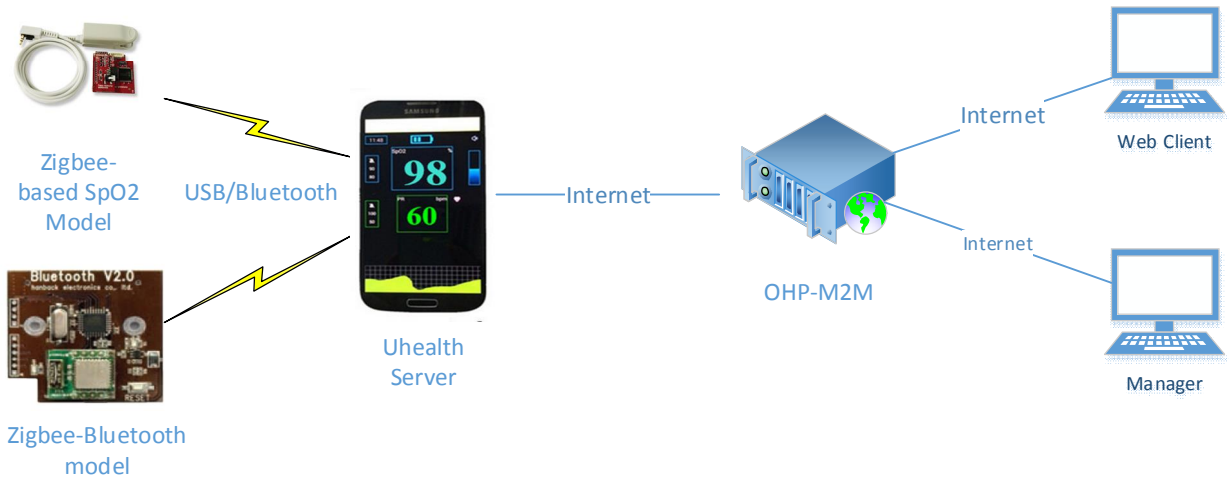
USER GUIDELINE.

System Architecture



The UHealthServer is a CoAP server that loads resources that is collected from the Medical device. OM2M is a platform that handles user requests (Managers and Clients) and relay them to the UHealthServer for response.

The user guideline is divided into two sections; The Installation procedure and How to use the OHP-M2M. They are explained explicitly as shown below;

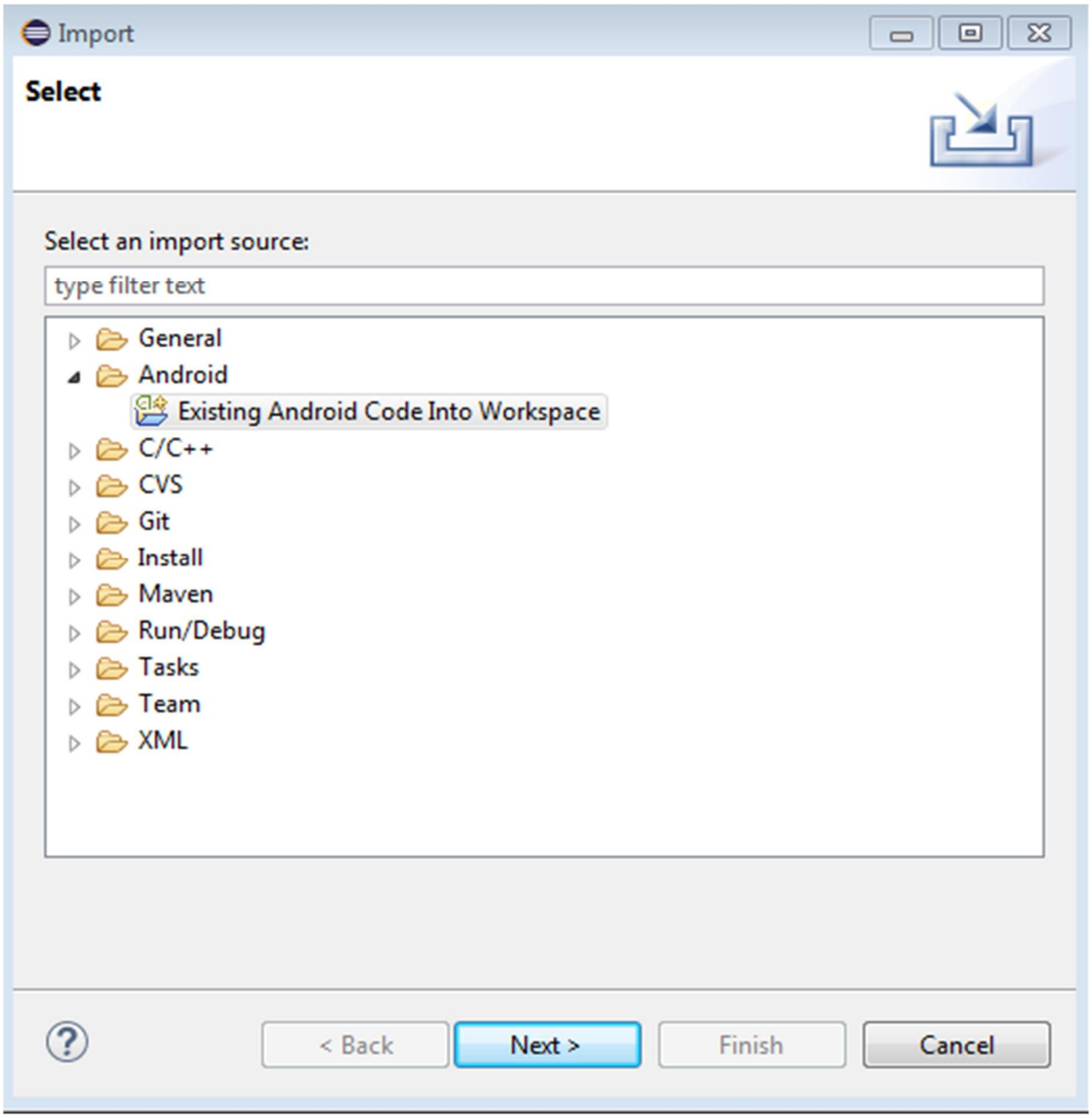
Installation Procedure

1. Follow the guidelines in the http://wiki.eclipse.org/OM2M/Clone to help you clone and build OM2M. You can use either Eclipse IDE or command line basing on what you prefer. Check and confirm that OM2M is installed and running well.
2. Download both the UHealthServer and the OHP-M2M from Github. The OHP-M2M download is an OSGI bundle and the UHealthServer is an Android Application.

