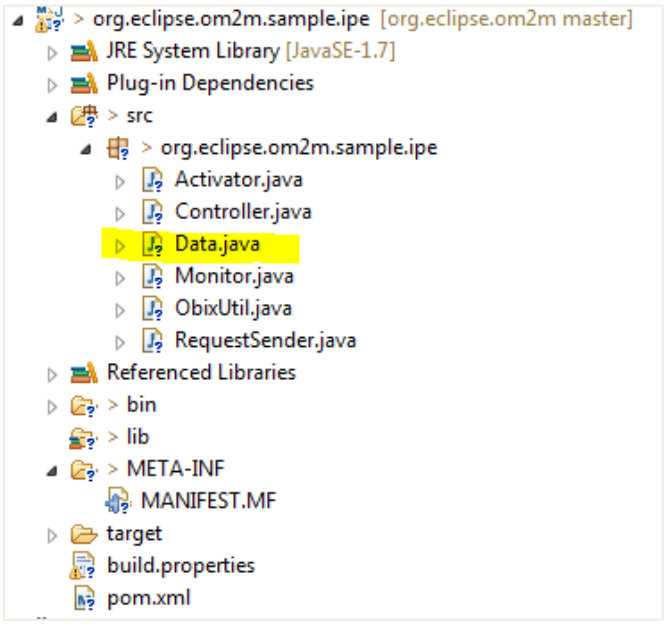
Usage example using programatical solution:

- In your recent created sample plugin project reate new class named Data.java



- In this class you have to declare which data will Cep server process. In our demo we will process only double values from the sensor. Modify it for your needs.

```
package org.eclipse.om2m.sample.ipe;

import si.fri.mag.gasperin.cep.utils.DataInterface;

public class Data implements DataInterface{
    double value;

    public Data(double value) {
        this.value = value;
    }

    public double getValue() {
        return value;
    }

    @Override
    public String toString() {
        return "" + value;
    }
}
```

- Open Monitor.java file. On top of that class add:
static CepHttpServlet cepServer;

- In constructor method of Monitor class add:
cepServer =
new CepHttpServlet(cseService,Data.class);

- In start() method of class add:
*cepServer.addCepRule(sensorId, "CEP_DATA", "select * from Data.win:length(2) having avg(value) > 50");*
cepServer.run();

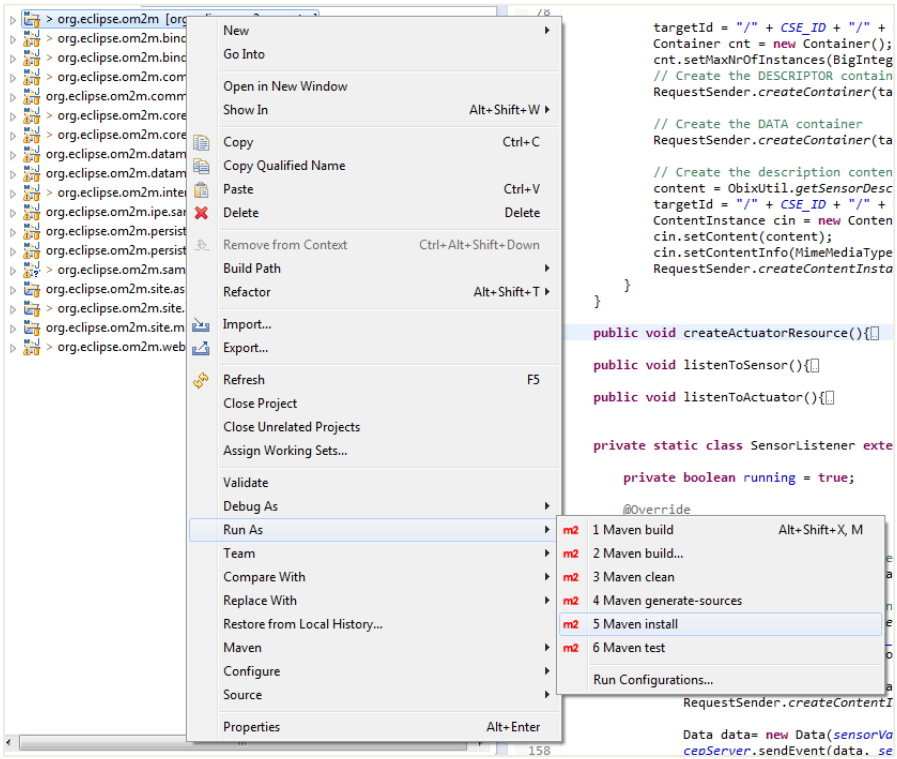
- In stop() method add:
cepServer.stopThread();

- In createSensorResources() method add:
cepServer.insertDevice(sensorId);

