Tutorial on Eclipse Leshan v2

Internet of Things (2IMN15) 2016-2017, University of Technology Eindhoven
By Leila F. Rahman (l.f.rahman@tue.nl)

Eclipse Leshan is an open source LWM2M programming framework in Java. This tutorial is about how to install Leshan and to develop LWM2M client and LWM2M server using Eclipse Leshan for the Internet of Things (21MN15) practical. This tutorial uses Leshan version 0.1.11-M15 which can be downloaded or cloned from https://github.com/eclipse/leshan.

In this tutorial, we provide some guides for the following:

1. Leshan Installation on Linux
2. Leshan Installation and Development on Windows using Eclipse IDE for Java
3. Run time screenshots of `leshan-server-demo`

1 LESHAN INSTALLATION ON LINUX

Install and run Leshan version 0.1.11-M15 by following the steps on:

https://github.com/eclipse/leshan

Apache Maven 3.2.5 and JDK 7 have been installed on the Raspberry Pi and their directory have been set to the `PATH` environment variable during any shell launch (.bashrc) and therefore can be called from any location, as shown in Figure 1 and Figure 2. Figure 9 shows example of leshan-demo-client execution with options.

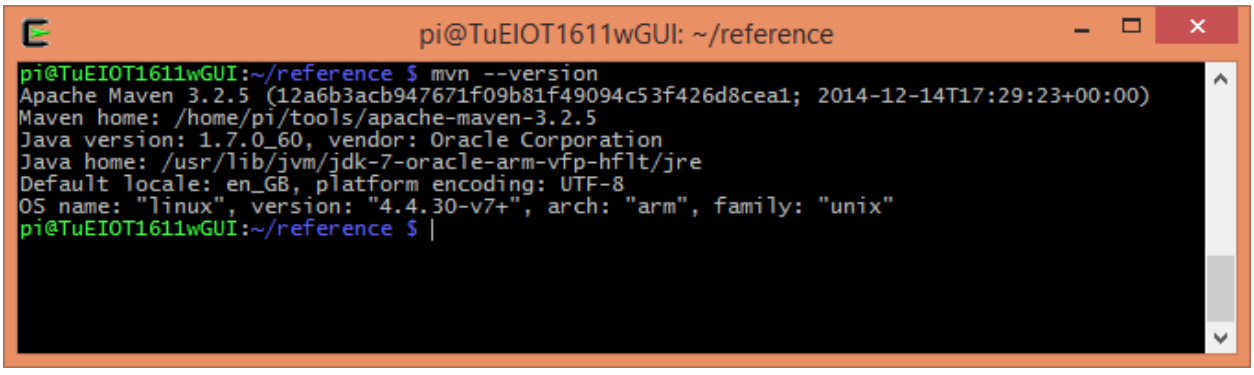

Figure 1. Version of Maven and Java Development Kit

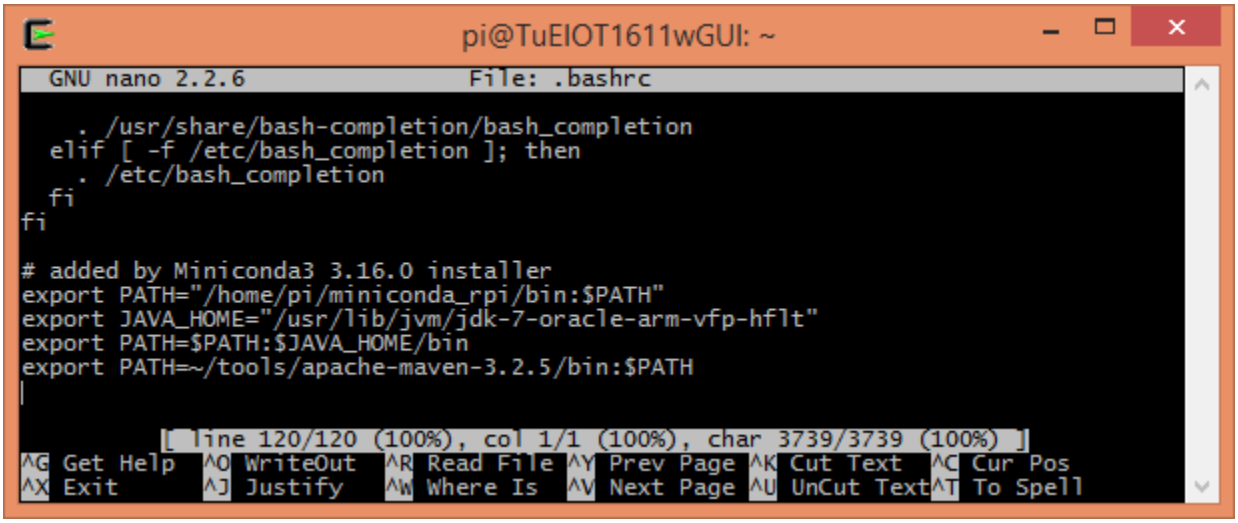

Figure 2. Setting directory of Maven and JDK to the PATH environment variable in .bashrc

2 LESHAN INSTALLATION AND DEVELOPMENT ON WINDOWS USING ECLIPSE IDE FOR JAVA

2.1 INSTALL LESHAN ON ECLIPSE IDE

1. Download and install Java SE 7 JDK (which include JRE) from http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html. If Java SE 7 JDK installer is not available, Java SE 8 JDK will work as well. It will result in more warnings, but they can be ignored.
2. Download and install Eclipse IDE for Java EE developer from http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/mars2
3. Download the zip distribution of `eclipse/leshan` version 0.1.11-M15 from https://github.com/eclipse/leshan
4. Extract the zip file into the Eclipse Workspace directory
5. Open Eclipse and import the Leshan project to Eclipse by accessing `File -> Import-> Maven -> Existing Maven Projects` and specify the directory of the project in the Eclipse Workspace folder
6. If using jre7, add jre7 to the `Installed JREs` menu in `Window -> Preferences -> Java -> Installed JREs`, and set jre7 as default execution environment
7. Under the `Installed JREs` menu, which is the `Execution Environment` menu, set the compatible JRE for JavaSE-1.7 to jre7.