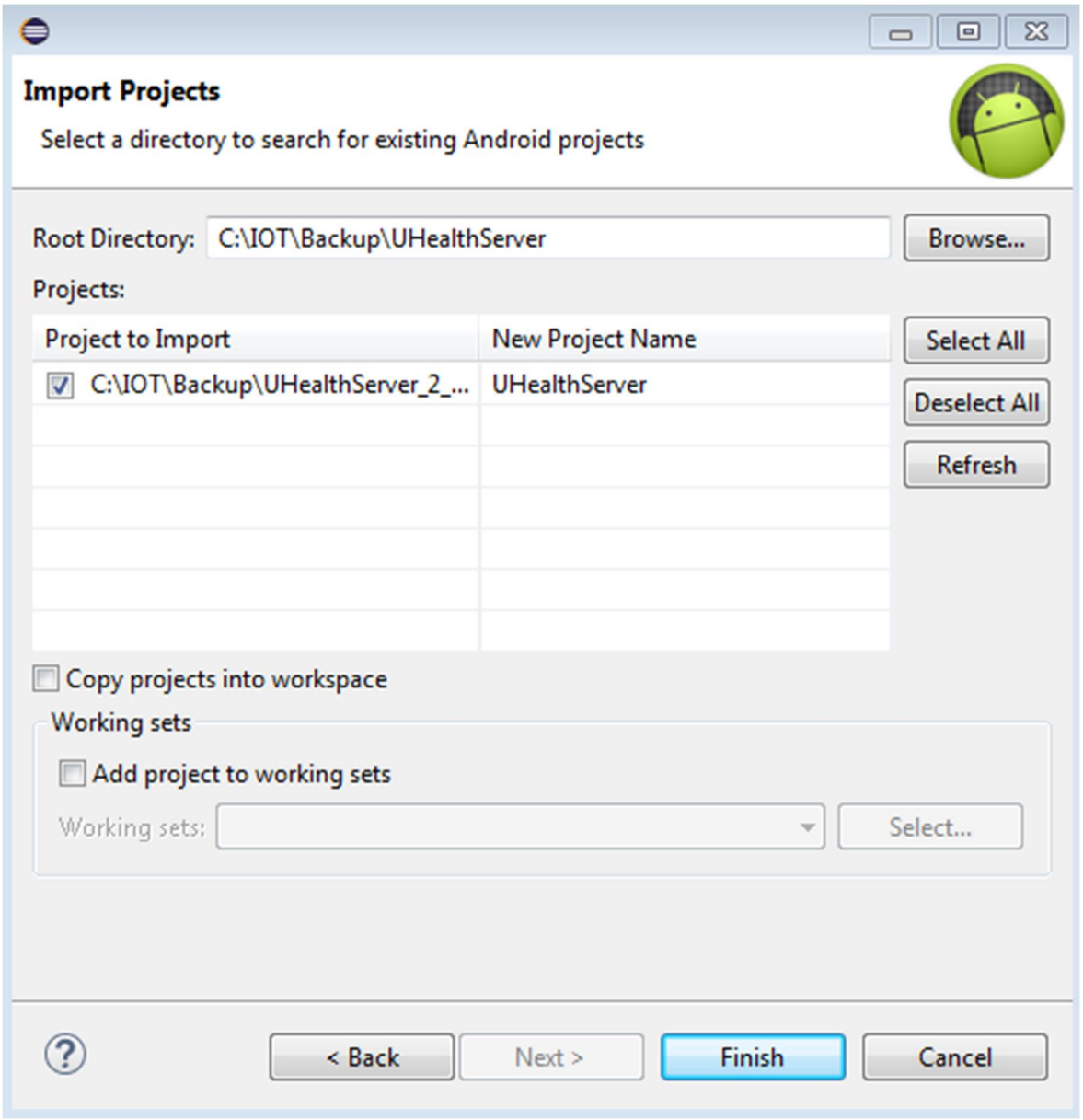
INSTALLING THE COAP UHEALTHSERVER.

3. Download the CoAP server from Github and follow the guidelines as follows.
4. Import the UHealthServer as an Android Project as shown below.

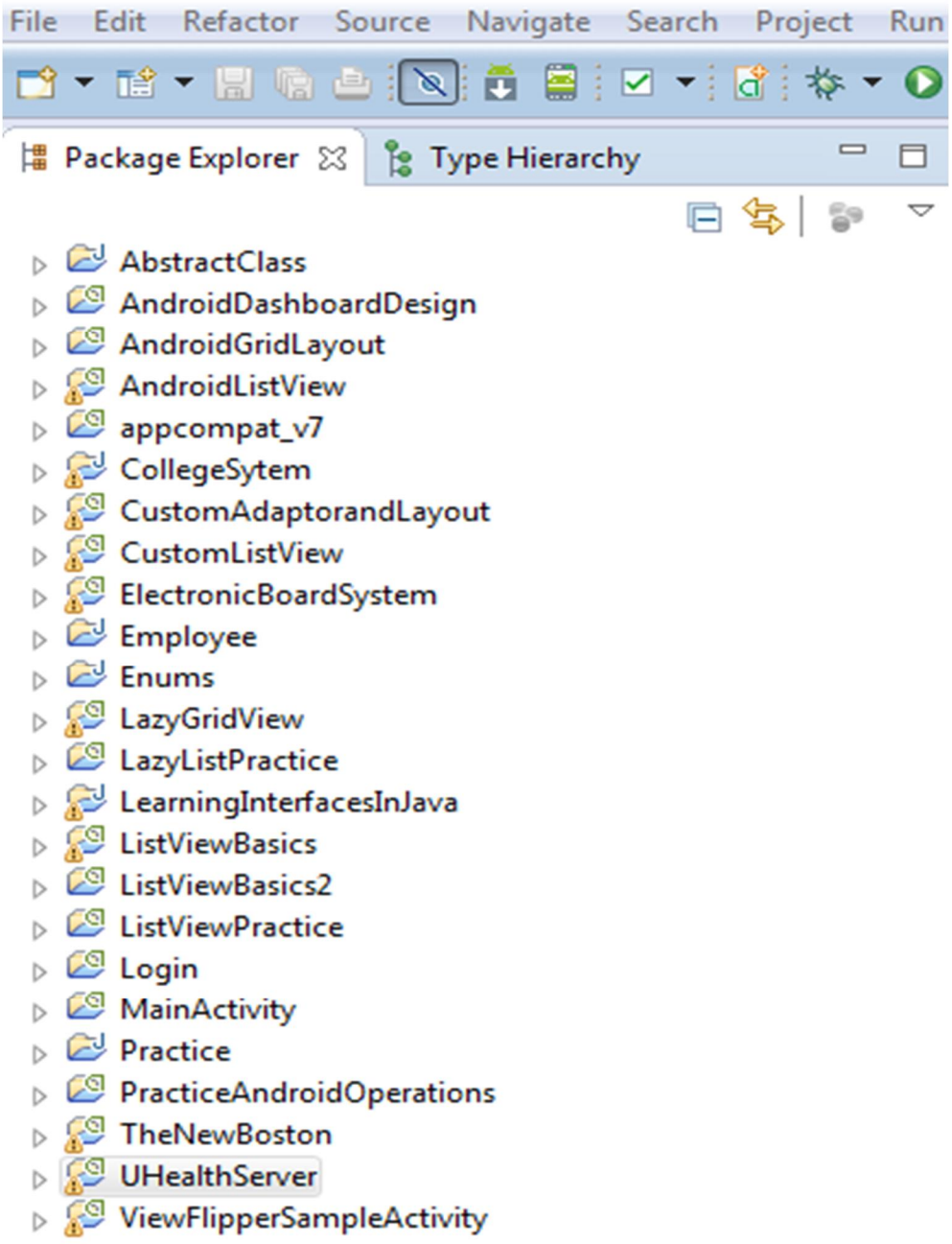
   a. Open Eclipse and click on File → Import Select Existing Android Code Into Workspace



