                               long eventID)
            throws java.lang.Exception
```

Saves the attendance of the user to a particular event in the SOS storage.

Parameters:

userID - The ID of the user that is attending the event.
eventID - The ID of the event that the user is attending.

Throws:

java.lang.Exception - Throws an exception if there is an issue in storing the attendance of the user in the database.

---

terminateConnection

```
public void terminateConnection()
            throws java.lang.Exception
```

Terminates connection to the database.

Throws:

java.lang.Exception - Throws an exception if database connection cannot be closed.

---

updateUserInformation

```
public void updateUserInformation(user.User user)
            throws java.lang.Exception
```

Updates the information of the user to the given information.

Parameters:

user - the user

Throws:

java.lang.Exception - If the email is already present in the database under a different user then an exception is thrown.

---

verifyUserLogin

```
public boolean verifyUserLogin(java.lang.String email,
                               java.lang.String
password)
                throws java.lang.Exception
```

Verifies the login of the user.

Parameters:

email - The email that the user provides when logging in.
password - The encrypted password the user provides when logging in.

Returns:

A boolean verifying if the user has correct credentials to log into SOS.

Throws:

java.lang.Exception - Throws an exception if the credentials are in an invalid format.

---

storage

Class DataStoreFacade_Driver

```
java.lang.Object
    |
    +--storage.DataStoreFacade_Driver
```

---

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

The Class DataStoreFacade_Driver.

Constructors

DataStoreFacade_Driver

```
public  DataStoreFacade_Driver()
```

Methods

main

```
public static void main(java.lang.String[] args)
```

The main method.

Parameters:

args - the arguments

Package user

Class Summary

NewUserBuilder

A Builder which creates new User objects.

User

A run-time representation of a User persistent object.

UserLoader

A class which creates a User object from a database-format User object (e.g., a SQL Table entry for User).

UserManager

A Singleton class which managers all the User functions.

UserUpdater

A class which deals with User modifications.

---

user

Class NewUserBuilder

```
java.lang.Object
    |
    +--user.NewUserBuilder
```

---

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

A Builder which creates new User objects. It is used to decouple the parts of the process of creating a new User from the actual User class, which is intended to only be a data wrapper class which can be easily parsed into the database format. Namely, this class implements the checks and validations necessary to create a valid User and will reject invalid ones. As part of this validation, it must interact with the SOS Security System classes that implement the password and access policies.

Constructors

NewUserBuilder

`public` **`NewUserBuilder`**`()`

Creates a new NewUserBuilder to instantiate the new User.

Methods

build

`public user.User` **`build`**`()`
`throws`
`java.lang.IllegalArgumentException`

Builds the User.

Returns:

an User object, or throws an error.

Throws:

java.lang.IllegalArgumentException - thrown if the organization being built is not complete.

---

isNotComplete

`public boolean` **`isNotComplete`**`()`

Checks if is not complete.

Returns:

true if the user is not complete false otherwise.

---

setEmail

`public user.NewUserBuilder **setEmail**(java.lang.String setEmail)`

Sets the email.

Parameters:

setEmail - the email

Returns:

the NewUserBuilder

---

setName

`public user.NewUserBuilder **setName**(java.lang.String name)`

Sets the name.

Parameters:

name - the name

Returns:

the NewUserBuilder

---

setPassword

`public user.NewUserBuilder **setPassword**(java.lang.String password)`

Sets the password.

Parameters:

password - the password

Returns:

the NewUserBuilder

---

setPrivacy

```
public user.NewUserBuilder setPrivacy(java.lang.String privacy)
```

Sets the privacy.

Parameters:

privacy - the privacy

Returns:

the NewUserBuilder

---

setUsername

```
public user.NewUserBuilder setUsername(java.lang.String username)
```

Sets the name.

Parameters:

username - the username

Returns:

the NewUserBuilder

---

user

Class User

```
java.lang.Object
    |
    +--user.User
```

---

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

A run-time representation of a User persistent object. This class is used as an intermediary for creation, retrieval, and modification of User data within the Java code (and the JVM). It is encodable (or serializable) to a database format (e.g., SQL Entry).

Fields

email

`protected java.lang.String` **`email`**
The email.

---

name

`protected java.lang.String` **`name`**
The name.

---

password

`protected java.lang.String` **`password`**
The password.

---

privacy

`protected java.lang.String` **`privacy`**
The privacy.

---

userName

`protected java.lang.String` **`userName`**
The user name.

---

user_id

`protected int` **`user_id`**
The user id.

Constructors

User

`protected` **`User`**`()`

Creates an empty user object, for the Builder.

User

```
protected  User(java.sql.ResultSet set)
           throws java.lang.Exception
```

Constructs a new User class. Called through the UserBuilder class. Attribute assignations are done through protected scope.

Parameters:

set - the set

Throws:

java.lang.Exception - the exception

Methods

getEmail

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

Gets the email.

Returns:

the email

getJSON

```
public org.json.JSONObject getJSON()
```

Returns the JSON form of this User.

Returns:

the json

getName

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

Gets the name.

Returns:

the name

getPassword

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

Gets the password.

Returns:

the password

getPrivacy

`public java.lang.String` **`getPrivacy`**`()`

Gets the privacy.

Returns:

the privacy

getUserName

`public java.lang.String` **`getUserName`**`()`

Gets the user name.

Returns:

the userName

getUser_id

`public int` **`getUser_id`**`()`

Gets the user id.

Returns:

the user_id

user

Class UserLoader

