

Table of Contents

Introduction	0
Preconditions	1
Installation	2
Configuration	3
Mapping key extractor	3.1
Simulator scenarios	3.2
Web Service support	4
Configuration	4.1
JMS support	5
Configuration	5.1
REST support	6
Configuration	6.1
Request mapping	6.2
Samples	7
simulator-sample-rest	7.1
simulator-sample-ws	7.2
simulator-sample-jms	7.3
Further Reading	8

Citrus Simulator

Welcome to the Citrus simulator

This is a standalone simulator application for different message transports such as Http REST APIs, SOAP WebService interface and JMS messaging.

Clients are able to access the simulator endpoints and the simulator answers with predefined response messages. The simulator response logic is very powerful and enables us to simulate any kind of server interface.

The simulator uses the test framework Citrus www.citrusframework.org

NOTE: *This project is still under construction!*

Contributions and feedback is highly appreciated!

Preconditions

The Citrus simulator requires some software installed on your localhost.

Java 8

The simulator is a Java application coded in Java 8. Following from that you need at least Java 8 to run it as a Spring Boot web application. Please make sure that you have Java development kit installed and set up. You can verify this with this command in a new terminal window.

```
> java -version
```

Build tools

The simulator uses Maven as build tool. If you only want to run a distribution artifact of the simulator you are fine with just Java on your machine. In case you want to build and maintain your simulator instance you need Maven to build your simulator application. We used Maven 3 when coding the simulator. You can verify the Maven installation on your host with this command:

```
> mvn -version
```

Browsers

The simulator provides a small web user interface when started. You can access this web UI with your browser. As we are in an early state in this project we do not invest much time in full cross-browser compatibility. We use Chrome and Firefox during development. So the simulator application is most likely to be 100% working on these two browsers. Of course other browsers might work without any limitations, too.

Installation

The Citrus simulator is started as a web application that uses Spring Boot. The easiest way to get started is to use a Maven archetype that creates a new project for you.

```
> mvn archetype:generate -DarchetypeGroupId=com.consol.citrus -DarchetypeArtifactId=citrus-si
```

If you execute the command above the Maven archetype generator will ask you some questions about versions and project names. Once you have completed the generation you get a new Maven project that is ready to use. The project is created in a new folder on your machine. Switch to that folder and continue to build the project.

Build project

You can directly build the new project with

```
> mvn install
```

This compiles, tests and packages the project. Now we can run the simulator:

```
> mvn spring-boot:run
```

You will see the application starting up. Usually you will see some console log output. The web server should start within seconds. Once the application is up and running you can open your browser and point to <http://localhost:8080>.

Now everything is set up and you can start to create some simulator scenarios.

Use simulator artifacts

The simulator is also available as packages web application artifact. First of all download the latest distribution which is a Java JAR file located at labs.consol.de/maven/repository:

Save the Java archive to a folder on your local machine and start the Spring boot web application. The downloaded JAR should be executable from command line like this:

```
java -jar citrus-simulator-0.6.jar
```

`citrus-simulator`

You will see the application starting up. Usually you will see some console log output. The web server should start within seconds. Once the application is up and running you can open your browser and point to <http://localhost:8080>.

That's it you are ready to use the Citrus simulator.

Further reading

- The [Citrus reference manual](#) gives you a detailed description of all Citrus features.
- Examples:
 - [Sample projects](#)
- [Contributing](#) explains how you can contribute to this project. Pull requests are highly appreciated!