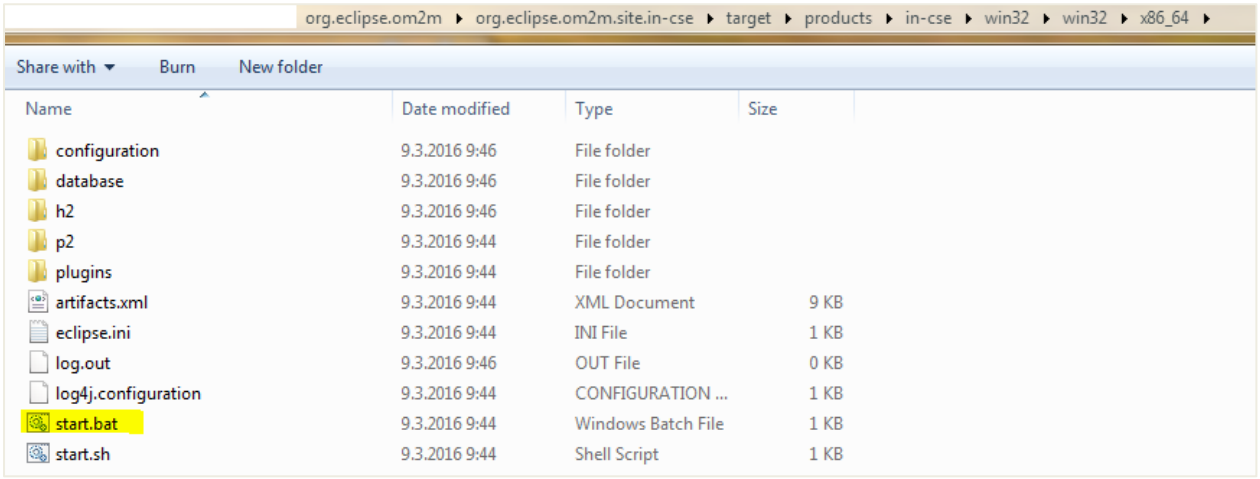
- In class SensorListener, in method run() add:
Data data= new Data(sensorValue);
cepServer.sendEvent(data, sensorId);

```
16  public class Monitor {
17
18      static CseService CSE;
19      static String CSE_ID = Constants.CSE_ID;
20      static String CSE_NAME = Constants.CSE_NAME;
21      static String REQUEST_ENTITY = Constants.ADMIN_REQUESTING_ENTITY;
22      static String ipeId = "sample";
23      static String actuatorId = "MY_ACTUATOR";
24      static String sensorId = "MY_SENSOR";
25      static boolean actuatorValue = false;
26      static int sensorValue = 0;
27      static String DESCRIPTOR = "DESCRIPTOR";
28      static String DATA = "DATA";
29
30      static CepHttpServlet cepServer;
31
32      private SensorListener sensorListener;
33      private ActuatorListener actuatorListener;
34
35      public Monitor(CseService cseService){
36          CSE = cseService;
37          cepServer = new CepHttpServlet(cseService, Data.class);
38      }
39
40      public void start(){
41          // Create sensor resources
42          createSensorResources();
43          // Listen for the sensor data
44          listenToSensor();
45
46          // Create required resources for the Actuator
47          createActuatorResource();
48          // Listen for the Actuator data
49          listenToActuator();
50          cepServer.addCepRule(sensorId, "CEP_DATA", "select * from Data.win:length(2) having avg(value) > 50");
51          cepServer.run();
52      }
53
54      public void stop(){
55          if(sensorListener != null && sensorListener.isAlive()){
56              sensorListener.stopThread();
57          }
58          if(actuatorListener != null && actuatorListener.isAlive()){
59              actuatorListener.stopThread();
60          }
61          cepServer.stopThread();
62      }
63
64      public void createSensorResources(){
65          String targetId, content;
66
67          targetId = "/" + CSE_ID + "/" + CSE_NAME;
68          AE ae = new AE();
69          ae.setRequestReachability(true);
70          ae.setAppID(ipeId);
71          ae.getPointOfAccess().add(ipeId);
72          ResponsePrimitive response = RequestSender.createAE(ae, sensorId);
73
74          if(response.getResponseStatusCode().equals(ResponseStatusCode.CREATED)){
75
76              cepServer.insertDevice(sensorId);
77
79              targetId = "/" + CSE_ID + "/" + CSE_NAME + "/" + sensorId;
80              Container cnt = new Container();
81              cnt.setMaxNrOfInstances(BigInteger.valueOf(10));
82              // Create the DESCRIPTOR container
83              RequestSender.createContainer(targetId, DESCRIPTOR, cnt);
84
85              // Create the DATA container
86              RequestSender.createContainer(targetId, DATA, cnt);
87
88              // Create the description contentInstance
89              content = ObixUtil.getSensorDescriptorRep(sensorId, ipeId);
90              targetId = "/" + CSE_ID + "/" + CSE_NAME + "/" + sensorId + "/" + DESCRIPTOR;
91              ContentInstance cin = new ContentInstance();
92              cin.setContent(content);
93              cin.setContentInfo(MimeMediaType.OBIX);
94              RequestSender.createContentInstance(targetId, cin);
95          }
96      }
97
98      public void createActuatorResource(){
127
128     public void listenToSensor(){
132
133     public void listenToActuator(){
137
138
139     private static class SensorListener extends Thread{
140
141         private boolean running = true;
142
143         @Override
144         public void run() {
145             while(running){
146                 // Simulate a random measurement of the sensor
147                 sensorValue = 10 + (int) (Math.random() * 100);
148
149                 // Create the data contentInstance
150                 String content = ObixUtil.getSensorDataRep(sensorValue);
151                 String targetId = "/" + CSE_ID + "/" + CSE_NAME + "/" + sensorId + "/" + DATA;
152                 ContentInstance cin = new ContentInstance();
153                 cin.setContent(content);
154                 cin.setContentInfo(MimeMediaType.OBIX);
155                 RequestSender.createContentInstance(targetId, cin);
156
157                 Data data= new Data(sensorValue);
158                 cepServer.sendEvent(data, sensorId);
159
160
161                 try {
162                     Thread.sleep(2000);
163                 } catch (InterruptedException e){
164                     e.printStackTrace();
165                 }
166             }
167
168         }
169
170         public void stopThread(){
171             running = false;
172         }
173
174     }
175
176     private static class ActuatorListener extends Thread{
212
213 }
```