2.2 DEVELOPING USER APPLICATION USING LESHAN-SERVER-DEMO

- Open `leshan-server-demo` project: a LWM2M demo server with a web UI.
- Fix the POM File by adding `<phase>package</phase>` below line 122
- Update the project by right clicking on the project folder and click `Maven-> Update Project…`
- Figure 3 shows the architecture of the `leshan-server-demo` project
- Modify client side files (JavaScript, HTML and CSS files) at `src/main/resources/webapp` folder according to your application requirements. You can use Angular JS front-end framework (http://campus.codeschool.com/courses/shaping-up-with-angular-js/intro) as used in the `leshan-server-demo` project, or you can use any other front-end framework for developing your user application front end.
- The JavaScript codes call services provided by the servlets (`package org.eclipse.leshan.server.demo.servlet`):
  - **`EventServlet`**: listens to registration and observation events
  - **`ClientServlet`**: gets all the clients connected to the server. Sends READ, WRITE, DELETE, EXECUTE, OBSERVE operations to LWM2M clients.
  - **`ObjectSpecServlet`**: gets pre-defined object specifications
  - **`SecurityServlet`**: manage DTLS security
- The following list HTTP API examples that are provided by the Java servlets:
  - `GET http://server_address/api/clients`
    - This API returns a JSON file describing all the clients registered in the Leshan Server's Client Registry as shown in Figure 14.
  - `GET http://server_address/api/clients/client_id/path_to_resource`
    - This API returns a JSON file describing the value of a resource in a LWM2M client as shown in Figure 15.
  - `GET http://server_address/api/objectspecs`
    - This API returns a JSON file describing all the object models recognized by the Leshan Server as shown in Figure 16.
- Look at the code of the servlets to find more HTTP APIs. You can develop new servlets which provide new services and their APIs for developing your user application back end and initialize the new servlets in the `LeshanServerDemo.java` class under the `// Create Servlet` comment.
- Run the project as Java Application, choose `LeshanServerDemo` as the main type. When asked for the goals on Maven configuration, input `eclipse:eclipse` and proceed with running the project.

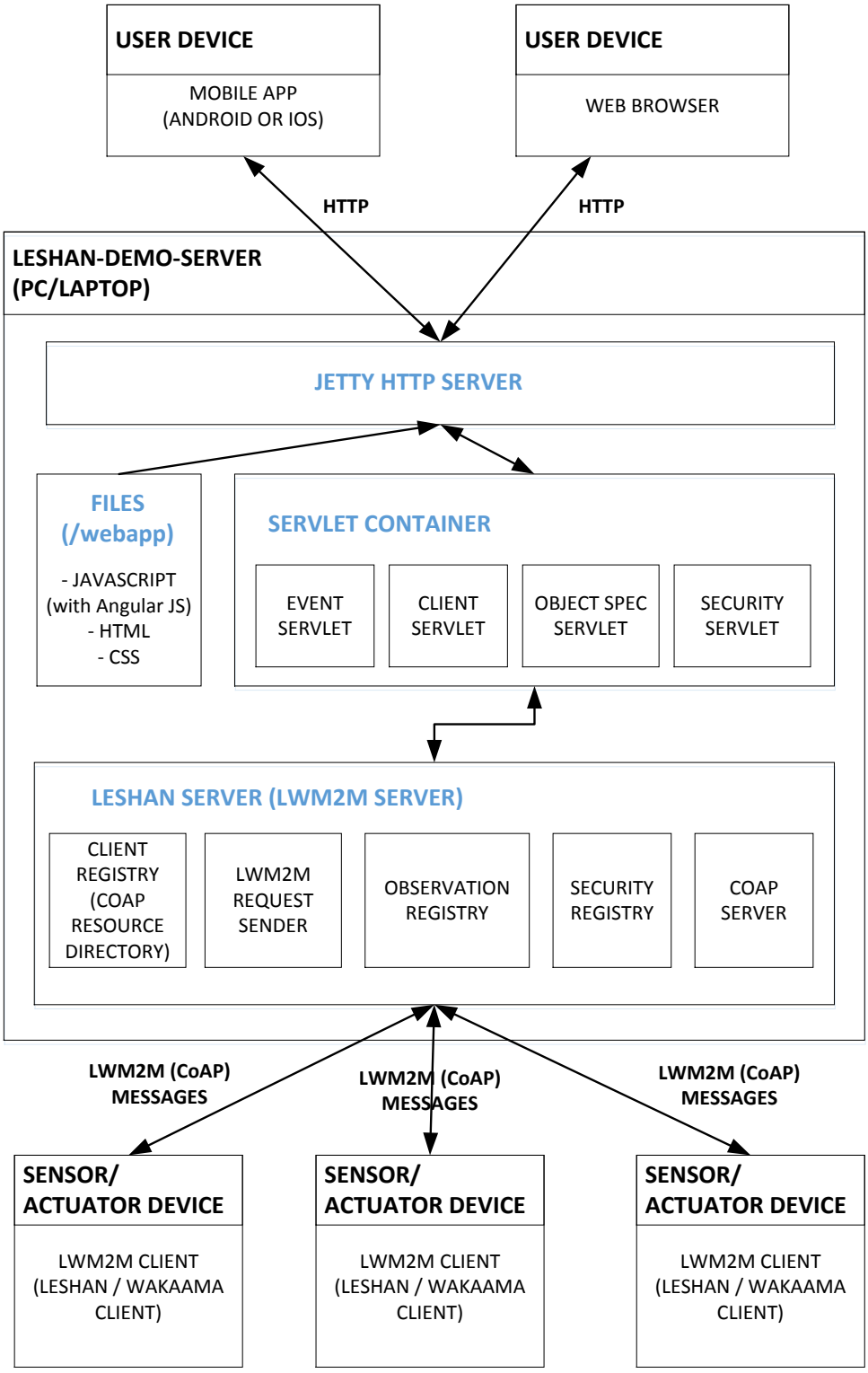

Figure 3. Architecture of `leshan-server-demo` Java project

2.3 DEVELOPING LWM2M CLIENT USING LESHAN-CLIENT-DEMO

- Open `leshan-client-demo` project: a sample of LWM2M client
- Define your object specifications (from the IPSO smart objects expansion pack, or self-defined objects) in an `objectspec.json` file. The standard objects (LWM2M objects and IPSO smart objects starter pack) are already defined in `/leshan-core/src/main/resources/oma-objects-spec.json`
- Create a new system environment variable "`MODELS_FOLDER`" that reference the folder with the `objectspec.json` file
- In the `package org.eclipse.leshan.client.demo`, create classes for your LWM2M objects extending the `BaseInstanceEnabler` class. In the `leshan-client-demo` project, three object classes are created: `MyDevice.java`, `MyLocation.java` and `RandomTemperatureSensor.java`. You can use one of these classes as a template for your new objects.
- In each object class, specify the actions for READ, WRITE, EXECUTE, DELETE requests on each resources of the object.
- Initialize the objects (under the `// Initialize object list` comment in the code)
- Run the project as Java Application. You can also pass on arguments to the leshan-client-demo execution by setting the arguments tab in run configuration. Right click on the `leshan-client-demo project -> run as -> run configurations …` (see Figure 4)
- The list of options can be found in the `LeshanClientDemo.java` source code or on Figure 8.

