

**A**daptive **A**utomation in **A**ssembly **For** **BLUE** collar workers satisfaction in **E**volvable context

**Web Porting Tool**

**Installation and Running**

# Executive summary

This document contains the guidelines to follow to install and set up the Web Porting tool.

Prerequisites

* Java 1.8
* Maven
* MySql 5.5
* SAR running
* Apache Tomcat 8+

**Create loa\_evaluation\_tool Schema**

Create a schema in Mysql having the name “loa\_evaluation\_tool” and execute the sql [script](https://github.com/Engineering-Research-and-Development/loa-evaluation-tool/tree/master/client) to allow the creation of all tables.

**Create a MySql user having the following configuration**

|  |  |
| --- | --- |
| Variable | Value |
| username | **loa\_user\_tool** |
| password | **loa\_user\_tool** |

**Create an environment variable for the CAMService URL.**

Set a new environment variable called **ENV\_SAR\_URL** containing the url of the CAMService

ENV\_SAR\_URL= *http://[host]:[port]/CAMService*

[Executive summary 2](#_Toc531961267)

[1 ONLY FOR DEVELOPERS 4](#_Toc531961268)

[1.1 CLONE PROJECT 4](#_Toc531961269)

[1.2 SERVER configuration 4](#_Toc531961270)

[1.3 Client configuration 5](#_Toc531961272)

[1.4 Run service 6](#_Toc531961277)

[2 ONLY FOR FINAL USERS 7](#_Toc531961278)

# ONLY FOR DEVELOPERS

## CLONE PROJECT

Open a terminal and run the following command:

*git clone* [*https://github.com/Engineering-Research-and-Development/loa-evaluation-tool.git*](https://github.com/Engineering-Research-and-Development/loa-evaluation-tool.git)

## SERVER configuration

**Setup configuration file**

Configure the file loa-evaluation-tool/server/src/main/resources/application.properties (if not yet present, create it) with the following values:

*server.port = <port where the application will be running> (e.g. 8080)*

*spring.datasource.url=<* *connection-url pointing to the loa\_evaluation\_tool schema> (*e.g. *jdbc:mysql://localhost:3306/loa\_evaluation\_tool?useSSL=false*

*spring.datasource.username= < valid username for* connecting to the db *> (e.g. loa\_user\_tool)*

*spring.datasource.password= <* *valid password for connecting to the db> (e.g. loa\_user\_tool)*

*spring.datasource.driver-class-name=<Driver of the selected db> (e.g. for mysql com.mysql.jdbc.Driver)*

*spring.jpa.hibernate.naming.physical-strategy= org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl*

*base.url=<url of the CAMService> (e.g. http://localhost:8080/CAMService)*

**Build project[[1]](#footnote-1)**

Move to the loa-evaluation-tool\server project directory and run the following command:

*mvn install*

a file LoAService.war will be created to the path “loa-evaluation-tool\server\target”.

**Server deployment**

To deploy the application under Apache Tomcat copy the LoAService.war file, created in the previous step, into the directory <apache tomcat>\webapps.

## Client configuration

**Prerequisites**

* Node.js (<https://nodejs.org/it/>)
* Angular 2+ (<https://angular.io/>)
* Module “angular-devkit/build-angular”; if not yet present open a terminal and run this command *npm install --save-dev @angular-devkit/build-angular*
* Module “angular/material”, if not yet present open a terminal and run this command *npm install --save @angular/material @angular/cdk*
* Grunt-war[[2]](#footnote-2). Open a terminal and run this command *npm install -g grunt-cli*
* Module Grunt-war. This is an useful module for creating war files (to install it, move to the client project folder and type: *npm install grunt-war --save-dev)*

**Build project[[3]](#footnote-3)**

Move to the directory loa-evaluation-tool\client project and run the following command:

*npm run build*

a directory LoAEvaluationTool will be created to the path “loa-evaluation-tool\client\dist”.

**Client deployment[[4]](#footnote-4)**

To deploy the application under Apache Tomcat copy the directory LoAEvaluationTool, created in the previous step, to the path <apache tomcat installation>\webapps.

**Build and deploy client project as WAR file**

If you prefer generate a war file also for the client project, move to the directory loa-evaluation-tool\client project and run the following command:

grunt “war”

a file LoAEvaluationTool.war will be created to the path “loa-evaluation-tool\client\dist”. To deploy it, copy the created LoAEvaluationTool.war file into the directory <apache tomcat>\webapps.

## Run service

**Start apache tomcat**

Run apache-tomcat-8.5.34\bin\startup.bat (or startup.sh)

**Open application**

Browse to the url

http://<host>:<port>/LoAEvaluationTool

**Shutdown apache tomcat**

Run apache-tomcat-8.5.34\bin\shutdown.bat (or shutdown.sh)

# ONLY FOR FINAL USERS

Download files LoAService.war and LoAEvaluationTool.war from the folder [loa-evaluation-tool](https://github.com/Engineering-Research-and-Development/loa-evaluation-tool)/build/[version].

**Application deployment**

Copy the downloaded files (LoAService.war and LoAEvaluationTool.war) to the path <apache tomcat installation>\webapps

**Start apache tomcat**

Run apache-tomcat-8.5.34\bin\startup.bat (or startup.sh)

**Open application**

Browse to the url

http://localhost:8080/LoAEvaluationTool

**Shutdown apache tomcat**

Run apache-tomcat-8.5.34\bin\shutdown.bat (or shutdown.sh)

1. Alternately to build, deploy and run the server project in one shot, move to the server folder and type: ***mvn spring-boot:run*** [↑](#footnote-ref-1)
2. Only if you want to generate a WAR file. [↑](#footnote-ref-2)
3. Alternately to build, deploy and run the client project in one shot under node.js, move to the server folder and type: ***ng serve***(in this case the client URL is http://localhost:4200). [↑](#footnote-ref-3)
4. Skip this step if you want to deploy the client project with a WAR file. [↑](#footnote-ref-4)