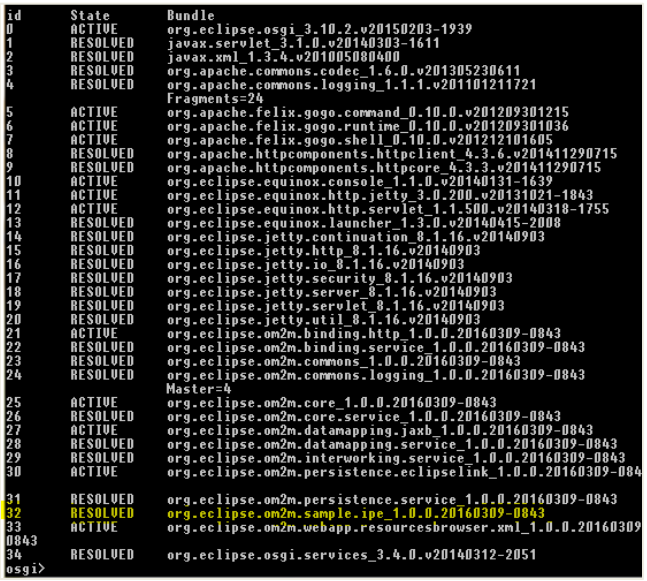
- Right click on org.eclipse.om2m project -> Run as -> Maven install



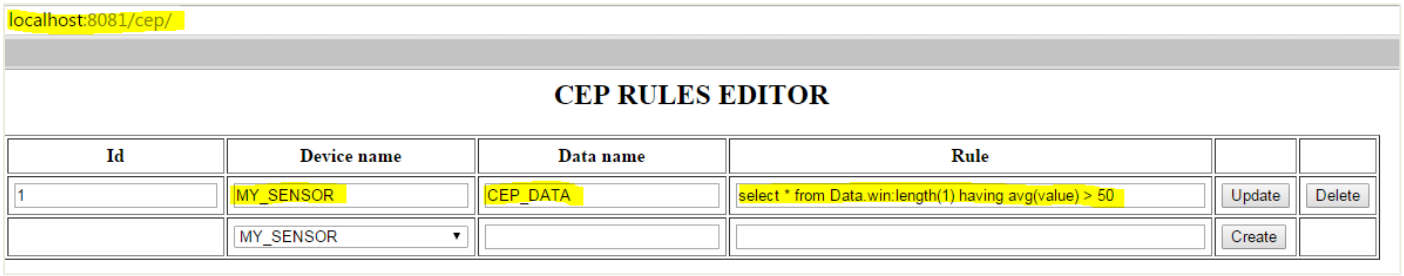
- Open explorer and go to:
 org.eclipse.om2m\org.eclipse.om2m.site.in-cse\target\products\in-cse\win32\win32\x86_64
- Double click on start.bat



- Type "ss" into comand line and look for id of your recently created plugin (e.g. org.eclipse.om2m.sample.ipe)
- Type "start id" (e.g. "start 32")



Check your new cep rule:
- Open "http://localhost:8081/cep" and login (default values for login are also "admin" : "admin"). Here you can add/delete/update cep rules for specific device.
- You should see already created cep rule in the CEP rules table editor.



Check your om2m platform:
- Open web browser, type into url "http://localhost:8080/webpage" and login (default values for login are "admin" : "admin"). Here you can monitor your data and data catched into predefined cep rules.
- Under MY_SENSOR -> DATA are all sensor measurments sent to OM2M platform
- Under MY_SENSOR -> CEP_DATA are all triggered messages by predefined cep rule