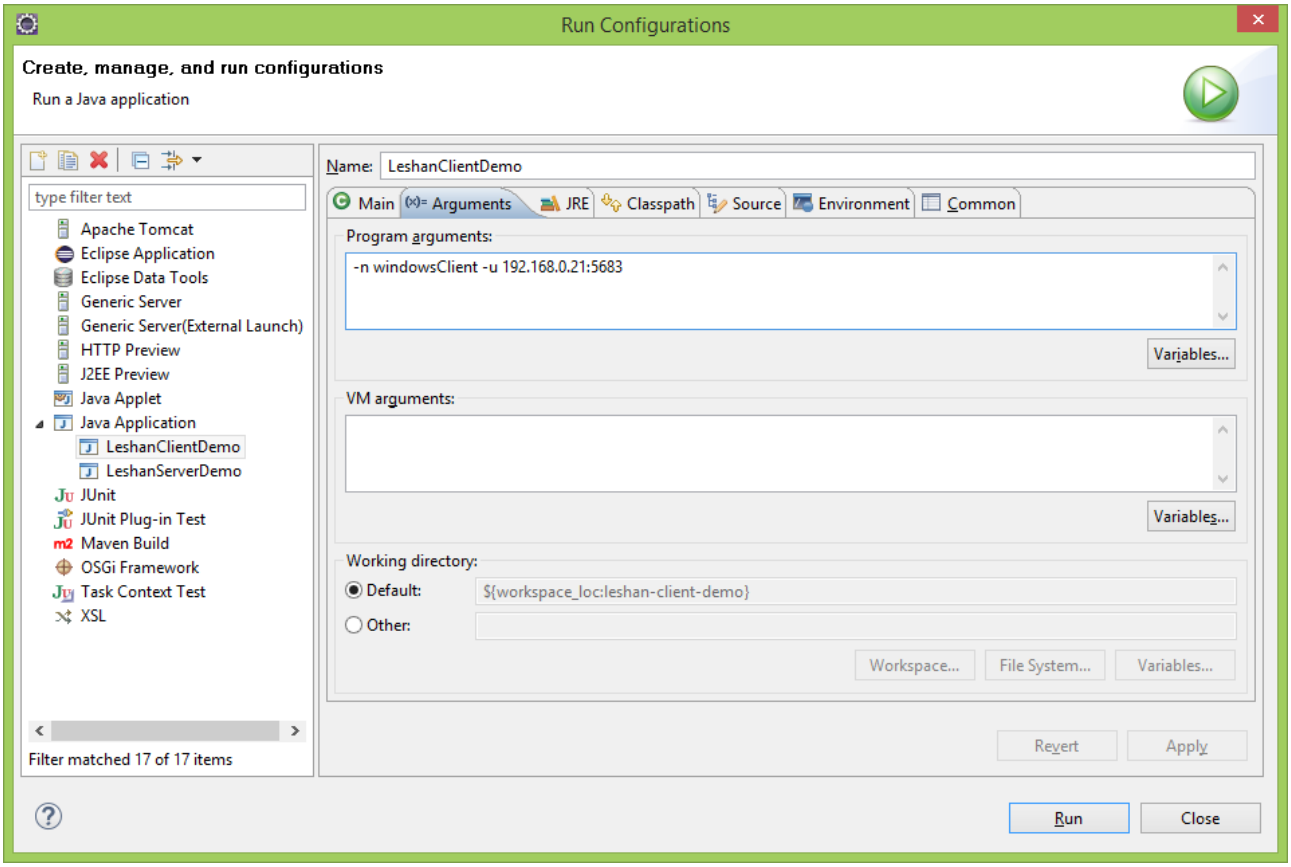

Figure 4. Setting arguments for leshan-client-demo project

2.4 DEPLOYING LESHAN-CLIENT-DEMO ON RASPBERRY PI

- To export the project into executable .jar file (to be deployed on Raspberry Pi for example), right click on the project and click `Export -> Java -> Runnable Jar File`
- Choose the first library handling option : Extract required libraries into generated file (see Figure 5)

