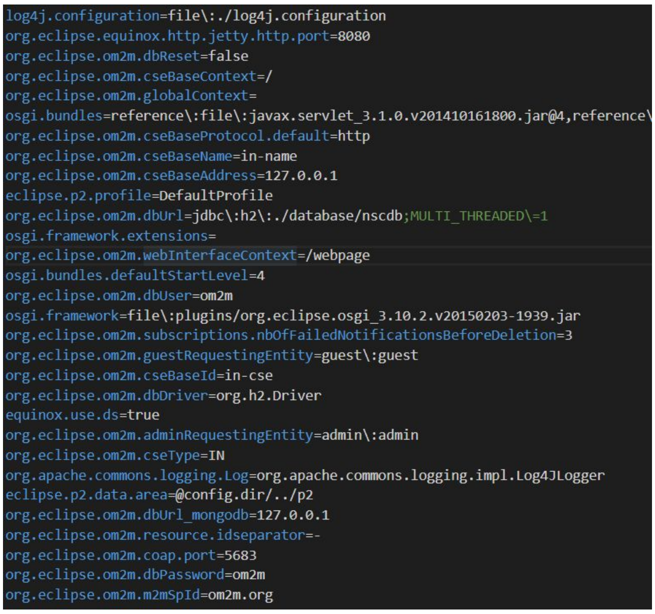
Middleware for IoT
Interact with the oneM2M RESTful architecture using Eclipse OM2M

BERRADA El Ghali
BERTA Pauline
CONCEICAO NUNES Joao

Configure the platform

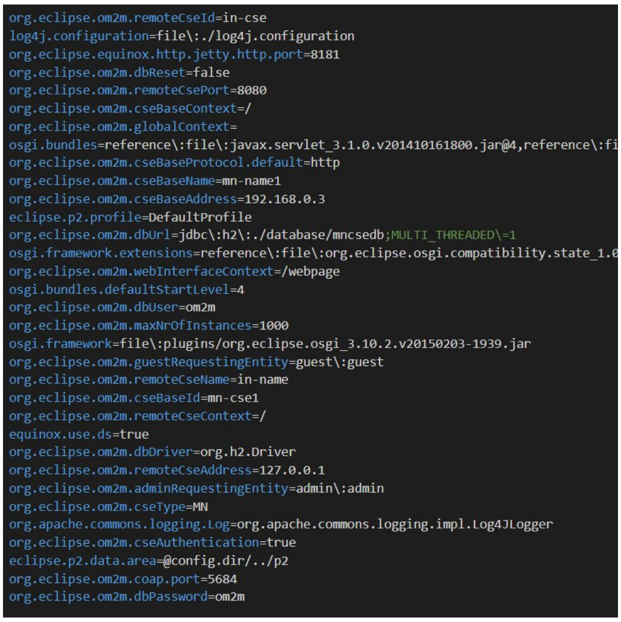
For the first steps, we chose the following configuration to the Server :

IN-CSE, with the Base IP address : localhost.