b. Select the Root Directory where the UHealthserver was saved. And Click finish



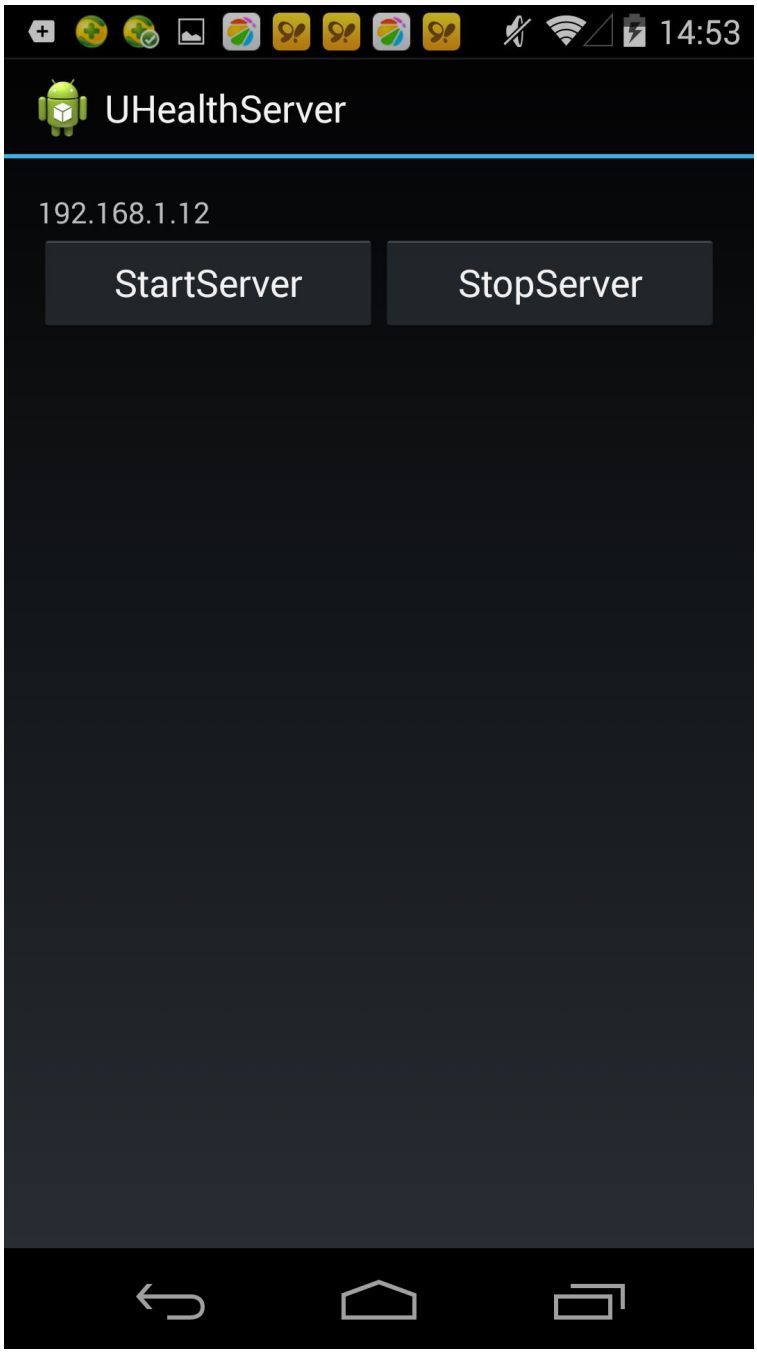
The UHealthServer will added in your work space and the files will be as shown below.



Connect your phone to act as the Emulator to run the UHealthServer.

Right Click on the UHealthserver ⟶ Run As ⟶ Android Application

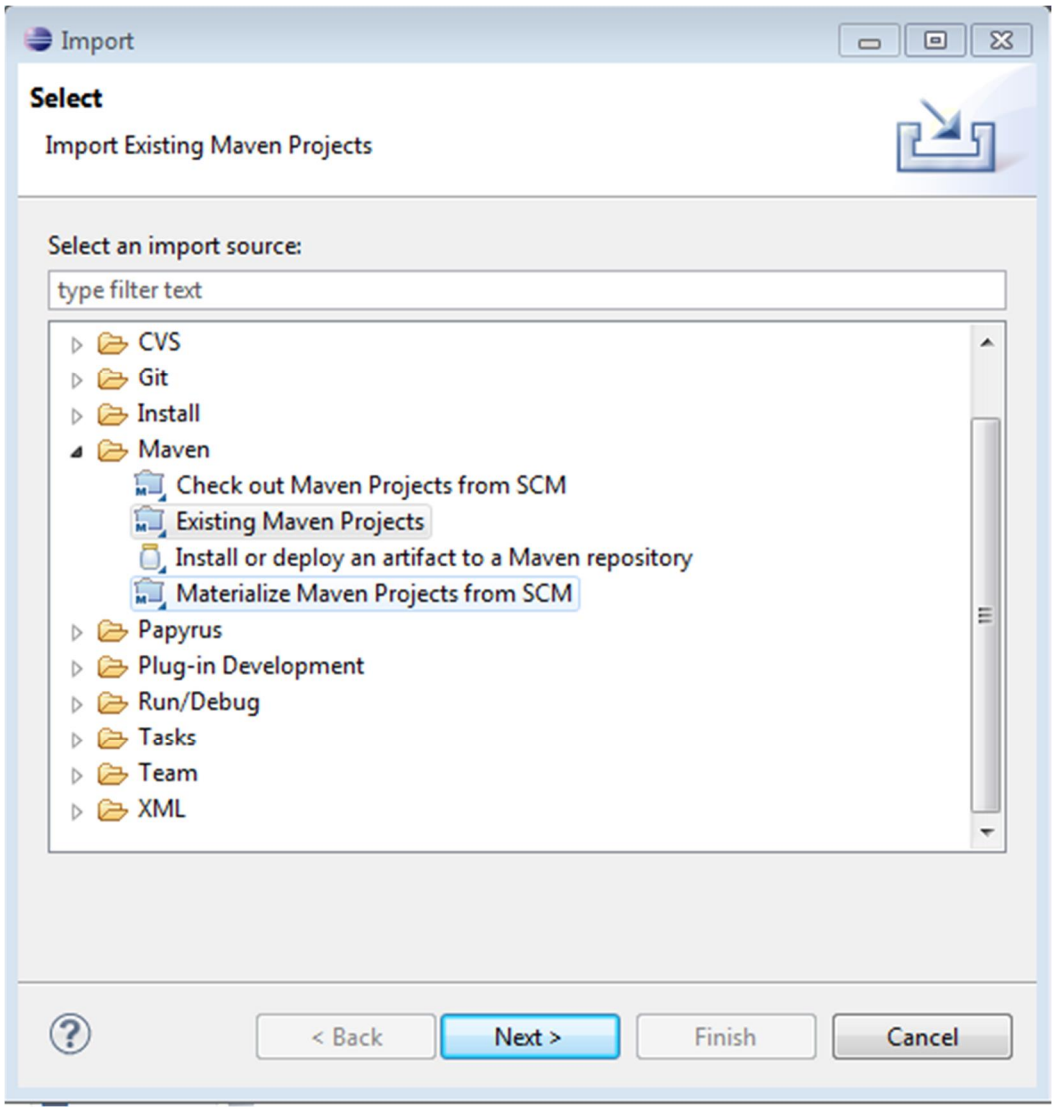
There are two buttons to start the server and Stop the server, StartServer and StopServer respectively.