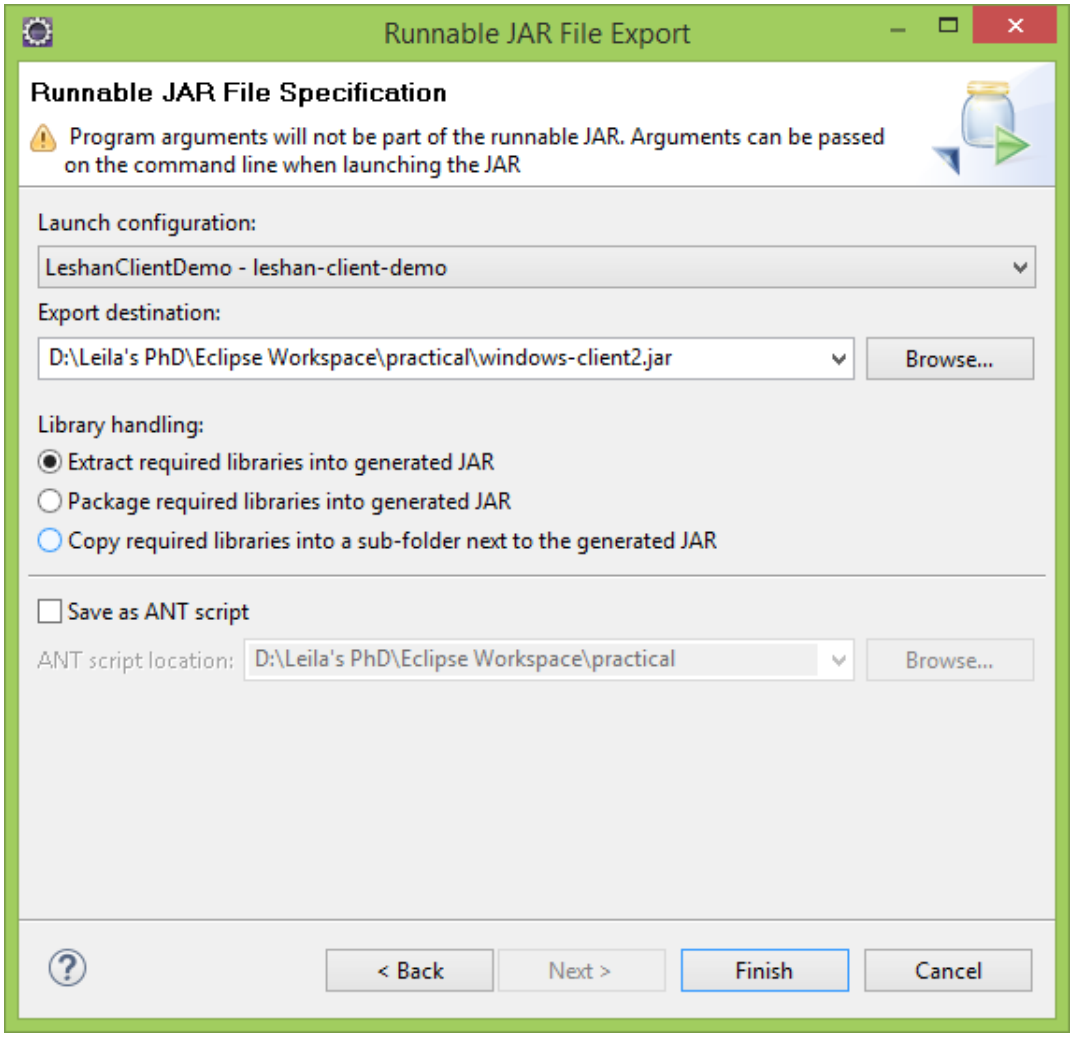

Figure 5. Runnable Jar Library Handling Option

- Once you have your .jar file, extract the jar file into a directory.
- Move the content of `/resources` directory, which consists of: `oma-objects-spec.json` and `simplelogger.properties` to the root directory. The structure of the jar file should look similar like `leshan-client-demo-*-SNAPSHOT-jar-with-dependencies.jar` as shown in Figure 6 .
- Repackage the jar file. Figure 7 shows the repackaged jar file. We have now files `oma-objects-spec.json` and `simplelogger.properties` in the root directory.

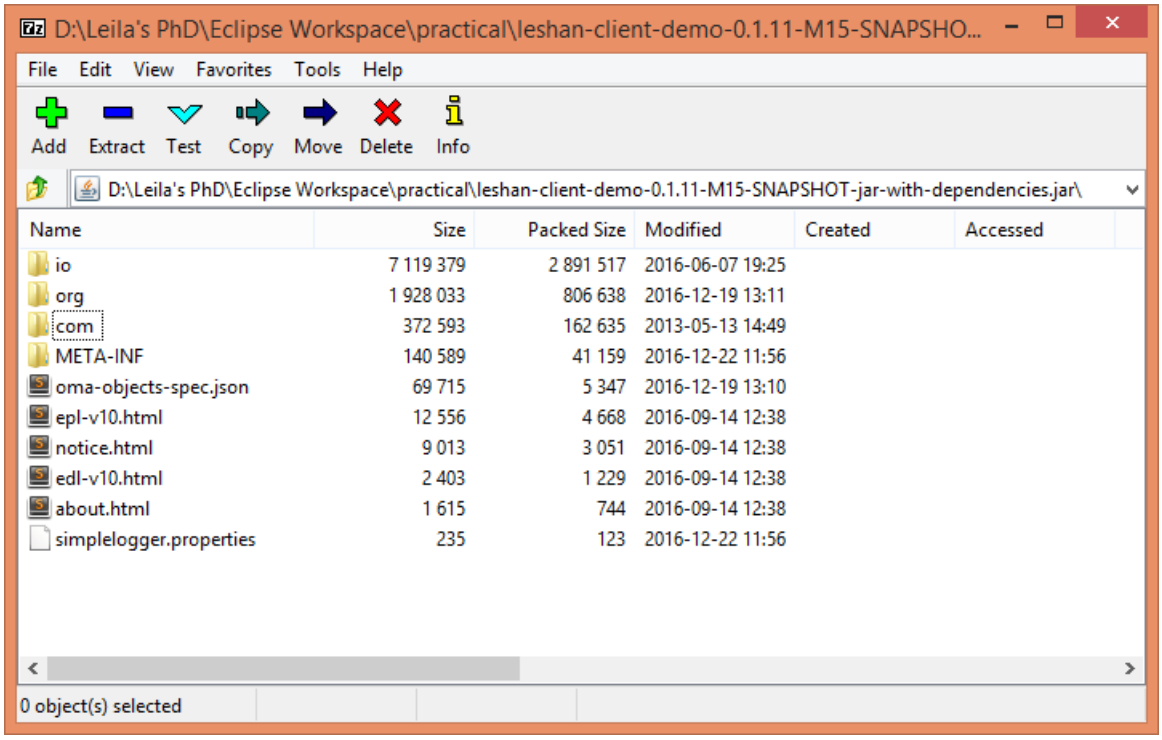

Figure 6. Structure of file leshan-client-demo-*-SNAPSHOT-jar-with-dependencies.jar