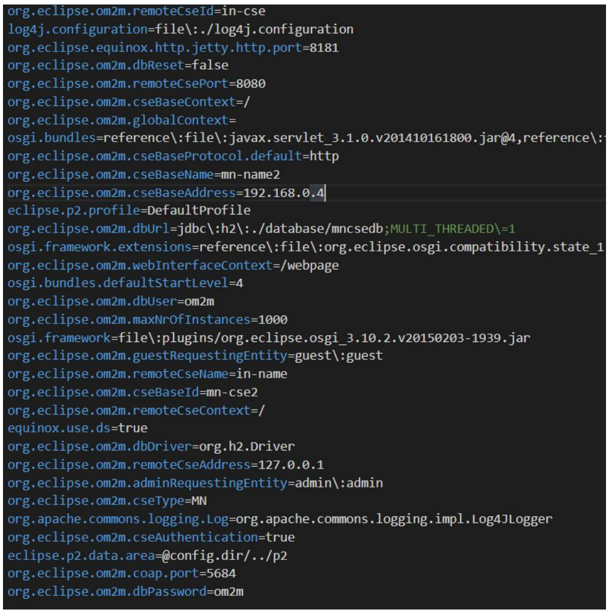


```
log4j.configuration=file\:./log4j.configuration
org.eclipse.equinox.http.jetty.http.port=8080
org.eclipse.om2m.dbReset=false
org.eclipse.om2m.cseBaseContext=/
org.eclipse.om2m.globalContext=
osgi.bundles=reference\:file\:javax.servlet_3.1.0.v201410161800.jar@4,reference\
org.eclipse.om2m.cseBaseProtocol.default=http
org.eclipse.om2m.cseBaseName=in-name
org.eclipse.om2m.cseBaseAddress=127.0.0.1
eclipse.p2.profile=DefaultProfile
org.eclipse.om2m.dbUrl=jdbc\:h2\:./database/nscdb;MULTI_THREADED\=1
osgi.framework.extensions=
org.eclipse.om2m.webInterfaceContext=/webpage
osgi.bundles.defaultStartLevel=4
org.eclipse.om2m.dbUser=om2m
osgi.framework=file\:plugins/org.eclipse.osgi_3.10.2.v20150203-1939.jar
org.eclipse.om2m.subscriptions.nbOfFailedNotificationsBeforeDeletion=3
org.eclipse.om2m.guestRequestingEntity=guest\:guest
org.eclipse.om2m.cseBaseId=in-cse
org.eclipse.om2m.dbDriver=org.h2.Driver
equinox.use.ds=true
org.eclipse.om2m.adminRequestingEntity=admin\:admin
org.eclipse.om2m.cseType=IN
org.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger
eclipse.p2.data.area=@config.dir/../p2
org.eclipse.om2m.dbUrl_mongodb=127.0.0.1
org.eclipse.om2m.resource.idseparator=-
org.eclipse.om2m.coap.port=5683
org.eclipse.om2m.dbPassword=om2m
org.eclipse.om2m.m2mSpId=om2m.org
```

Then for the two gateways we choose the following configuration:

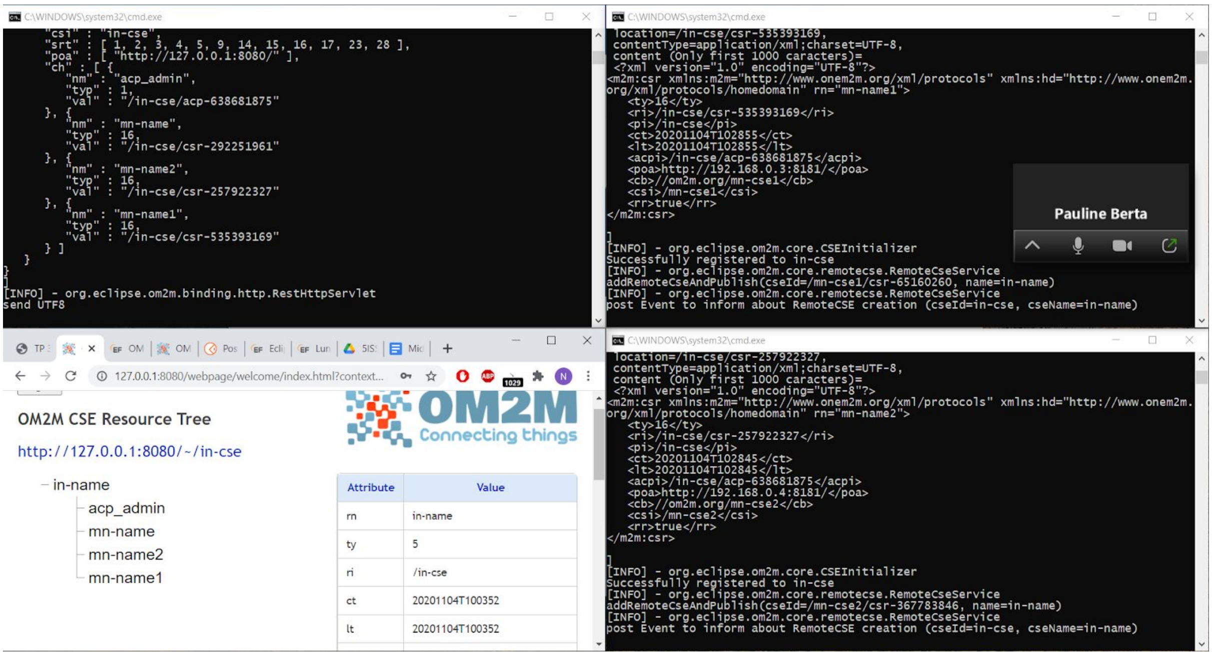


```
org.eclipse.om2m.remoteCseId=in-cse
log4j.configuration=file\:./log4j.configuration
org.eclipse.equinox.http.jetty.http.port=8181
org.eclipse.om2m.dbReset=false
org.eclipse.om2m.remoteCsePort=8080
org.eclipse.om2m.cseBaseContext=/
org.eclipse.om2m.globalContext=
osgi.bundles=reference\:file\:javax.servlet_3.1.0.v201410161800.jar@4,reference\:fi
org.eclipse.om2m.cseBaseProtocol.default=http
org.eclipse.om2m.cseBaseName=mn-name1
org.eclipse.om2m.cseBaseAddress=192.168.0.3
eclipse.p2.profile=DefaultProfile
org.eclipse.om2m.dbUrl=jdbc\:h2\:./database/mncsedb;MULTI_THREADED\=1
osgi.framework.extensions=reference\:file\:org.eclipse.osgi.compatibility.state_1.0
org.eclipse.om2m.webInterfaceContext=/webpage
osgi.bundles.defaultStartLevel=4
org.eclipse.om2m.dbUser=om2m
org.eclipse.om2m.maxNrOfInstances=1000
osgi.framework=file\:plugins/org.eclipse.osgi_3.10.2.v20150203-1939.jar
org.eclipse.om2m.guestRequestingEntity=guest\:guest
org.eclipse.om2m.remoteCseName=in-name
org.eclipse.om2m.cseBaseId=mn-cse1
org.eclipse.om2m.remoteCseContext=/
equinox.use.ds=true
org.eclipse.om2m.dbDriver=org.h2.Driver
org.eclipse.om2m.remoteCseAddress=127.0.0.1
org.eclipse.om2m.adminRequestingEntity=admin\:admin
org.eclipse.om2m.cseType=MN
org.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger
org.eclipse.om2m.cseAuthentication=true
eclipse.p2.data.area=@config.dir/../p2
org.eclipse.om2m.coap.port=5684
org.eclipse.om2m.dbPassword=om2m
```



```
org.eclipse.om2m.remoteCseId=in-cse
log4j.configuration=file\:./log4j.configuration
org.eclipse.equinox.http.jetty.http.port=8181
org.eclipse.om2m.dbReset=false
org.eclipse.om2m.remoteCsePort=8080
org.eclipse.om2m.cseBaseContext=/
org.eclipse.om2m.globalContext=
osgi.bundles=reference\:file\:javax.servlet_3.1.0.v201410161800.jar@4,reference\:
org.eclipse.om2m.cseBaseProtocol.default=http
org.eclipse.om2m.cseBaseName=mn-name2
org.eclipse.om2m.cseBaseAddress=192.168.0.4
eclipse.p2.profile=DefaultProfile
org.eclipse.om2m.dbUrl=jdbc\:h2\:./database/mncsedb;MULTI_THREADED\=1
osgi.framework.extensions=reference\:file\:org.eclipse.osgi.compatibility.state_1
org.eclipse.om2m.webInterfaceContext=/webpage
osgi.bundles.defaultStartLevel=4
org.eclipse.om2m.dbUser=om2m
org.eclipse.om2m.maxNrOfInstances=1000
osgi.framework=file\:plugins/org.eclipse.osgi_3.10.2.v20150203-1939.jar
org.eclipse.om2m.guestRequestingEntity=guest\:guest
org.eclipse.om2m.remoteCseName=in-name
org.eclipse.om2m.cseBaseId=mn-cse2
org.eclipse.om2m.remoteCseContext=/
equinox.use.ds=true
org.eclipse.om2m.dbDriver=org.h2.Driver
org.eclipse.om2m.remoteCseAddress=127.0.0.1
org.eclipse.om2m.adminRequestingEntity=admin\:admin
org.eclipse.om2m.cseType=MN
org.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger
org.eclipse.om2m.cseAuthentication=true
eclipse.p2.data.area=@config.dir/../p2
org.eclipse.om2m.coap.port=5684
org.eclipse.om2m.dbPassword=om2m
```

For the two gateways, we chose localhost as the remoteCseAddress, so they can communicate with the server. For their own IP Address, we choose two fake IP addresses. As we can see, the two gateways are connected to the server, as they should.