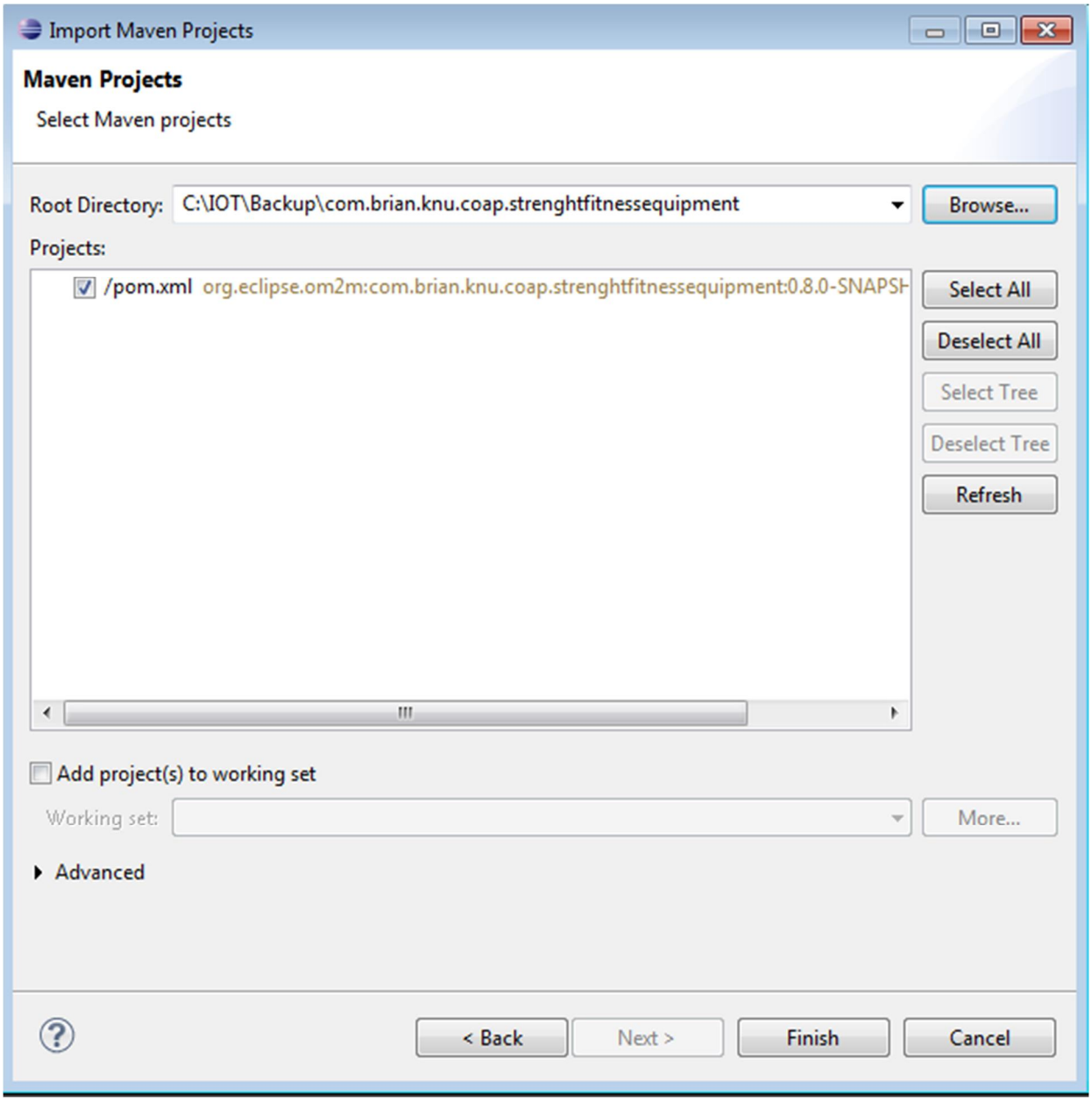
The UHealthserver runs and it shows the IP of the server and the buttons to start and stop the server as shown above in the screen shot.

