Tutorial on Eclipse Leshan v2

Internet of Things (2IMN15) 2016-2017, University of Technology Eindhoven By Leila F. Rahman (I.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

- Download and install Java SE 7 JDK (which include JRE) from <u>http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html</u>. 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 (<u>http://campus.codeschool.com/courses/shaping-up-with-angular-js/intro</u>) as used in the leshanserver-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:
 - o 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.
 - - This API returns a JSON file describing the value of a resource in a LWM2M client as shown in Figure 15.
 - o 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/omaobjects-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-clientemo 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.

٢		Run Configurations
Create, manage, and run configu Run a Java application	rations	
Image: Second Secon	Name: LeshanCli G Main (№= Arg Program argum -n windowsCli VM arguments: Working director ● Default: ○ Other:	ientDemo puments JRE % Classpath % Source Environment Common nents: ent -u 192.168.0.21:5683 Variables Variables Variables S{workspace_loc:leshan-client-demo} Korkspace File System Variables
< >> Filter matched 17 of 17 items		Re <u>v</u> ert Appl <u>v</u>
?		<u>R</u> un Close

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)

0	Runnable JAR File Export	– 🗆 🗙							
Runnable JAR Fi	ile Specification	8							
Program arguments will not be part of the runnable JAR. Arguments can be passed on the command line when launching the JAR									
Launch configuration	n:								
LeshanClientDemo -	- leshan-client-demo	~							
Export destination:									
D:\Leila's PhD\Eclip	se Workspace\practical\windows-client2.jar 🗸 🗸	Browse							
Library handling: Extract required lib Package required Copy required libr	Library handling: Extract required libraries into generated JAR Package required libraries into generated JAR Copy required libraries into a sub-folder next to the generated JAR								
Save as ANT scrip	t								
ANT script location:	D:\Leila's PhD\Eclipse Workspace\practical <	Browse							
?	< Back Next > Finish	Cancel							

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-objectsspec.json and simplelogger.properties in the root directory.

D:\Leila's PhD\Eclipse	Workspace\practi	cal\leshan-clier	nt-demo-0.1.11	-M15-SNA	psho – 🗖	x
File Edit View Favorites	Tools Help					
Add Extract Test Copy	→ X 1 Move Delete Info					
🏂 📓 D:\Leila's PhD\Eclipse	e Workspace\practical\	leshan-client-dem	o-0.1.11-M15-SNAF	SHOT-jar-wit	th-dependencies.jar\	~
Name	Size	Packed Size	Modified	Created	Accessed	
🌗 io	7 119 379	2 891 517	2016-06-07 19:25			
퉲 org	1 928 033	806 638	2016-12-19 13:11			
] com	372 593	162 635	2013-05-13 14:49			
📗 META-INF	140 589	41 159	2016-12-22 11:56			
📓 oma-objects-spec.json	69 715	5 347	2016-12-19 13:10			
📓 epl-v10.html	12 556	4 668	2016-09-14 12:38			
Notice.html	9 013	3 051	2016-09-14 12:38			
📓 edl-v10.html	2 403	1 229	2016-09-14 12:38			
📓 about.html	1 615	744	2016-09-14 12:38			
simplelogger.properties	235	123	2016-12-22 11:56			
<						>
0 object(s) selected						

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

D:\Leila's PhD\E	clipse Workspace\p	practical\windo	ws-client\wind	ows-client2.jar\	. – 🗆	x
File Edit View Favorites	Tools Help					
Add Extract Test Conv	→ X i					
Add Extract Test Copy D:\Leila's PhD\Eclips	e Workspace\practical\v	windows-client\wi	ndows-client2.jar\			~
Name	Size	Packed Size	Modified	Created	Accessed	
io 🛛	7 119 379	2 876 547	2016-12-22 13:29	2016-12-22 13:29	2016-12-22 13:29	
🔰 org	1 915 226	799 358	2016-12-22 13:29	2016-12-22 13:29	2016-12-22 13:29	
🐌 com	372 593	161 364	2016-12-22 13:29	2016-12-22 13:29	2016-12-22 13:29	
🐌 META-INF	184 503	57 447	2016-12-22 13:29	2016-12-22 13:29	2016-12-22 13:29	
resources	69 951	5 597	2016-12-22 13:29	2016-12-22 13:29	2016-12-22 13:29	
📓 oma-objects-spec.json	69 716	5 475	2016-12-22 12:16	2016-12-22 13:29	2016-12-22 13:29	
📓 epl-v10.html	12 556	4 578	2016-12-22 13:26	2016-12-22 13:29	2016-12-22 13:29	
📓 notice.html	9 013	3 003	2016-12-22 13:26	2016-12-22 13:29	2016-12-22 13:29	
鼞 edl-v10.html	2 403	1 224	2016-12-22 13:26	2016-12-22 13:29	2016-12-22 13:29	
📓 about.html	1 615	737	2016-12-22 13:26	2016-12-22 13:29	2016-12-22 13:29	
simplelogger.properties	235	122	2016-12-22 12:16	2016-12-22 13:29	2016-12-22 13:29	
<						>
0 object(s) selected						

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.

