**INSTALLATION GUIDE**

1. Installation and configuration 3

1.1. Requirements 3

1.2. Installation SLA Tool 3

1. Download project 3

2. Creating the mysql database 3

3. Importing the code to Eclipse 4

4. Configuring 4

4. Running localy the SLA 4

5. Testing everything with cURL 5

APPENDIX A 5

Enforcement setup 5

Metrics retriever 5

Metrics validator 5

Periodic execution 5

## 1. Installation and configuration

### 1.1. Requirements

- Oracle JDK 1.6.x - You can download the correctly one for your platform at: [JDK 1.6.x](http://www.oracle.com/technetwork/java/javase/downloads/jdk6-downloads-1637591.html). After installed, you have to make sure JAVA\_HOME variable is set pointing to the folder Java was just installed.

- Database to install the database schema for the service. [MySQL](http://www.mysql.com/).

- Maven - You can download from this address: [Maven download](http://maven.apache.org/download.html) .

## 1.2. Installation SLA Tool

#### 1. Download project

Download the project using maven from

<http://atossla.atosresearch.eu/svn/atossla/>

$ mvn checkout

#### 2. Creating the mysql database

From mysql command tool, create a database with this name: slarepositorydb (with the user root)

       create database atossla;

And create the user:

       CREATE USER atossla@localhost IDENTIFIED BY 'password';

              GRANT ALL PRIVILEGES ON atossla.\* TO atossla@localhost WITH GRANT OPTION;

From command prompt create scratch database:

       $ mvn compile exec:java -f sla-repository/pom.xml

#### 3. Importing the code to Eclipse

The core of the ATOSSLA has been developed using the Eclipse Java IDE, although others Java editors could be used, here we only provide the instructions about how to import the code into Eclipse.

The first step is to tell Maven to create the necessary Eclipse project files executing this:

       $ mvn eclipse:eclipse

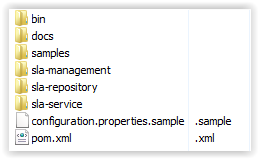
The previous command is going to generate the eclipse project files: .settings, .classpath, and .project. Again, please never upload those files to the svn, it is going to deconfigure the eclipse of other developers (it is easy to fix, just an annoying waste of time).

After it, you can from your eclipse import the project. Go to “import project from file” go to the trunk folder, and you should see the “ATOSSLA” project ready to be imported in your Eclipse.

#### 4. Configuring

The project is made up of three main modules and can be found under <http://atossla.atosresearch.eu/svn/atossla/trunk> SVN

* SLA Repository
* SLA Management
* SLA Service



A **configuration.properties.samp**le that is placed in the parent directory has to be copied to **configuration.properties**.

Several parameters can be configured through this file.

1. ***tomcat.directory*** when building, war will be automatically copied to this directory
2. ***db.\**** allows to configure the database username, password and name in case it has been changed from the proposed one in the section ‘2. Creating the mysql database’. It can be selected if queries from hibernate must be shown or not.
3. ***log.\**** allowsto configure the log files to be generated and the level of information.
4. ***manager.enformcement.\**** several parameters from the enforcement can be personalized
5. ***service.basicsecurity.\**** basic security is enabled. These parameters can be used to set the user name and password to access to the rest services.

#### 4. Running localy the SLA

In the parent directory there is also a pom.xml.To execute the SLA on Tomcat, just execute the following maven command on the path of sla-service module

$ mvn tomcat:run

or start-up the tomcat.

#### 5. Testing everything with cURL

The easiest way to play with the SLAATOS is to use [cURL](http://curl.haxx.se/).

Modifications are doing in the same way, but using -X PUT instead of -X POST.

### APPENDIX A

### Enforcement setup

The enforcement is a periodically executed procedure that checks if a SLA is being satisfied. In short, the enforcement check if a set of metrics fulfill the Service Level Objectives defined in the Guarantee Terms.

#### Metrics retriever

The metrics retriever is an object that implements eu.atos.sla.monitoring.IMetricsRetriever interface and is responsible to get the metrics to be evaluated in the enforcement. It is just an adaptor to query the monitoring system.

* agreementId: SLA agreement identifier
* serviceName.
* variable: name of the metric.
* begin, end: date of metrics to retrieve.
* maxResults: maximum number of values to retrieve.

The metrics retriever is configured in **configuration.properties.**

#### Metrics validator

A metrics validator is an object that evaluates the Service Level Objectives defined in an agreement, and it has to implement eu.atos.sla.evaluation.IMetricsValidator.

The metrics validator is configured in **configuration.properties**

#### Periodic execution

The execution of the enforcement is determined by a cron-styled rule defined in **configuration.properties** (spawnlookup variable).