INSTALLING THE OHP-M2M

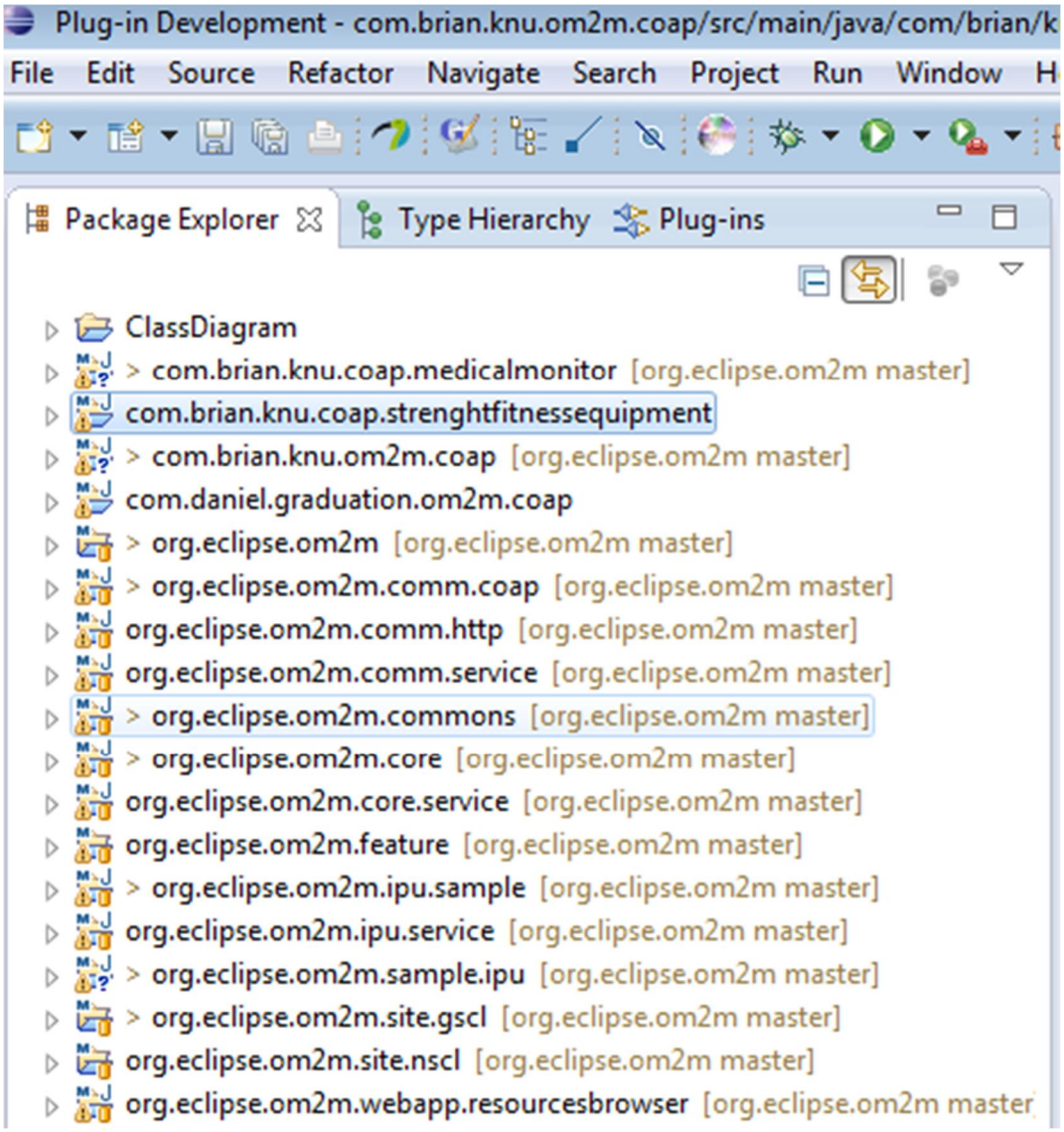
1. Import the downloaded OHP-M2M as a Maven Project. Import Existing Maven Projects



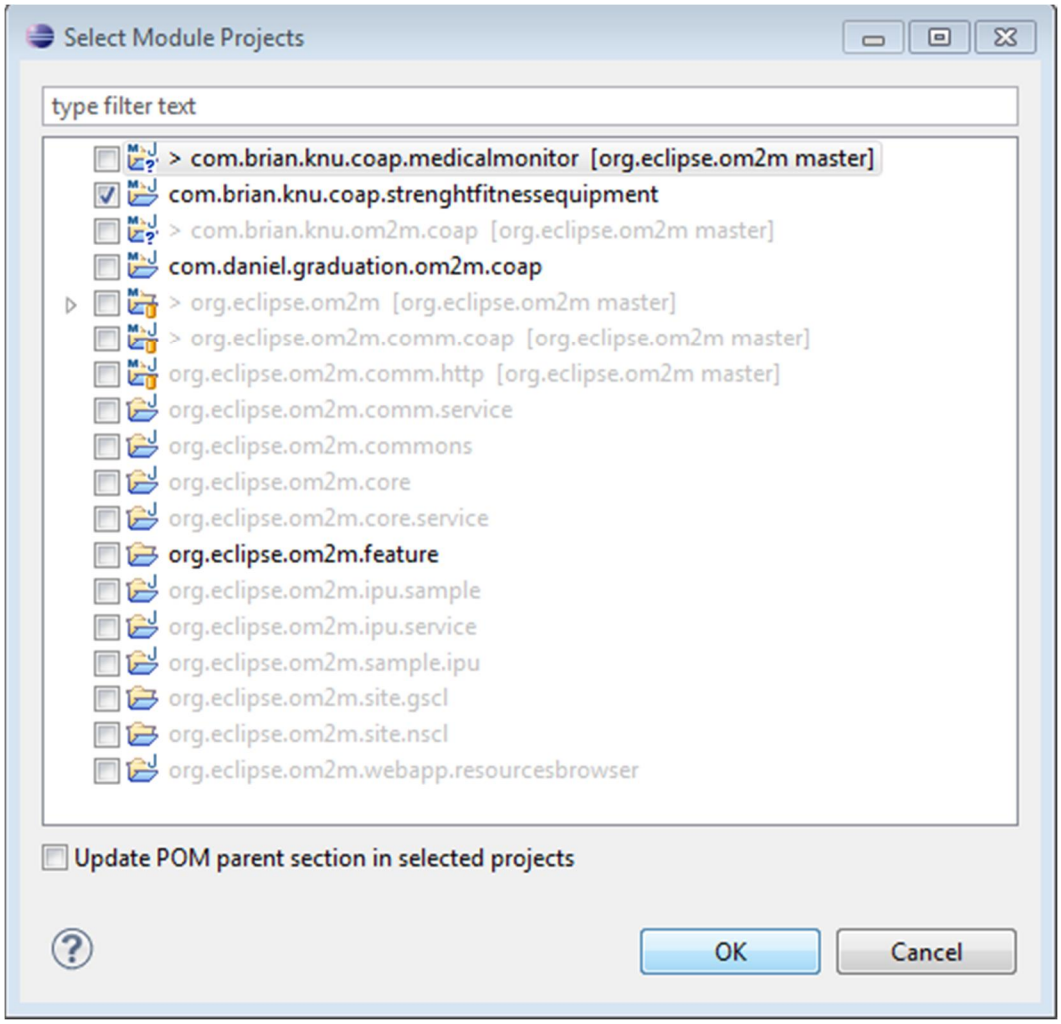
2. Select the location where the OHP-M2M was downloaded and click finish.



