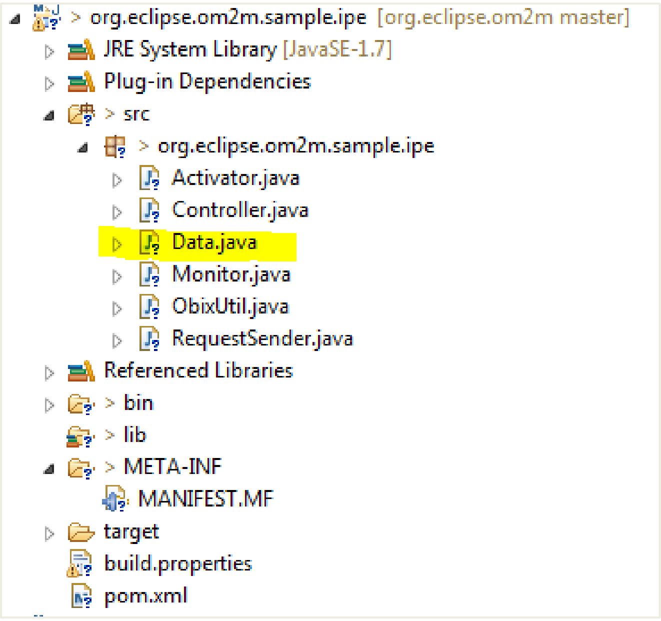
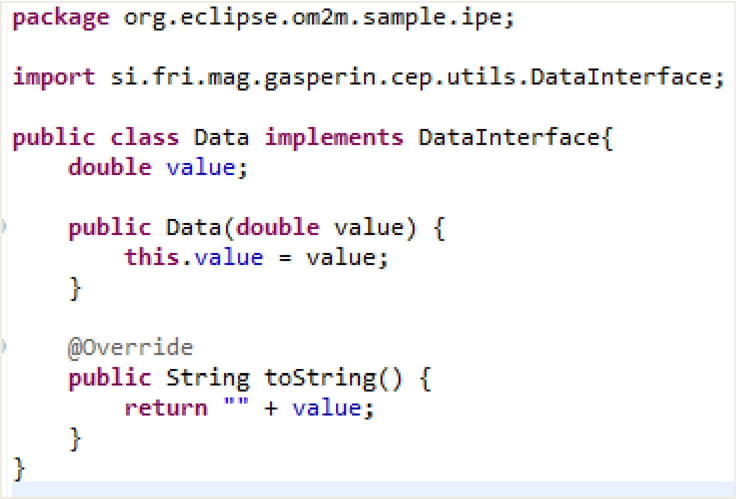
Usage example using GUI:

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



```
> org.eclipse.om2m.sample.ipe [org.eclipse.om2m master]
    JRE System Library [JavaSE-1.7]
    Plug-in Dependencies
    > src
        > org.eclipse.om2m.sample.ipe
            Activator.java
            Controller.java
            Data.java
            Monitor.java
            ObixUtil.java
            RequestSender.java
    Referenced Libraries
    > bin
    > lib
    > META-INF
        MANIFEST.MF
    target
    build.properties
    pom.xml
```