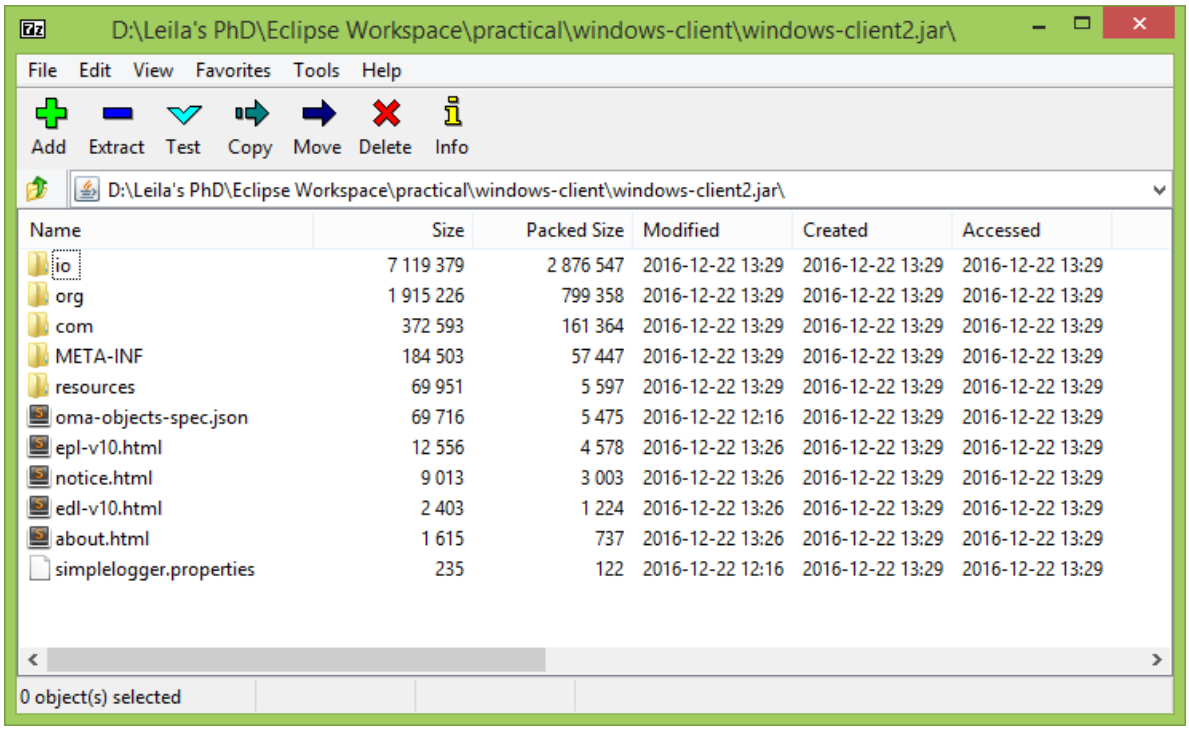

Figure 7. Structure of the repackaged windows-client2.jar

- To get the options of the leshan-demo-client execution, you can type: `java –jar jarfilename.jar –help` as shown in Figure 8.

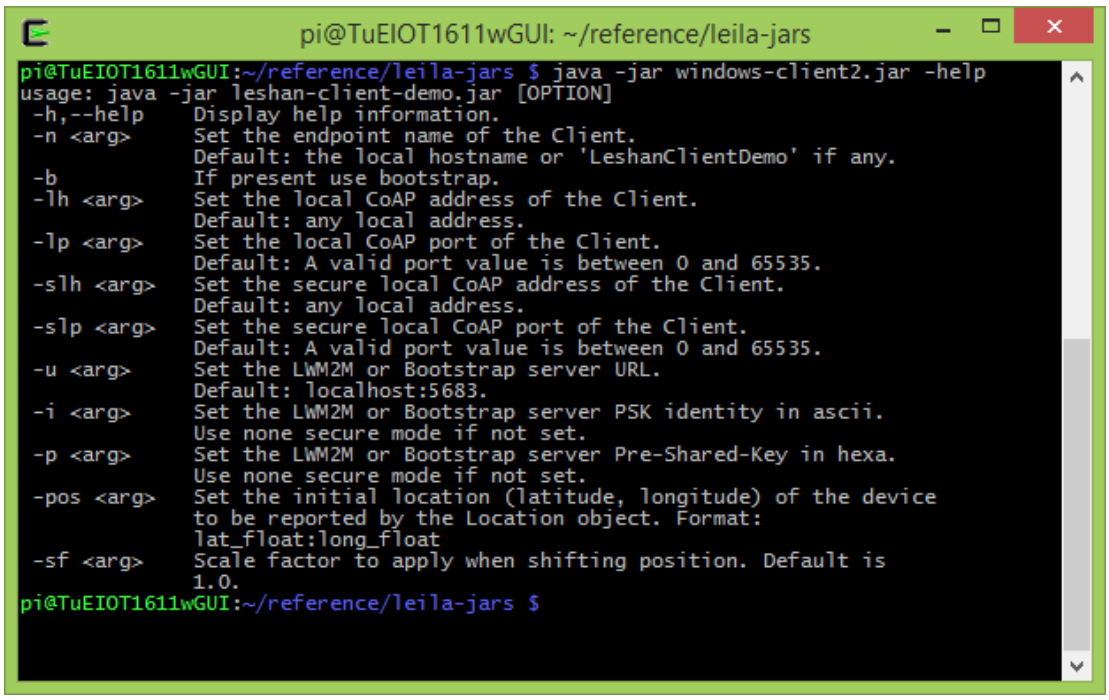

Figure 8. Execution options for leshan-client-demo

- Figure 9 shows example of leshan-demo-client execution with options

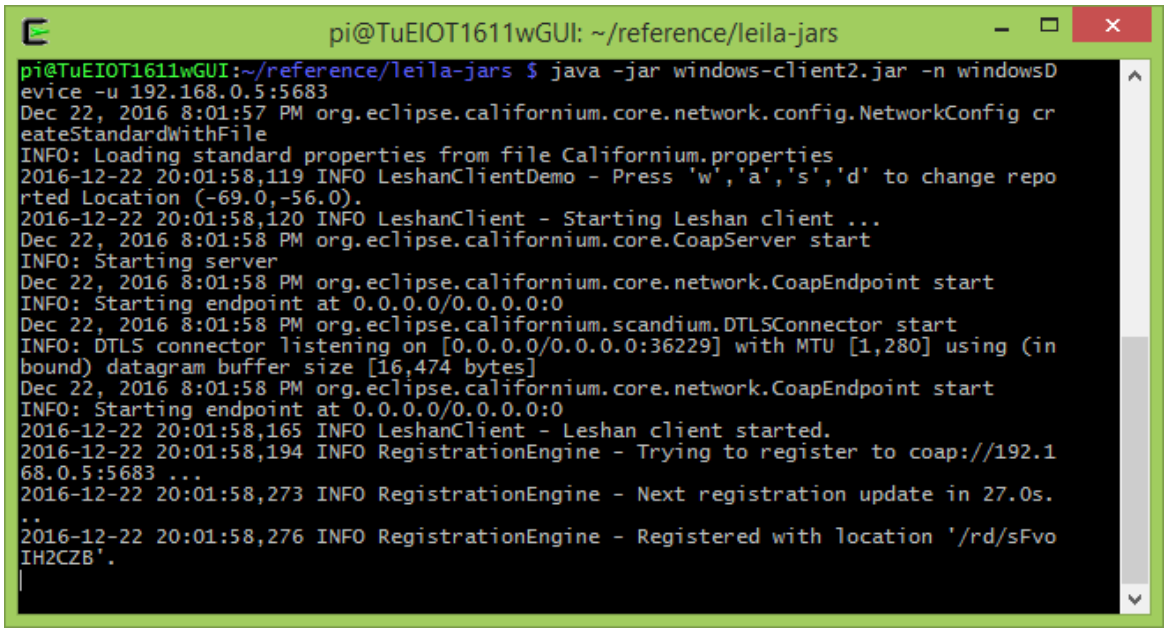

Figure 9. Examples of using the option in leshan-client-demo execution

3 RUN TIME SCREENSHOTS

In this section we will show you run time screen shots of `leshan-server-demo` (a demo server with web-UI) and a LWM2M client, in this case the `leshan-client-demo` of Leshan version 0.1.11-M15.