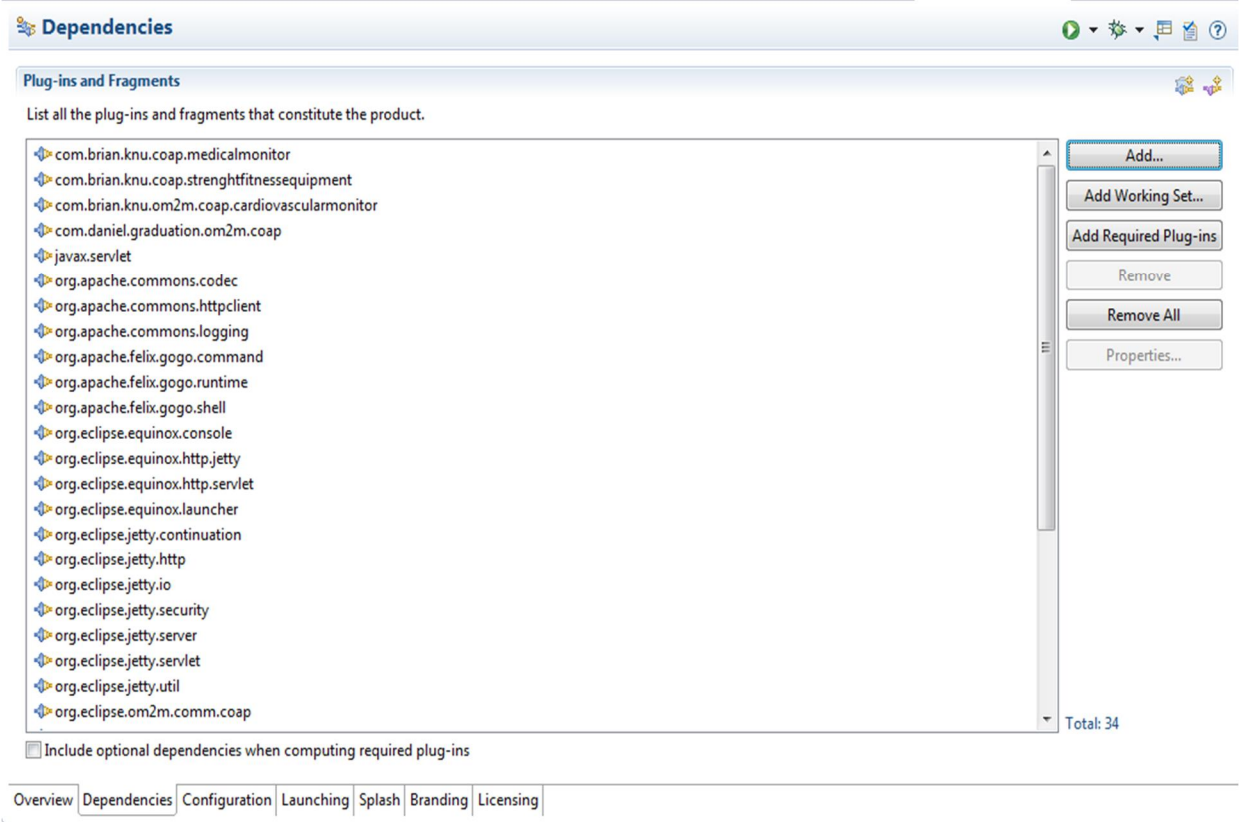
The OHP-M2M is added as a Maven project in your workspace called com.brian.knu.strenghtfitnessequipment



3. Open the pom.xml file found in the org.elcipse.om2m and under the Modules click on Add button and add the com.brian.knu.strenghtfitnessequipment module.

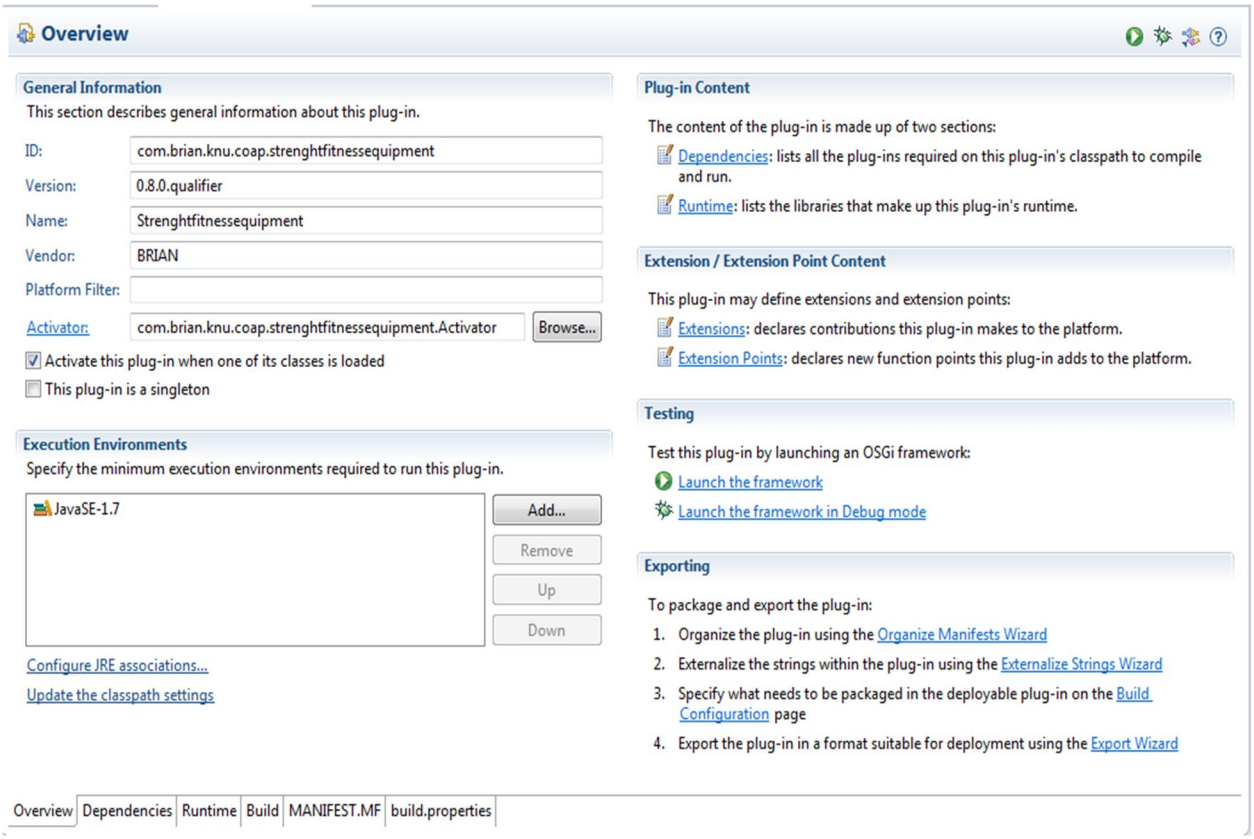


4. In the org.elcipse.om2m ⟶ org.eclipse.om2m.site.gscl, open the om2m.product file and under the dependencies, add the imported file in the OM2M.