```
java.lang.Object
    |
    +--user.UserLoader
```

---

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

A class which creates a User object from a database-format User object (e.g., a SQL Table entry for User). This class decouples the parsing from the database to the system logic from the UserManager class and can be extended to include internal checks for data integrity purposes.

Constructors

UserLoader

```
public  UserLoader()
```

Instantiates a new user loader.

Methods

LoadUser

```
public user.User LoadUser(java.sql.ResultSet results)
                  throws java.lang.Exception
```

Creates a User from a database-format entry.

Parameters:

results - The set of details found in the storage of the SOS.

Returns:

a User object with the given attributes.

Throws:

java.lang.Exception - the exception

---

user

Class UserManager

```
java.lang.Object
    |
    +--user.UserManager
```

---

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

A Singleton class which managers all the User functions. This class receives dispatched actions from the SOS Dispatcher and completes that action using objects internal to its subsystem. It also is in charge of interacting with the SOS Data Store Façade directly. Part of the role of this class is to parse front-end format user data (e.g., JSON-String defining a new User) and calling the appropriate functions on the other classes according to that data. It also is in charge of encoding a User object into database format objects (e.g., SQL Table entry for User).

Constructors

UserManager

`protected` **`UserManager`**`()`

A protected or private constructor ensures that no other class has access to the Singleton.

Methods

ChangeUserDetails

```
public org.json.JSONObject
ChangeUserDetails(org.json.JSONObject json)
                          throws java.lang.Exception
```

Changes the details of the user in the SOS system.

Parameters:

userID - The ID of the user that wants to change their information.
json2 - The JSON string with the user information and their changes.

Throws:

java.lang.Exception - Throws an exception if an error occurs while attempting to change user information.

CreateNewProfile

```
public org.json.JSONObject
CreateNewProfile(org.json.JSONObject json)
                          throws java.lang.Exception
```

Creates a new profile when the user registers to the SOS site.

Parameters:

input - A JSON string representing the user's information.

Throws:

java.lang.Exception - Throws an exception if there was an issue creating the user's profile.

LoadUser

```
public org.json.JSONObject LoadUser(org.json.JSONObject
payload)
```

Creates a User from a database-format entry. Done by calling the UserLoader class.

Parameters:

payload - The ID of the user that we want

Returns:

a User object with the given attributes.

getMembersOfOrganization

```
public org.json.JSONObject
getMembersOfOrganization(org.json.JSONObject payload)
```

Returns all the members of a given organization.

instance

```
public static user.UserManager instance()
```

Gives the instance of the UserManager, or creates one if none exists.

Returns:

the unique instance of this class.

login

```
public org.json.JSONObject login(org.json.JSONObject
payload)
```

Checks if the parameters given by a log-in attempt are valid, and returns the user information if so.

main

```
public static void main(java.lang.String[] args)
```

user

Class UserUpdater

```
java.lang.Object
    |
    +--user.UserUpdater
```

public class **UserUpdater**

extends java.lang.Object

A class which deals with User modifications. User modifications are done on the system logic-level User object first and are only finalized once they are stored to the database. The UserUpdater decouples these modifications from the UserManager class and from the User class itself and implements checks and validations in the same way that NewUserBuilder does. It also ensures that every modification to the User class is saved to the SOS Data Store.

Constructors

UserUpdater

```
public  UserUpdater()
```

Methods

ChangeUser

```
public void ChangeUser(user.User user,
                       java.util.Map update)
```

Updates a User object with the given changes.

Parameters:

user - the User that will be updated.
update - the update

---

makeUpdatesMap

```
public java.util.Map makeUpdatesMap(org.json.JSONObject
json)
```

Creates an update-map from a json payload.

Parameters:

json - the json

Returns:

the map

Package utils

Class Summary

Constants

The Class Constants.

JSONTranslator

The Class JSONTranslator.

---

utils

Class Constants

```
java.lang.Object
    |
    +--utils.Constants
```

---

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

The Class Constants.

Fields

DB_HOSTNAME

`public static final java.lang.String` **`DB_HOSTNAME`**
The Constant DB_HOSTNAME.

DB_PORT

`public static final int` **`DB_PORT`**
The Constant DB_PORT.

SERVER_HOSTNAME

`public static final java.lang.String` **`SERVER_HOSTNAME`**
The Constant SERVER_HOSTNAME.

SERVER_PORT

`public static final int` **`SERVER_PORT`**
The Constant SERVER_PORT.

Constructors

Constants

`public` **`Constants`**`()`

utils

Class JSONTranslator

```
java.lang.Object
    |
    +--utils.JSONTranslator
```

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

The Class JSONTranslator.

Constructors

JSONTranslator

`public` **`JSONTranslator`**`()`

Methods

resultSetToJSONArray

```
public static org.json.JSONArray
resultSetToJSONArray(java.sql.ResultSet set)
                                    throws
java.lang.Exception
```

Result set to JSON array.

Parameters:

set - the set

Returns:

the JSON array

Throws:

java.lang.Exception - the exception

---

resultSetToJSONObject

```
public static org.json.JSONObject
resultSetToJSONObject(java.sql.ResultSet set)
                                     throws
java.lang.Exception
```

Result set to JSON object.

Parameters:

set - the set

Returns:

the JSON object

Throws:

java.lang.Exception - the exception