C:\WINDOWS\system32\cmd.exe

```
    "csi" : "in-cse",
    "srt" : [ 1, 2, 3, 4, 5, 9, 14, 15, 16, 17, 23, 28 ],
    "poa" : [ "http://127.0.0.1:8080/" ],
    "ch" : [ {
      "nm" : "acp_admin",
      "typ" : 1,
      "val" : "/in-cse/acp-638681875"
    }, {
      "nm" : "mn-name",
      "typ" : 16,
      "val" : "/in-cse/csr-292251961"
    }, {
      "nm" : "mn-name2",
      "typ" : 16,
      "val" : "/in-cse/csr-257922327"
    }, {
      "nm" : "mn-name1",
      "typ" : 16,
      "val" : "/in-cse/csr-535393169"
    } ]
  }
}
]
[INFO] - org.eclipse.om2m.binding.http.RestHttpServlet
send UTF8
```

C:\WINDOWS\system32\cmd.exe

```
 location=/in-cse/csr-535393169,
 contentType=application/xml;charset=UTF-8,
 content (Only first 1000 caracters)=
 <?xml version="1.0" encoding="UTF-8"?>
<m2m:csr xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m.
org/xml/protocols/homedomain" rn="mn-name1">
   <ty>16</ty>
   <ri>/in-cse/csr-535393169</ri>
   <pi>/in-cse</pi>
   <ct>20201104T102855</ct>
   <lt>20201104T102855</lt>
   <acpi>/in-cse/acp-638681875</acpi>
   <poa>http://192.168.0.3:8181/</poa>
   <cb>//om2m.org/mn-cse1</cb>
   <csi>/mn-cse1</csi>
   <rr>true</rr>
</m2m:csr>

]
[INFO] - org.eclipse.om2m.core.CSEInitializer
Successfully registered to in-cse
[INFO] - org.eclipse.om2m.core.remotecse.RemoteCseService
addRemoteCseAndPublish(cseId=/mn-cse1/csr-65160260, name=in-name)
[INFO] - org.eclipse.om2m.core.remotecse.RemoteCseService
post Event to inform about RemoteCSE creation (cseId=in-cse, cseName=in-name)
```

127.0.0.1:8080/webpage/welcome/index.html?context...

OM2M CSE Resource Tree

http://127.0.0.1:8080/~/in-cse

- in-name
  - acp_admin
  - mn-name
  - mn-name2
  - mn-name1

Attribute	Value
rn	in-name
ty	5
ri	/in-cse
ct	20201104T100352
lt	20201104T100352

C:\WINDOWS\system32\cmd.exe

```
 location=/in-cse/csr-257922327,
 contentType=application/xml;charset=UTF-8,
 content (Only first 1000 caracters)=
 <?xml version="1.0" encoding="UTF-8"?>
<m2m:csr xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m.
org/xml/protocols/homedomain" rn="mn-name2">
   <ty>16</ty>
   <ri>/in-cse/csr-257922327</ri>
   <pi>/in-cse</pi>
   <ct>20201104T102845</ct>
   <lt>20201104T102845</lt>
   <acpi>/in-cse/acp-638681875</acpi>
   <poa>http://192.168.0.4:8181/</poa>
   <cb>//om2m.org/mn-cse2</cb>
   <csi>/mn-cse2</csi>
   <rr>true</rr>
</m2m:csr>

]
[INFO] - org.eclipse.om2m.core.CSEInitializer
Successfully registered to in-cse
[INFO] - org.eclipse.om2m.core.remotecse.RemoteCseService
addRemoteCseAndPublish(cseId=/mn-cse2/csr-367783846, name=in-name)
[INFO] - org.eclipse.om2m.core.remotecse.RemoteCseService
post Event to inform about RemoteCSE creation (cseId=in-cse, cseName=in-name)
```