E	pi@TuEIOT1611wGUI: ~/reference/leila-jars	-		×
pi@TuEIOT1611 usage: java -	LwGUI:~/reference/leila-jars \$ java -jar windows-client2.jar - -jar leshan-client-demo.jar [OPTION]	hel	p	^
-h,help	Display help information.			
-n kargs	Default: the local hostname or 'LeshanClientDemo' if any.			
-b	If present use bootstrap.			
- m karys	Default: any local address.			
-lp <arg></arg>	Set the local CoAP port of the Client.			
-slh <arg></arg>	Set the secure local CoAP address of the Client.			
	Default: any local address.			
-sip kang>	Default: A valid port value is between 0 and 65535.			
-u <arg></arg>	Set the LWM2M or Bootstrap server URL.			
-i <arg></arg>	Set the LWM2M or Bootstrap server PSK identity in ascii.			
	Use none secure mode if not set.			
-p <arg></arg>	Set the LWM2M or Bootstrap server Pre-Shared-Key in nexa. Use none secure mode if not set.			
-pos <arg></arg>	Set the initial location (latitude, longitude) of the device	2		
	lo be reported by the Location object. Format: lat_float:long_float			
-sf <arg></arg>	Scale factor to apply when shifting position. Default is			
pi@TuEIOT1611	LwGUI:~/reference/leila-jars \$			
				\sim

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.

💩 Leshan Server Demo 🛛 🗙 🔽		▲ _ □ ×
← → C (i) localhost:8080/#/clients		☆ • :
المعالمة المعامة المعامة المعام الم	CLIENTS	SECURITY
	Conne	cted clients: 0
Powered by: Lightweight M2M Leshan Project - Report a bug - Version : (@version@	

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

RAHMAN, L.F. (L.F.RAHMAN@TUE.NL) 12/22/16

TUTORIAL ON ECLIPSE LESHAN V2

ی Les	han 🗙 🎦 localho	× Cocalho ×) localho ×	ු localh × ි chrom ෙ	
$\leftarrow \rightarrow$	C i localhost:80	80/#/clients			☆ • :
\$	ÅLESH ف	N		CLIENTS	SECURITY
				Conne	cted clients: 1
	Client Endpoint	Registration ID	Registration Date	Last Update	
	TuEIOT1611wGUI	ptP2XQCpU2	Nov 23, 2016 12:32:53 AM	Nov 23, 2016 12:32:53	AM 🚯
		▶ pi@TuEIOT161 pi@TuEIOT1611wGUI:~ nt-demo-0.1.11-M15- Nov 22, 2016 11:32: ig createStandardWi INFO: Loading stand 2016-11-22 23:32:30 Nov 22, 2016 11:32: INFO: Starting serv Nov 22, 2016 11:32: INFO: Starting endp 2016-11-22 23:32:30 Nov 22, 2016 11:32: INFO: Starting endp 2016-11-22 23:32:30 92.168.0.5:5683 2016-11-22 23:32:31 .05 2016-11-22 23:32:31 .05 2016-11-22 23:32:31 .05 2016-11-22 23:32:31 .05	11wGUI: ~/Leshan_M15/leshan /Leshan_M15/leshan-client-de SNAPSHOT-jar-with-dependenci 30 PM org.eclipse.californiu thFile lard properties from file Cal 0,816 INFO LeshanClientDemo - 13.0,-131.0). 0,819 INFO LeshanClient - Sta 30 PM org.eclipse.californiu or 30 PM org.eclipse.californiu or listening on [0.0.0.0/0.0. 1 buffer size [16,474 bytes] 30 PM org.eclipse.californiu or listening on [0.0.0.0/0.0. 1 buffer size [16,474 bytes] 30 PM org.eclipse.californiu or listening on [0.0.0.0/0.0. 1 buffer size [16,474 bytes] 30 PM org.eclipse.californiu ont at 0.0.0/0.0.0.000 ,891 INFO LeshanClient - Les 0,943 INFO RegistrationEngine .098 INFO RegistrationEngine	-client-demo/target mo/target \$ java -jar 16 es.jar -u 192.168.0.5:56 m.core.network.config.Net ifornium.properties Press 'w','a','s','d' 1 mting Leshan client m.core.CoapServer start m.core.network.CoapEndpo m.scandium.DTLSConnector 0.0:37274] with MTU [1,2 m.core.network.CoapEndpo han client started. = Trying to register to = Next registration upo = Registered with locat	eshan-clie shan-clie sas etworkConf co change pint start start 280] using pint start co coap://1 date in 27 tion '/rd/