NOTE

In case there is any problem with the installation check out the version filed in the Manifest.MF file in the com.brian.knu.coap.strenghtfitnessequipment folder.


Overview
General Information
This section describes general information about this plug-in.
ID:
com.brian.knu.coap.strenghtfitnessequipment
Version:
0.8.0.qualifier
Name:
Strenghtfitnessequipment
Vendor:
BRIAN
Platform Filter:
Activator:
com.brian.knu.coap.strenghtfitnessequipment.Activator
Browse...
Activate this plug-in when one of its classes is loaded
This plug-in is a singleton
Execution Environments
Specify the minimum execution environments required to run this plug-in.
JavaSE-1.7
Add...
Remove
Up
Down
Configure JRE associations...
Update the classpath settings
Plug-in Content
The content of the plug-in is made up of two sections:
Dependencies: lists all the plug-ins required on this plug-in's classpath to compile and run.
Runtime: lists the libraries that make up this plug-in's runtime.
Extension / Extension Point Content
This plug-in may define extensions and extension points:
Extensions: declares contributions this plug-in makes to the platform.
Extension Points: declares new function points this plug-in adds to the platform.
Testing
Test this plug-in by launching an OSGi framework:
Launch the framework
Launch the framework in Debug mode
Exporting
To package and export the plug-in:
1. Organize the plug-in using the Organize Manifests Wizard
2. Externalize the strings within the plug-in using the Externalize Strings Wizard
3. Specify what needs to be packaged in the deployable plug-in on the Build Configuration page
4. Export the plug-in in a format suitable for deployment using the Export Wizard
Overview
Dependencies
Runtime
Build
MANIFEST.MF
build.properties

How to use the OHP-M2M

1. Update the com.brian.knu.coap.strenghtfitnessequipment by Right clicking on the project. Maven ⟶ Update Project.
2. Click on Right Click on the org.eclipse.om2m master folder, Run As ⟶ Maven Maven Install and ensure the build is success full.

```
[INFO] Installing C:\Users\Brian\git\org.eclipse.om2m\org.eclipse.om2m.site.gscl\target\p2artifacts.xml to C:\Users\Brian\.m2\repository\org\eclipse\om2m\org.eclipse.om
[INFO]
[INFO] --- tycho-p2-plugin:0.20.0:update-local-index (default-update-local-index) @ org.eclipse.om2m.site.gscl ---
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] org.eclipse.om2m :: parent ....................... SUCCESS [0.095s]
[INFO] org.eclipse.om2m :: commons ....................... SUCCESS [0.895s]
[INFO] org.eclipse.om2m :: comm service .................. SUCCESS [0.043s]
[INFO] org.eclipse.om2m :: ipu service ................... SUCCESS [0.053s]
[INFO] org.eclipse.om2m :: core service .................. SUCCESS [0.053s]
[INFO] org.eclipse.om2m :: core .......................... SUCCESS [0.735s]
[INFO] org.eclipse.om2m :: comm http ..................... SUCCESS [0.041s]
[INFO] org.eclipse.om2m :: comm coap ..................... SUCCESS [0.049s]
[INFO] org.eclipse.om2m :: sample ipu .................... SUCCESS [0.057s]
[INFO] org.eclipse.om2m :: webapp resourcesbrowser ....... SUCCESS [0.046s]
[INFO] org.eclipse.om2m :: nscl product .................. SUCCESS [4.099s]
[INFO] com.brian.knu.coap.medicalmonitor ................. SUCCESS [0.113s]
[INFO] com.brian.knu.coap.strenghtfitnessequipment ....... SUCCESS [0.097s]
[INFO] com.brian.knu.om2m.coap.cardiovascularmonitor ..... SUCCESS [0.092s]
[INFO] com.daniel.graduation :: OM2MCoap ................. SUCCESS [0.110s]
[INFO] org.eclipse.om2m.sample.ipu ....................... SUCCESS [0.036s]
[INFO] org.eclipse.om2m :: gscl product .................. SUCCESS [4.328s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 33.130s
[INFO] Finished at: Wed May 13 16:33:30 KST 2015
[INFO] Final Memory: 94M/789M
[INFO] ------------------------------------------------------------------------
```

3. Run the UHealthServer and get the IP of the server. Open the StrenghtFitnessMonitor file and add the IP of the server as shown below.

StrengthFitnessMonitor.java

```
                break;
            case 2650:
                Compound systemTypeSpecList = new SystemTypeSpecList(deviceName);

                makeCompound(systemTypeSpecList, ele);
                cvDIM.addCompound(systemTypeSpecList);
                break;
            default:
                break;
            }
        }

    };

    public void listenToCV() {
        new Thread() {

            public void observerStrengthFitness(String uri) {

                CoapClient client = new CoapClient(uri + "/strenghtfitnessmonitor"); // This creates a coap client
                                                                                     //listen to strengthfitness

                CoapObserveRelation relation1 = client
                        .observe(myStrenghtFitnessHandler);
            }

            @Override
            public void run() {
                // TODO Auto-generated method stub

                observerStrengthFitness("coap://192.168.1.12:5683");


            }

        }.start();
    }
```

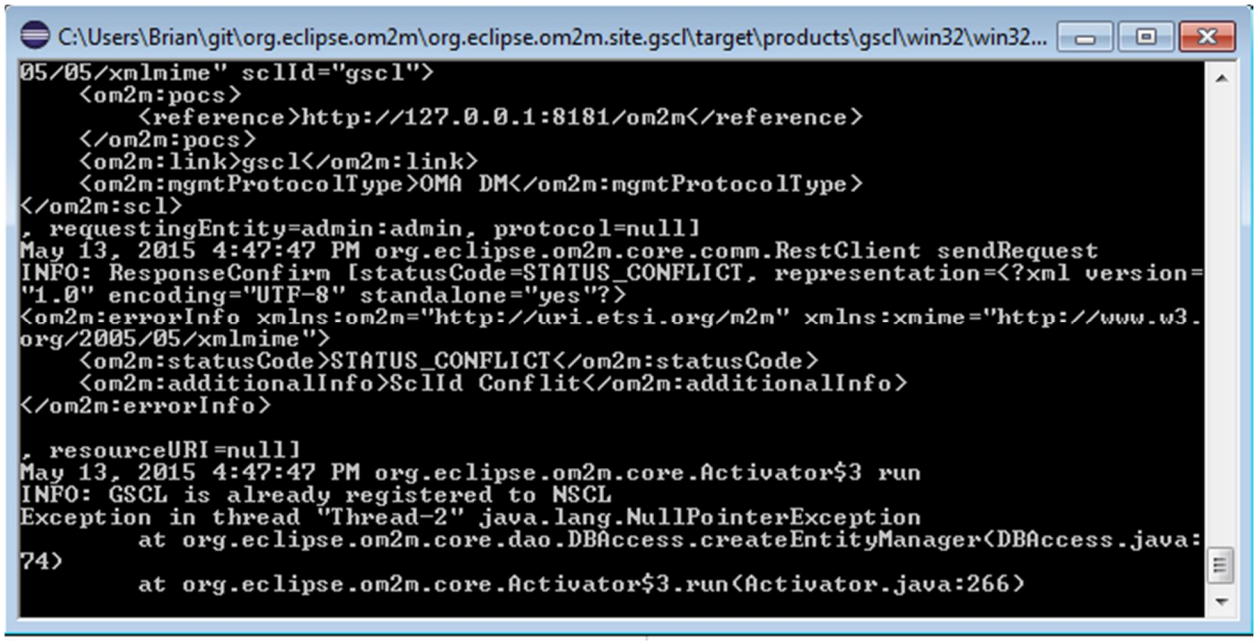
1. Open the C:\Users\UserName\git\org.eclipse.om2m\org.eclipse.om2m.site.nscl\target\products\nscl\OperatingSystempath and run the Start.bat file.