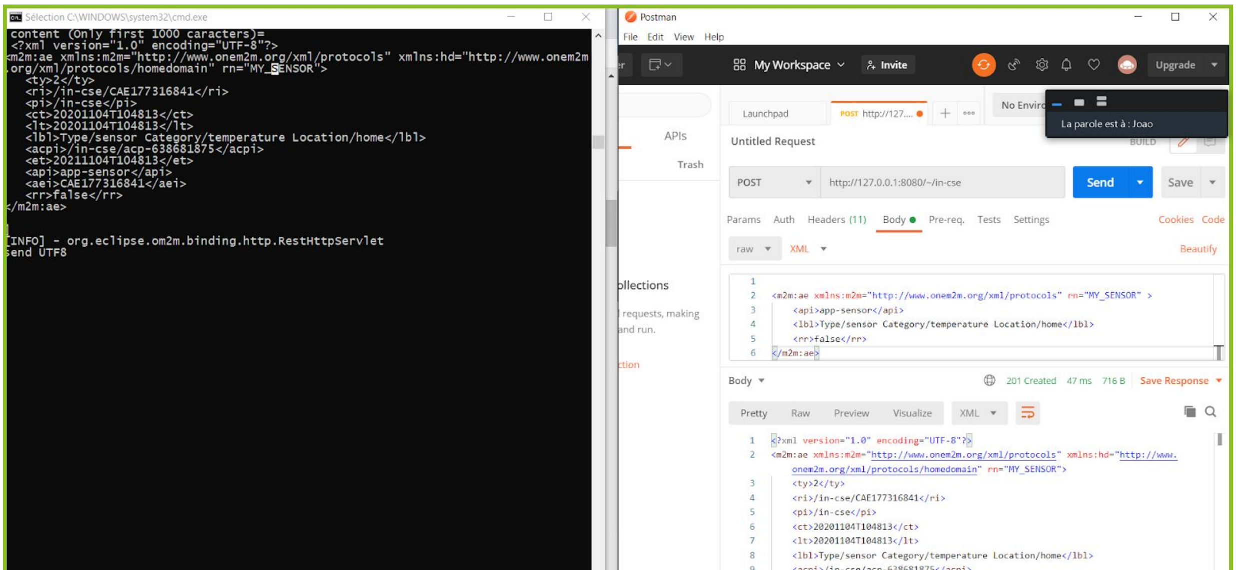
After these first configuration steps, that helped us to familiarize with the system, we installed the postman client and the REST API.

Then we followed the tutorial: https://wiki.eclipse.org/OM2M/one/REST_API that allows us to understand and use the different requests.

The scenario : Simulate several sensors and register an application that will monitor the new values pushed by the sensors using the sub/notification mechanism.

1- Create 3 AE

Creation of an AE:



Sélection C:\WINDOWS\system32\cmd.exe

```
 content (Only first 1000 caracters)=
 <?xml version="1.0" encoding="UTF-8"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m
.org/xml/protocols/homedomain" rn="MY_SENSOR">
   <ty>2</ty>
   <ri>/in-cse/CAE177316841</ri>
   <pi>/in-cse</pi>
   <ct>20201104T104813</ct>
   <lt>20201104T104813</lt>
   <lbl>Type/sensor Category/temperature Location/home</lbl>
   <acpi>/in-cse/acp-638681875</acpi>
   <et>20211104T104813</et>
   <api>app-sensor</api>
   <aei>CAE177316841</aei>
   <rr>false</rr>
</m2m:ae>

]
[INFO] - org.eclipse.om2m.binding.http.RestHttpServlet
send UTF8
```

POST http://127.0.0.1:8080/~/in-cse

raw XML

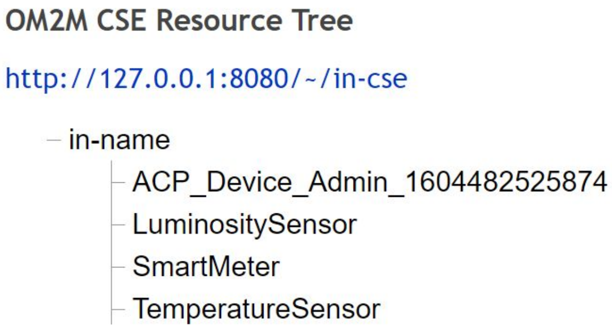
```
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="MY_SENSOR" >
    <api>app-sensor</api>
    <lbl>Type/sensor Category/temperature Location/home</lbl>
    <rr>false</rr>
</m2m:ae>
```

