The SELIS Publish/Subscribe Java Library

The SELIS Publish/Subscribe Java Library allows a Java client to interact with the SELIS Publish/Subscribe, by establishing and maintaining a TLS connection to the the SELIS Publish/Subscribe. It allows Java client to publish and receive messages by subscribing for Messages fulfilling the custom criteria.

Requirements

The SELIS Publish/Subscribe Java library require **at least JDK7** for compilation and runtime. Additionally it requires access to the running SELIS Publish/Subscribe system; for testing:

- locally (requires manual deployment on local machine): https://localhost:20000,
- hosted in ICCS, used for integration and testing: https://selis-dev-pubsub.cslab.ece.ntua.gr:20000.

The root certificate is required in order to verify the identity of the SELIS Publish/Subscribe system. The example Java application provides root.crt file which can be used for tests with the PubSub running locally (localhost).

Version & dependencies

Current implementation of the SELIS Publish/Subscribe Java Library is

0.3.0.

The SELIS Publish/Subscribe Java Library has the following runtime dependencies:

GroupId	ArtifactId	Version
com.google.code.gson	gson	2.8.0
org.slf4j	slf4j-api	1.7.5

Download

The SELIS Publish/Subscribe Java library it is not publicly accessible for downloading. It is neither published in any public Maven repository.

The library is available in compiled form as .JAR archive which should be added as a dependency to the Java project.

API Guide

All API classess are located in the top-level package: **de.tu_dresden.selis.pubsub**. The main classes/interfaces used by client application are:

- PubSub
- Message
- Callback
- Subscription

In the following sections, it is be described how to establish the connection to the SELIS Publish/Subscribe system using PubSub class, how to publish new Messages and how to subscribe for Messages creating custom Subscriptions and registering Callbacks.

Connecting to the SELIS Publish/Subscribe system

To connect to the SELIS Publish/Subscribe system an instance of PubSub class has to be created using IP/Hostname and Port number of the Pub/Sub. The first argument is the file location of the Root CA certificate which was used to sign the certificate of the running instance of the SELIS Publish/Subscribe system.

```
PubSub pubSub = new PubSub("SelisRootCA.pem", "0.0.0.0", 2000
0)
```

Once the PubSub is created, it establishes and keeps open the SSL connections to the SELIS Publish/Subscribe system. Once the connection is no longer required it is important that the client application explicite closes it, in order to clean all resources and close the connections.

```
pubSub.close();
```

PubSub class implements the autoCloseable interface, which allows to close the PubSub and release all the resources automatically. To use it,

create a PubSub in a try() block:

```
try (PubSub pubSub = new PubSub("0.0.0.0", 20000)) {
    // publish/subscribe messages
}
// here, Java automatically closed the PubSub
```

Publish

Once the connection to the SELIS Publish/Subscribe system has been established, client can start publishing messages.

Messages accepted by the SELIS Publish/Subscribe system have to be in the form of key:value pairs. To simplify the creation of messages, the SELIS Publish/Subscribe Java Library provides a Message class, which can be filled with the data and send directly to the SELIS Publish/Subscribe system.

```
Message msg = new Message();
msg.put("OrderId", "ABC123");
msg.put("price", "100");
msg.put("quality", "0.80");
```

The Subscription class allows to quickly define the subscription criteria. To create the subscription it is required that the client’s authentication token is provided. The SELIS Publish/Subscribe system uses the authentication token to verify the identity of the subscriber and fetch his access rules

from SELIS authorization service. The access rules are applied internally limiting the messages forwarded to the subscriber. The client's authentication token has to be first obtained from the SSO service.

```
Subscription subscription = new Subscription("d5644e8105ad77c
3c3324ba693e83d8fffd54950");
```

The next step requires to provide subscription' criterias. Single criteria is represented by Rule class, where one Subscription can have many Rules. Each Rule defines the key, the value and the matching criteria, f.e.: price<=80, orderId=ABC123, quality>=0.5, etc. Rules are always treated as conjunction, therefore only messages matching all of the criteria are delivered to the subscriber. To create subscription of alternative rules, convert it to few subscriptions of excluding rules.

In the following example, there are created Rules to match the Messages which defines the price lower than 80 and Quality greater or equal 0.5:

```
subscription.add(new Rule("price", "80", RuleType.LE));
subscription.add(new Rule("quality", "0.5", RuleType.GE));
```

Once the SELIS Publish/Subscribe system processes the Message which matches the Subscription' criterias, it notifies the Subscriber on the Callback provided during the registration of Subscription. Therefore the Subscriber is obligated to create the Callback method, on which it receives and handles the Message delivered by the SELIS Publish/Subscribe system.

The SELIS Publish/Subscribe Java Library provides the interface Callaback which requires to implement single onMessage() method. The message delivered from the SELIS Publish/Subscribe system is pushed to this method. In the following example, the Subscriber's callback simply prints the message content to the standard output:

```
public class MyCustomCallBack implements de.tu_dresden.selis.
pubsub.Callback {
    @Override
    public void onMessage(Message message) {
        System.out.println("Received {}", message);
    }
}
```

Once the Subscriber has defined the Subscription and the Callback, it can register them in the SELIS Publish/Subscribe system by invoking the subscribe() method:

```
pubSub.subscribe(subscription, new MyCustomCallBack());
```

The unchecked PubSubConnectionException can be thrown when the connection to the SELIS Publish/Subscribe system could not be established. Please see Error-handling section of this document for more information about error handling.

Error-handling

The SELIS Publish/Subscribe Java Library defines its own Exception hierarchy, where the highest Exception is PubSubException. All of the others exceptions inherits from PubSubException.

PubSubException is an unchecked exception, therefore it does not require from the Client to catch the exceptions which could be potentially thrown during the publish/subscribe operations. Nevertheless, for production application it is recommended to handle such exceptions.

Logging

The SELIS Publish/Subscribe Java Library uses org.slf4j.Logger API to provide some Debuging informations. The Library itself does not come with any implementation of org.slf4j.Logger, so it is up to the Client which implementation to choose. Please follow slf4j documentation for more information: https://www.slf4j.org/manual.html

For the prototyping we suggest to use the slf4j-simple, which does not require any initial configuration. Download from: https://www.slf4j.org/download.html

Maven dependency:

```
<dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.7.5</version>
```

```
</dependency>
```