C:\Windows\system32\cmd.exe

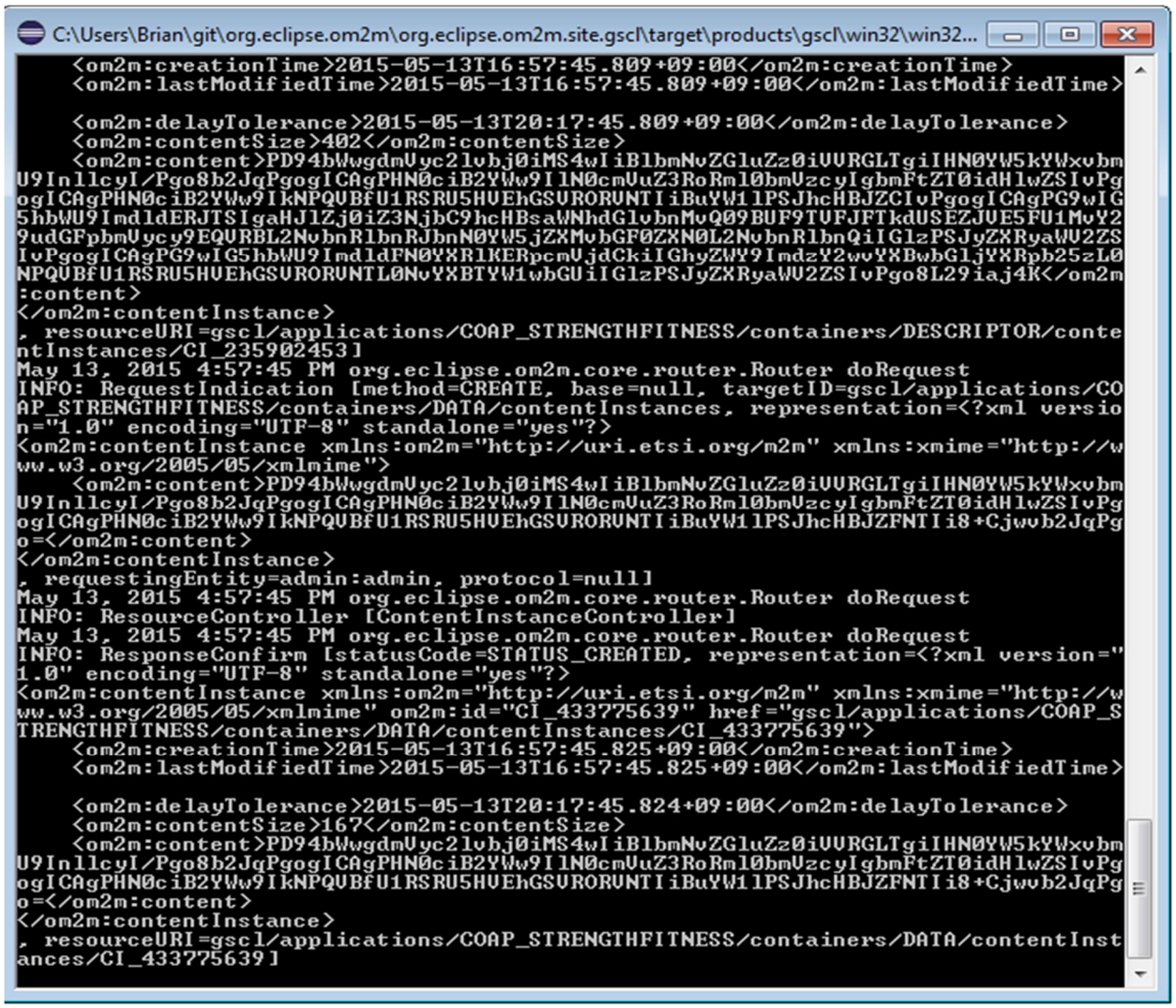
```
<om2m:discovery xmlns:om2m="http://uri.etsi.org/m2m" xmlns:xmime="http://www.w3.
org/2005/05/xmlmime">
    <om2m:matchSize>2</om2m:matchSize>
    <om2m:discoveryURI>
        <reference>nscl/accessRights/AR_ADMIN</reference>
        <reference>nscl</reference>
    </om2m:discoveryURI>
</om2m:discovery>
, resourceURI=null]
May 13, 2015 4:44:29 PM org.eclipse.om2m.comm.http.RestHttpServlet service
INFO: HttpResponse [statusCode=200]
May 13, 2015 4:44:29 PM org.eclipse.om2m.core.comm.RestClient sendRequest
INFO: ResponseConfirm [statusCode=STATUS_OK, representation=<?xml version="1.0"
encoding="UTF-8" standalone="yes"?>
<om2m:discovery xmlns:om2m="http://uri.etsi.org/m2m" xmlns:xmime="http://www.w3.
org/2005/05/xmlmime">
    <om2m:matchSize>2</om2m:matchSize>
    <om2m:discoveryURI>
        <reference>nscl/accessRights/AR_ADMIN</reference>
        <reference>nscl</reference>
    </om2m:discoveryURI>
</om2m:discovery>

, resourceURI=null]
```

2. Open the C:\Users\UserName\git\org.eclipse.om2m\org.eclipse.om2m.site.gscl\target\products\gscl\ OperatingSystempath and run the start.bat file



3. Type SS in gscl command line to list all your om2m projects. Start the project that you want to run by typing in the command line Start [Project Number] as listed in the command line.



4. Run the local server IP 127.0.0.1:8080 in the URL of a browser.

   The Om2m page appears and with the basic configuration use

   Username=”admin” and Password =”admin” to access the resources.

5. After you login into the OM2M platform, you can find the medical device resources in gscl directory.