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

					÷ _		×
💩 Lesha 🗙 🗋 loc	alh 🗙 🎦 Iocalh 🗙 🌔	🖞 localh 🗙 🗸	🕒 localh 🗙 🎦 localh 🗙 🎦 d	thron ×			
$ullet$ $ ightarrow$ $egin{array}{c}$ $egin{array}{c} \ egin{array}{c} \ egin{$	host:8080/#/clients/TuEl0	OT1611wGUI			ର 🕁	0	:
	LVM2M Server	/1		¥			-
			Cruste New Instance				
	Instance 0		Otserve				
	Lifetime		Otserve				- 10
	Default Minimum Period	/1/0/2	Otserve 🕨 🔳 Read Write				
	Default Maximum Period	/1/0/3	Otserve 🕨 🔳 Peed Write				
	Disable	/1/0/4	Exec O				- 11
	Disable Timeout	/1/0/5	Otserve 🕨 🔳 Feed Write				- 10
	Notification Storing When Disabled or	/1/0/8	Observe Otserve Otserve Otserve				- 11
	Binding		Observe				- 11
	Registration Update Trigger	/1/0/8	Exar: O				- 11
	Device	10					
	Device	13		-			
	Instance 0	73/0	Otserve F S Peed Write Delete				
	Manufacturer Medel Number		Otserve F E Read				
	Sedal Number		Observe P Peed				
	Firmware Version		Otserve P B Read				
	Reboot	/3/0/4	Exac O				
	Factory Reset	/3/0/5	Exer O				- 11
	Available Power Sources	/3/0/8	Otserve 🕨 🔳 Read				- 11
	Power Source Voltage	/3/0/7	Otserve 🕨 🔳 Read				- 11
	Power Source Current	/3/0/8	Otserve 🕨 📕 Read				- 11
	Battery Level	/3/0/9	Otserve				- 11
	Memory Free		Otserve F E Read				
	Error Code Beset Error Code	/3/0/12	Observe P P Peed				
	Current Time		Charges B Basel Works				
	UTC Offset	/3/0/14	Otserve > Past Write				
	Timezone	/3/0/15	Otserve 🕨 🔳 Read Write				
	Supported Binding and Modes	/3/0/18	Ctosrve 🕨 🔳 Peed				
	Location	/8		~			
	Instance 0	/8/0	Citaserve 🕨 🔳 Fourd Write Debute				
	Latitude	/8/0/0	Otsserve 🕨 🔳 Read				
	Longitude	/8/0/1	Otzerve 🕨 🔳 Paed				
	Altitude	/8/0/2	Otserve F Read				
	Uncertainty	19/0/4	Observe				
	Timestamp	/8/0/5	Ctoserve > Food Food				
	IPSO Temperature	1303		~			
			Create New Instance				
	Instance U		Otserve				
	Max Measured Value		Observe Man Hand -5.8				
	Min Range Value	/3303/0/5803	Otherve P B Read				
	Max Range Value	/3303/0/5804	Otserve 🕨 🔳 Read				
	Reset Min and Max Measured Values	/3303/0/5805	Exac O				
	Sensor Value	/3303/0/5700 🔿	Observe > E Food -5				- 11
	Sensor Units	/3303/0/5701	Otserve 🕨 🔳 Pased Cel				
					0	AP message	-

Figure 12. Object and resources of TuEIOT1611wGUI

U localnost:0000/#/clients/TuElOT1611WG	UI		
Timestamp	/6/0/5	Observe Read	
PSO Temperature	/3303		*
		Create New Instance	
Instance 0	/3303/0	Observe 🕨 🔳 Read Write Delete	
Min Measured Value	/3303/0/5601	Observe Read -8.8	
Max Measured Value	/3303/0/5602	Observe 🕨 🔳 Read 21.7	
Min Range Value	/3303/0/5603	Observe 🕨 🔳 Read	
Max Range Value	/3303/0/5604	Observe 🕨 🔳 Read	
Reset Min and Max Measured Values	/3303/0/5605	Exec 🜣	
Sensor Value	/3303/0/5700 👁	Observe Read -7.7	
Sensor Units	/3303/0/5701	Observe 🕨 🔳 Read 🛛 🛛 Cel	
Min Range Value Max Range Value Reset Min and Max Measured Values Sensor Value Sensor Units	/3303/0/5603 /3303/0/5604 /3303/0/5605 /3303/0/5700 /3303/0/5701	Observe ► Read Observe ► Read Exec Φ Observe ► Read -7.7 Observe ► Read cel	

Figure 13. Object IPSO Temperature and its resources



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

ل Lesi 🗙		-		×		
←→ C	localhost:8080/api/clients/TuElOT1611wGUI/3303/0/5700/	☆	o	:		
{"status":"C	{"status":"CONTENT","content":{"id":5700,"value":11.9}}					

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



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