- 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.



```java
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;
    }

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

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

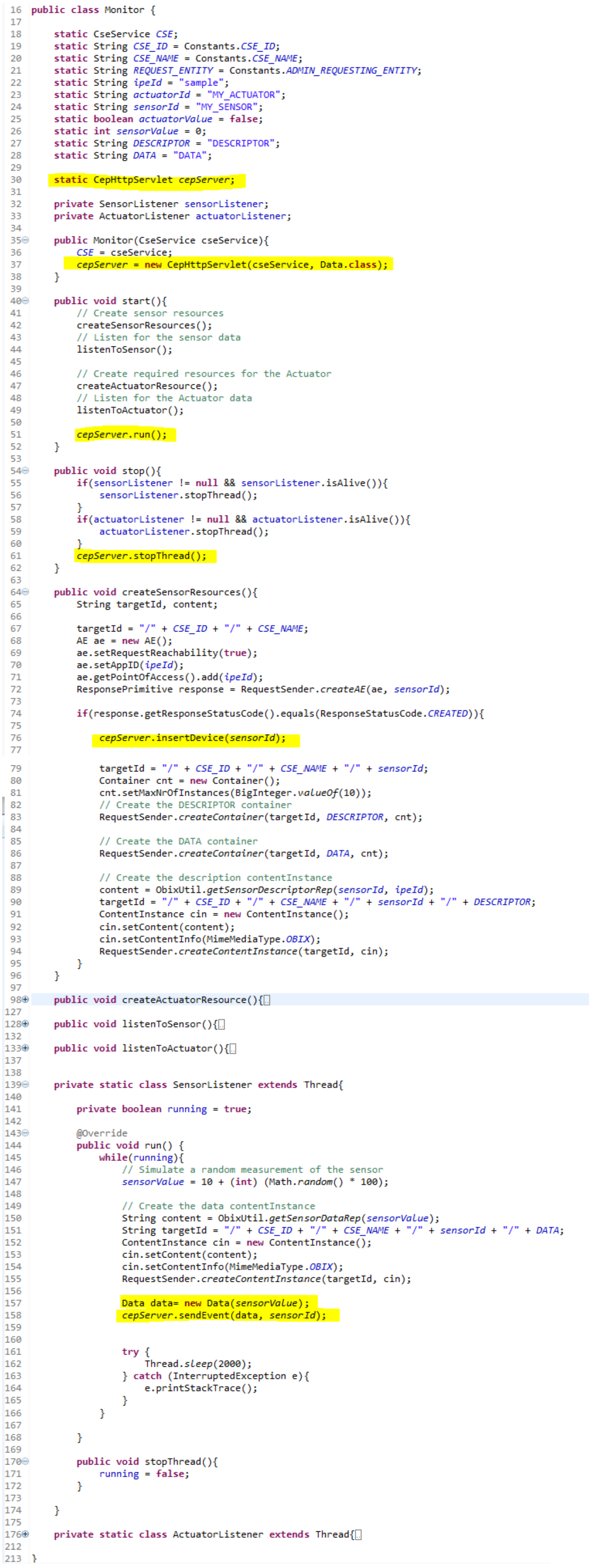
- In constructor method of Monitor class add:
cepServer = new CepHttpServlet(cseService,Data.class);

- In start() method of class add:
cepServer.run();

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

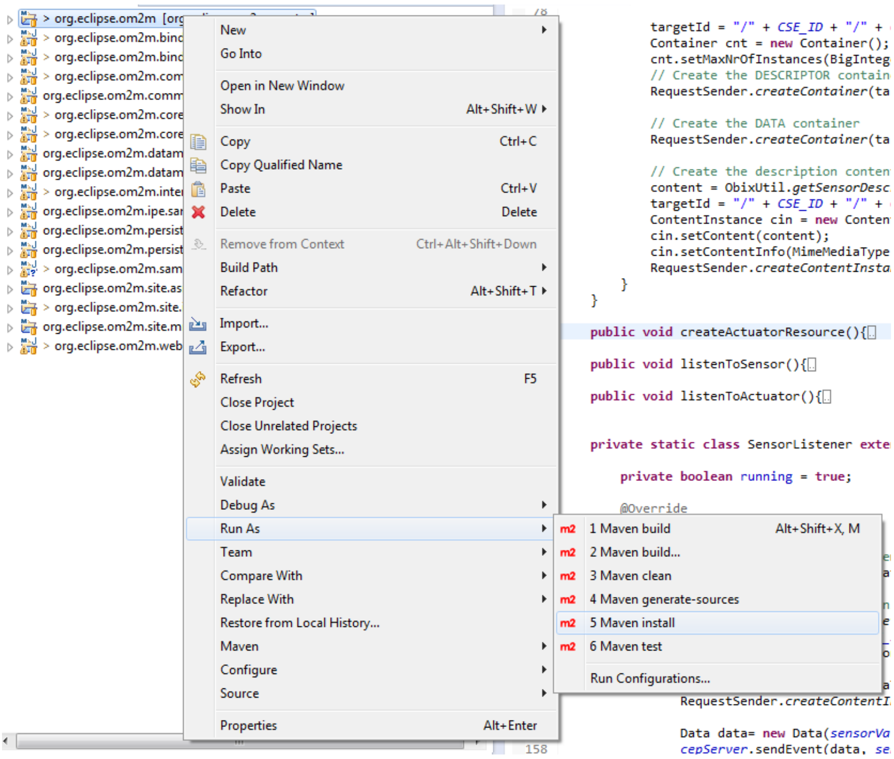
- In createSensorResources() method add:
cepServer.insertDevice(sensorId);

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



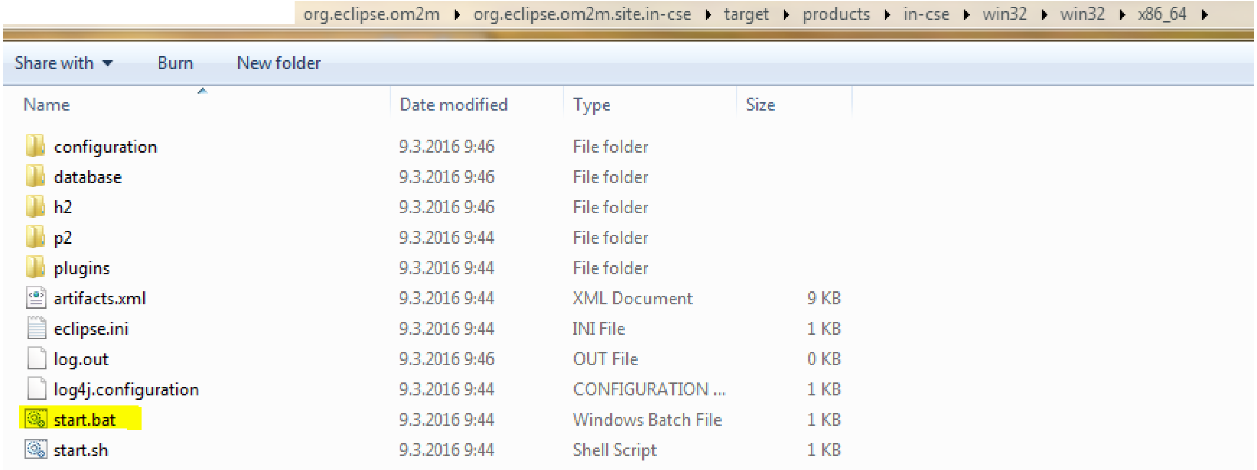
```java
public class Monitor {