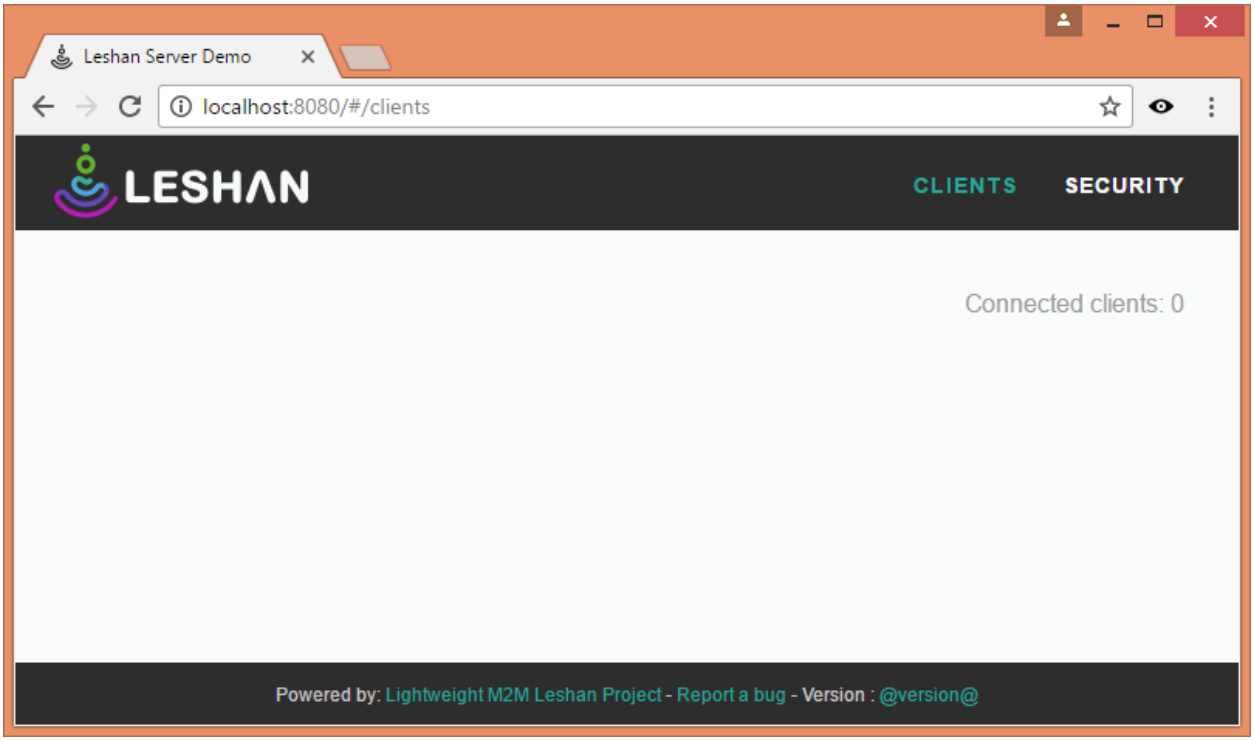

Figure 10. LeshanServerDemo Web Server is running on http://localhost:8080 on laptop with IP 192.168.0.5

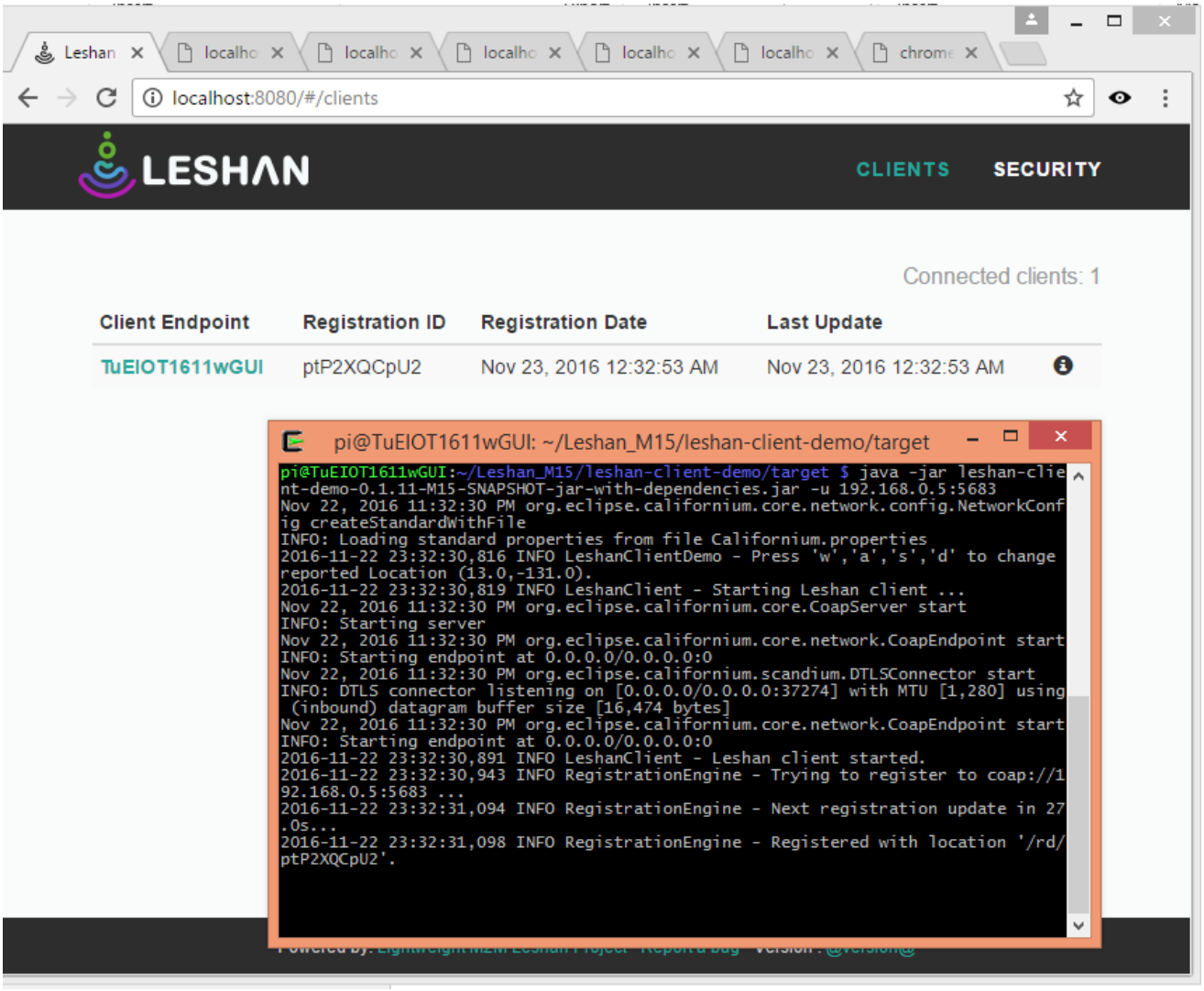

Figure 11. Client TuEIOT1611wGUI (LeshanClientDemo on Raspberry Pi) is registered on LeshanServerDemo