    static CseService CSE;
    static String CSE_ID = Constants.CSE_ID;
    static String CSE_NAME = Constants.CSE_NAME;
    static String REQUEST_ENTITY = Constants.ADMIN_REQUESTING_ENTITY;
    static String ipeId = "sample";
    static String actuatorId = "MY_ACTUATOR";
    static String sensorId = "MY_SENSOR";
    static boolean actuatorValue = false;
    static int sensorValue = 0;
    static String DESCRIPTOR = "DESCRIPTOR";
    static String DATA = "DATA";

    static CepHttpServlet cepServer;

    private SensorListener sensorListener;
    private ActuatorListener actuatorListener;

    public Monitor(CseService cseService){
        CSE = cseService;
        cepServer = new CepHttpServlet(cseService, Data.class);
    }

    public void start(){
        // Create sensor resources
        createSensorResources();
        // Listen for the sensor data
        listenToSensor();

        // Create required resources for the Actuator
        createActuatorResource();
        // Listen for the Actuator data
        listenToActuator();

        cepServer.run();
    }

    public void stop(){
        if(sensorListener != null && sensorListener.isAlive()){
            sensorListener.stopThread();
        }
        if(actuatorListener != null && actuatorListener.isAlive()){
            actuatorListener.stopThread();
        }
        cepServer.stopThread();
    }

    public void createSensorResources(){
        String targetId, content;

        targetId = "/" + CSE_ID + "/" + CSE_NAME;
        AE ae = new AE();
        ae.setRequestReachability(true);
        ae.setAppID(ipeId);
        ae.getPointOfAccess().add(ipeId);
        ResponsePrimitive response = RequestSender.createAE(ae, sensorId);

        if(response.getResponseStatusCode().equals(ResponseStatusCode.CREATED)){

            cepServer.insertDevice(sensorId);

            targetId = "/" + CSE_ID + "/" + CSE_NAME + "/" + sensorId;
            Container cnt = new Container();
            cnt.setMaxNrOfInstances(BigInteger.valueOf(10));
            // Create the DESCRIPTOR container
            RequestSender.createContainer(targetId, DESCRIPTOR, cnt);

            // Create the DATA container
            RequestSender.createContainer(targetId, DATA, cnt);