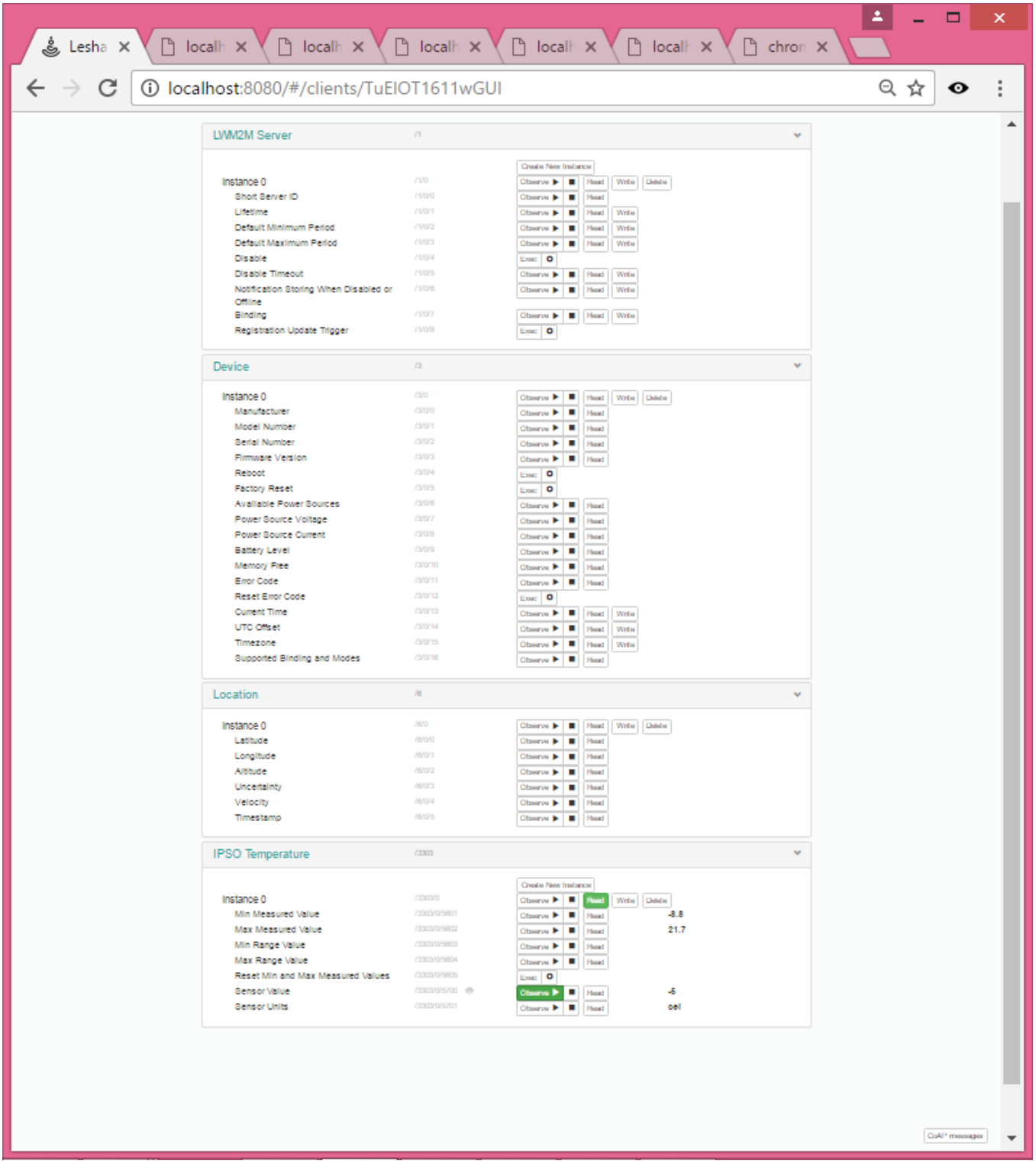

Figure 12. Object and resources of TuEIOT1611wGUI

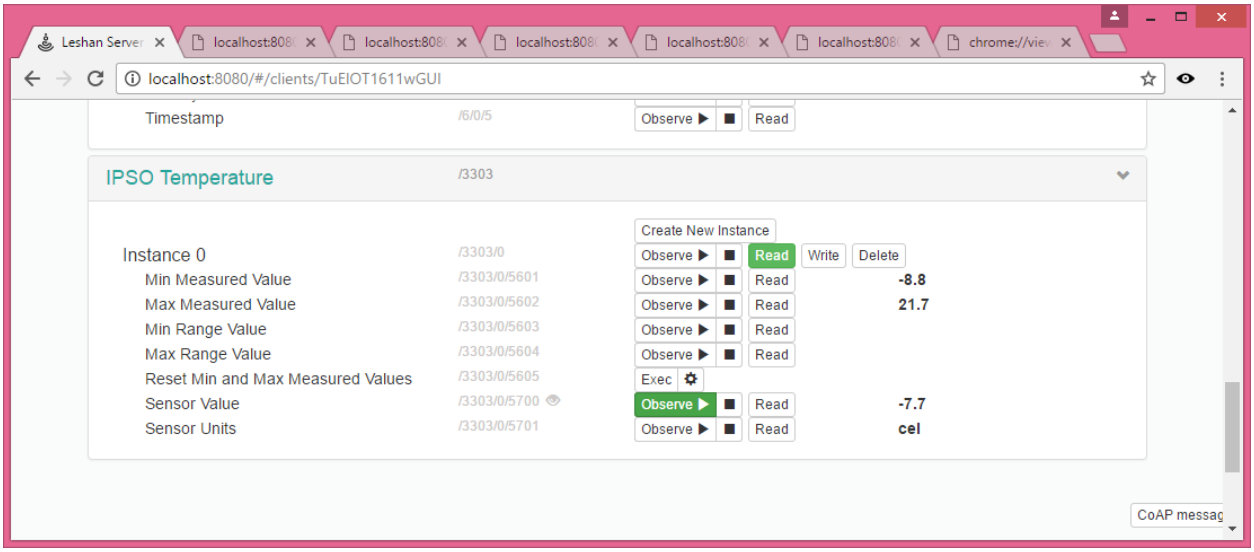

Figure 13. Object IPSO Temperature and its resources

Figure 14. Output of the `/api/clients` HTTP request

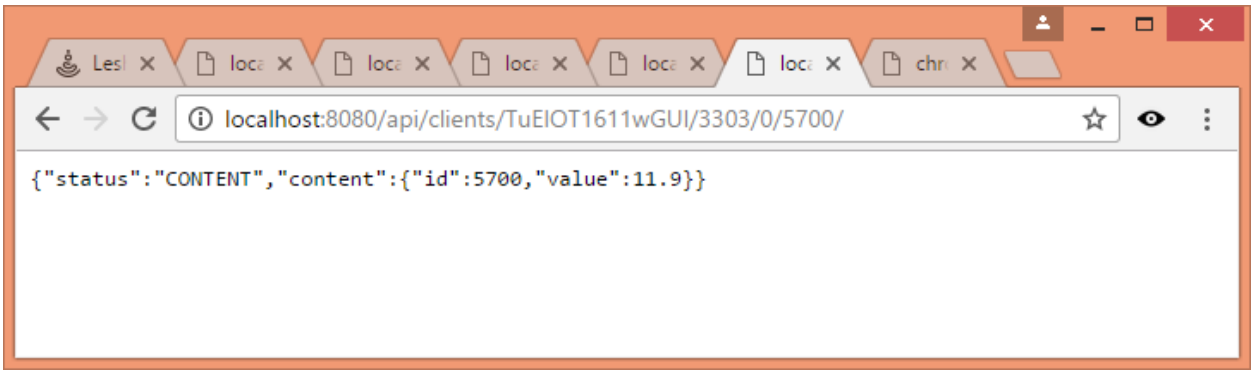

Figure 15. Output of the `/api/clients/TuEIOT1611wGUI/3303/0/5700/` HTTP request

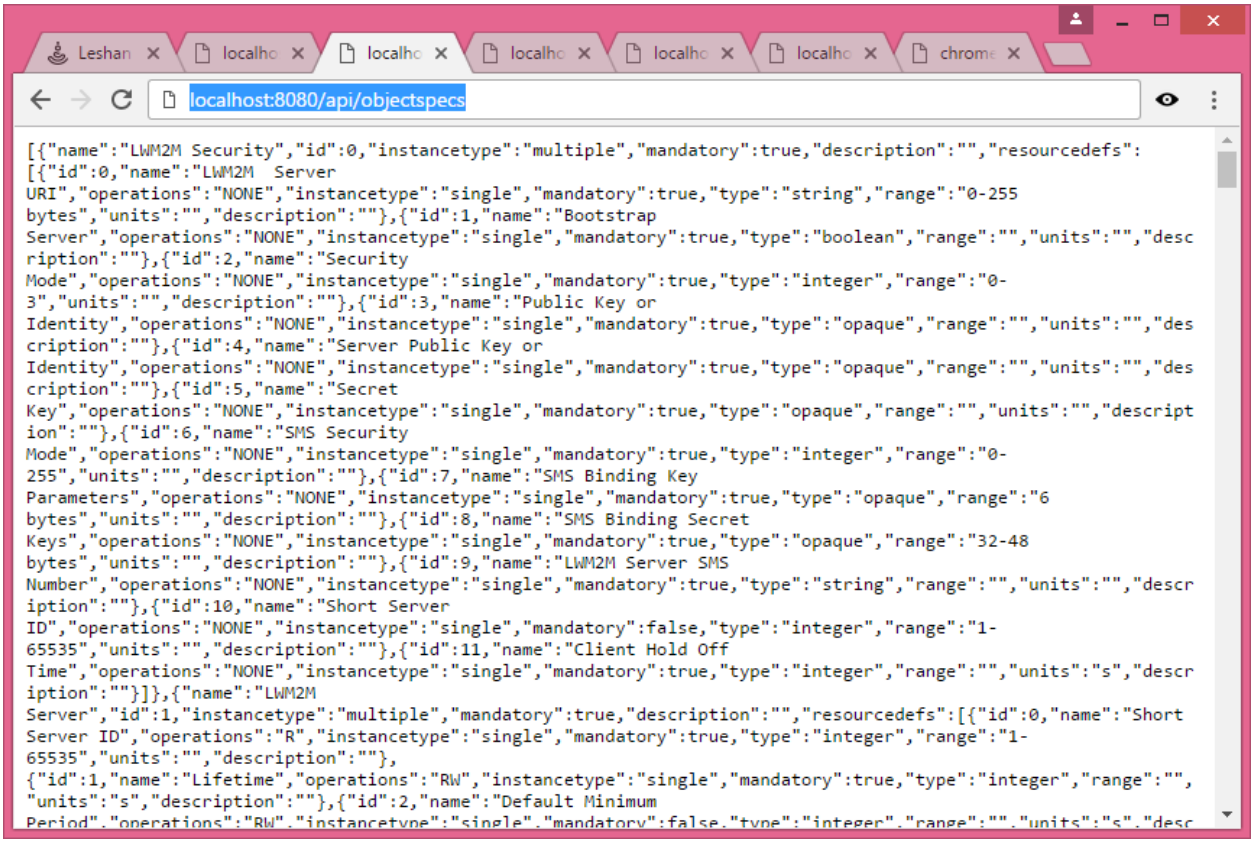

Figure 16. Output of the `/api/objectspecs` HTTP request