            // Create the description contentInstance
            content = ObixUtil.getSensorDescriptorRep(sensorId, ipeId);
            targetId = "/" + CSE_ID + "/" + CSE_NAME + "/" + sensorId + "/" + DESCRIPTOR;
            ContentInstance cin = new ContentInstance();
            cin.setContent(content);
            cin.setContentInfo(MimeMediaType.OBIX);
            RequestSender.createContentInstance(targetId, cin);
        }
    }

    public void createActuatorResource(){□

    public void listenToSensor(){□

    public void listenToActuator(){□


    private static class SensorListener extends Thread{

        private boolean running = true;

        @Override
        public void run() {
            while(running){
                // Simulate a random measurement of the sensor
                sensorValue = 10 + (int) (Math.random() * 100);

                // Create the data contentInstance
                String content = ObixUtil.getSensorDataRep(sensorValue);
                String targetId = "/" + CSE_ID + "/" + CSE_NAME + "/" + sensorId + "/" + DATA;
                ContentInstance cin = new ContentInstance();
                cin.setContent(content);
                cin.setContentInfo(MimeMediaType.OBIX);
                RequestSender.createContentInstance(targetId, cin);

                Data data= new Data(sensorValue);
                cepServer.sendEvent(data, sensorId);


                try {
                    Thread.sleep(2000);
                } catch (InterruptedException e){
                    e.printStackTrace();
                }
            }

        }

        public void stopThread(){
            running = false;
        }

    }

    private static class ActuatorListener extends Thread{□

}
```

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



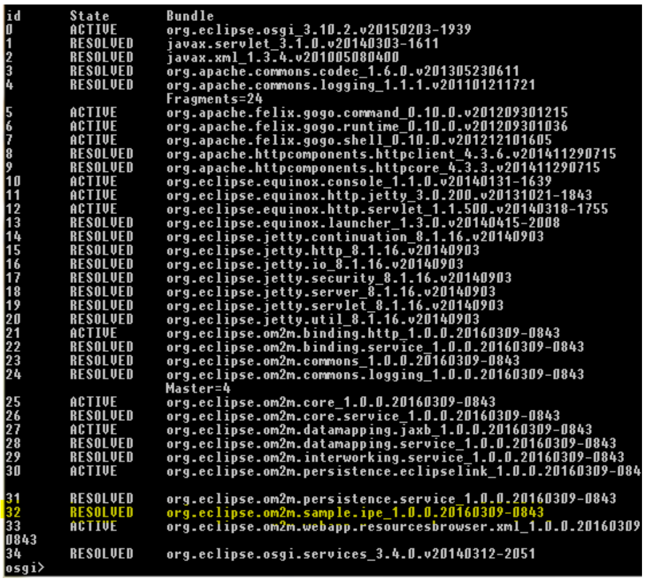
```
            targetId = "/" + CSE_ID + "/" +
            Container cnt = new Container();
            cnt.setMaxNrOfInstances(BigInteg
            // Create the DESCRIPTOR contain
            RequestSender.createContainer(ta

            // Create the DATA container
            RequestSender.createContainer(ta

            // Create the description conten
            content = ObixUtil.getSensorDesc
            targetId = "/" + CSE_ID + "/" +
            ContentInstance cin = new Conten
            cin.setContent(content);
            cin.setContentInfo(MimeMediaType
            RequestSender.createContentInsta
        }
    }

    public void createActuatorResource(){

    public void listenToSensor(){

    public void listenToActuator(){


    private static class SensorListener exte

        private boolean running = true;

        @Override

                RequestSender.createContentI

                Data data= new Data(sensorVa
                cepServer.sendEvent(data, se
```

- 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


org.eclipse.om2m ▸ org.eclipse.om2m.site.in-cse ▸ target ▸ products ▸ in-cse ▸ win32 ▸ win32 ▸ x86_64 ▸
Share with ▾
Burn
New folder

Name	Date modified	Type	Size
configuration	9.3.2016 9:46	File folder	
database	9.3.2016 9:46	File folder	
h2	9.3.2016 9:46	File folder	
p2	9.3.2016 9:44	File folder	
plugins	9.3.2016 9:44	File folder	
artifacts.xml	9.3.2016 9:44	XML Document	9 KB
eclipse.ini	9.3.2016 9:44	INI File	1 KB
log.out	9.3.2016 9:46	OUT File	0 KB
log4j.configuration	9.3.2016 9:44	CONFIGURATION ...	1 KB
start.bat	9.3.2016 9:44	Windows Batch File	1 KB
start.sh	9.3.2016 9:44	Shell Script	1 KB

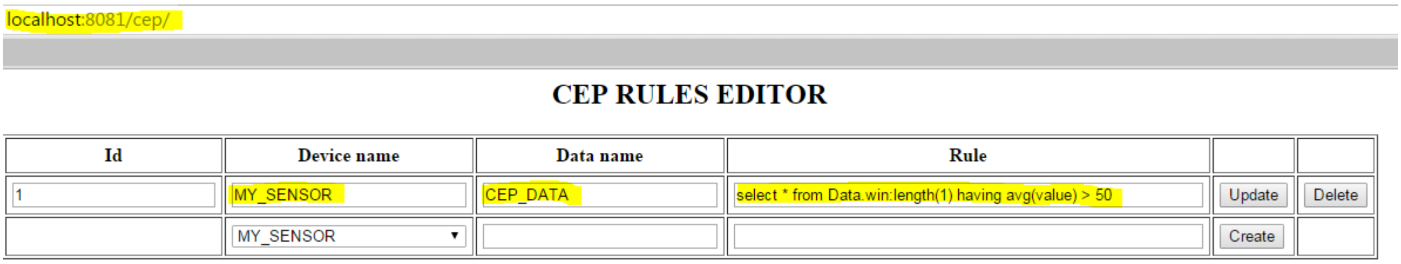
- Type "ss" into comand line and look for id of your recently created plugin (e.g. org.eclipse.om2m.sample.ipe)
- Type "start id" (e.g. "start 32")



```
id      State       Bundle
0       ACTIVE      org.eclipse.osgi_3.10.2.v20150203-1939
1       RESOLVED    javax.servlet_3.1.0.v20140303-1611
2       RESOLVED    javax.xml_1.3.4.v201005080400
3       RESOLVED    org.apache.commons.codec_1.6.0.v201305230611
4       RESOLVED    org.apache.commons.logging_1.1.1.v201101211721
                    Fragments=24
5       ACTIVE      org.apache.felix.gogo.command_0.10.0.v201209301215
6       ACTIVE      org.apache.felix.gogo.runtime_0.10.0.v201209301036
7       ACTIVE      org.apache.felix.gogo.shell_0.10.0.v201212101605
8       RESOLVED    org.apache.httpcomponents.httpclient_4.3.6.v201411290715
9       RESOLVED    org.apache.httpcomponents.httpcore_4.3.3.v201411290715
10      ACTIVE      org.eclipse.equinox.console_1.1.0.v20140131-1639
11      ACTIVE      org.eclipse.equinox.http.jetty_3.0.200.v20131021-1843
12      ACTIVE      org.eclipse.equinox.http.servlet_1.1.500.v20140318-1755
13      RESOLVED    org.eclipse.equinox.launcher_1.3.0.v20140415-2008
14      RESOLVED    org.eclipse.jetty.continuation_8.1.16.v20140903
15      RESOLVED    org.eclipse.jetty.http_8.1.16.v20140903
16      RESOLVED    org.eclipse.jetty.io_8.1.16.v20140903
17      RESOLVED    org.eclipse.jetty.security_8.1.16.v20140903
18      RESOLVED    org.eclipse.jetty.server_8.1.16.v20140903
19      RESOLVED    org.eclipse.jetty.servlet_8.1.16.v20140903
20      RESOLVED    org.eclipse.jetty.util_8.1.16.v20140903
21      ACTIVE      org.eclipse.om2m.binding.http_1.0.0.20160309-0843
22      RESOLVED    org.eclipse.om2m.binding.service_1.0.0.20160309-0843
23      RESOLVED    org.eclipse.om2m.commons_1.0.0.20160309-0843
24      RESOLVED    org.eclipse.om2m.commons.logging_1.0.0.20160309-0843
                    Master=4
25      ACTIVE      org.eclipse.om2m.core_1.0.0.20160309-0843
26      RESOLVED    org.eclipse.om2m.core.service_1.0.0.20160309-0843
27      ACTIVE      org.eclipse.om2m.datamapping.jaxb_1.0.0.20160309-0843
28      RESOLVED    org.eclipse.om2m.datamapping.service_1.0.0.20160309-0843
29      RESOLVED    org.eclipse.om2m.interworking.service_1.0.0.20160309-0843
30      ACTIVE      org.eclipse.om2m.persistence.eclipselink_1.0.0.20160309-084

31      RESOLVED    org.eclipse.om2m.persistence.service_1.0.0.20160309-0843
32      RESOLVED    org.eclipse.om2m.sample.ipe_1.0.0.20160309-0843
33      ACTIVE      org.eclipse.om2m.webapp.resourcesbrowser.xml_1.0.0.20160309
0843
34      RESOLVED    org.eclipse.osgi.services_3.4.0.v20140312-2051
osgi>
```

Add 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.
- Under Device name select "MY_SENSOR"
- Under Data name type "CEP_DATA"
- Under Rule type "select * from Data.win:length(2) having avg(value) > 50" (which means if OM2M recieves two sequential sensor measurments which average is bigger then 50, then cep rule is triggered)
- Click Create



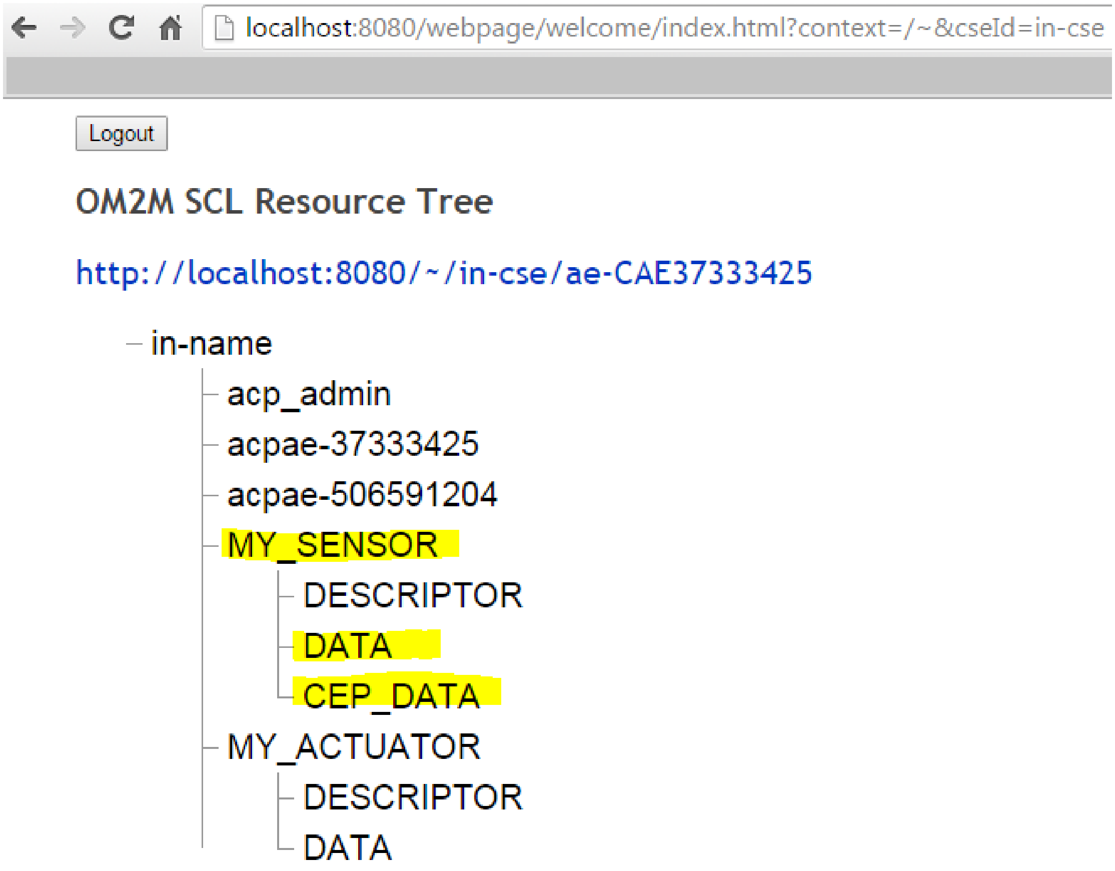
localhost:8081/cep/

CEP RULES EDITOR

Id	Device name	Data name	Rule		
1	MY_SENSOR	CEP_DATA	select * from Data.win:length(1) having avg(value) > 50	Update	Delete
	MY_SENSOR ▾			Create	

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



localhost:8080/webpage/welcome/index.html?context=/~&cseId=in-cse

OM2M SCL Resource Tree

http://localhost:8080/~/in-cse/ae-CAE37333425

```
- in-name
    acp_admin
    acpae-37333425
    acpae-506591204
    MY_SENSOR
        DESCRIPTOR
        DATA
        CEP_DATA
    MY_ACTUATOR
        DESCRIPTOR
        DATA
```