Body 201 Created 47 ms 716 B

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.
    onem2m.org/xml/protocols/homedomain" rn="MY_SENSOR">
    <ty>2</ty>
    <ri>/in-cse/CAE177316841</ri>
    <pi>/in-cse</pi>
    <ct>20201104T104813</ct>
    <lt>20201104T104813</lt>
    <lbl>Type/sensor Category/temperature Location/home</lbl>
```

As we can see on this image, we started by creating an AE named “MY_SENSOR” to see if the various steps written on the tutorial actually worked. After seeing the AE created and working properly, we created 3 AE as demanded.

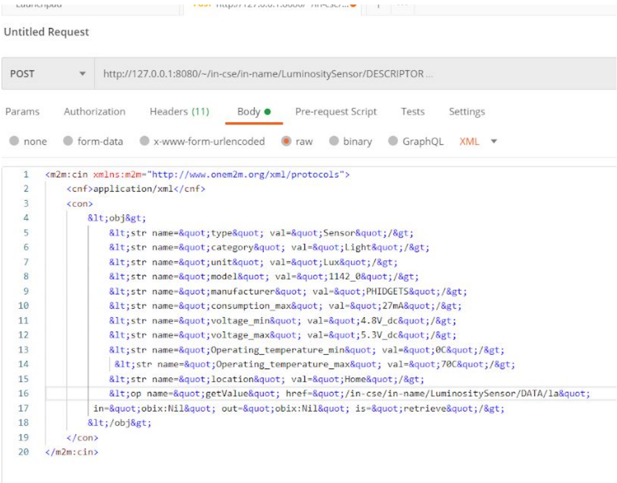


OM2M CSE Resource Tree

http://127.0.0.1:8080/~/in-cse

```
– in-name
    ACP_Device_Admin_1604482525874
    LuminositySensor
    SmartMeter
    TemperatureSensor
```

Then for each one of the sensors we created a DESCRIPTOR and a DATA container, using the following command :

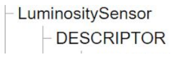


Untitled Request

POST http://127.0.0.1:8080/~/in-cse/in-name/LuminositySensor/DESCRIPTOR ...

Params
Authorization
Headers (11)
Body ●
Pre-request Script
Tests
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL XML

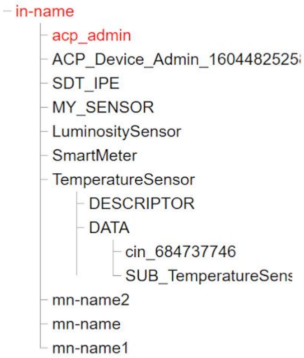
```
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
    <cnf>application/xml</cnf>
    <con>
        &lt;obj&gt;
            &lt;str name=&quot;type&quot; val=&quot;Sensor&quot;/&gt;
            &lt;str name=&quot;category&quot; val=&quot;Light&quot;/&gt;
            &lt;str name=&quot;unit&quot; val=&quot;Lux&quot;/&gt;
            &lt;str name=&quot;model&quot; val=&quot;1142_0&quot;/&gt;
            &lt;str name=&quot;manufacturer&quot; val=&quot;PHIDGETS&quot;/&gt;
            &lt;str name=&quot;consumption_max&quot; val=&quot;27mA&quot;/&gt;
            &lt;str name=&quot;voltage_min&quot; val=&quot;4.8V_dc&quot;/&gt;
            &lt;str name=&quot;voltage_max&quot; val=&quot;5.3V_dc&quot;/&gt;
            &lt;str name=&quot;Operating_temperature_min&quot; val=&quot;0C&quot;/&gt;
             &lt;str name=&quot;Operating_temperature_max&quot; val=&quot;70C&quot;/&gt;
            &lt;str name=&quot;location&quot; val=&quot;Home&quot;/&gt;
            &lt;op name=&quot;getValue&quot; href=&quot;/in-cse/in-name/LuminositySensor/DATA/la&quot;
         in=&quot;obix:Nil&quot; out=&quot;obix:Nil&quot; is=&quot;retrieve&quot;/&gt;
        &lt;/obj&gt;
    </con>
</m2m:cin>
```



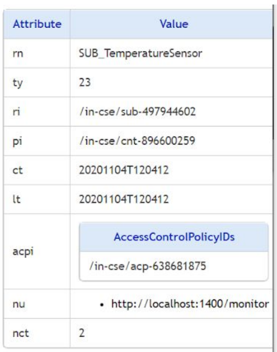
```
LuminositySensor
    DESCRIPTOR
```

The DESCRIPTOR and DATA containers were created using the configuration given on the Lab paper, as shown on the image above.

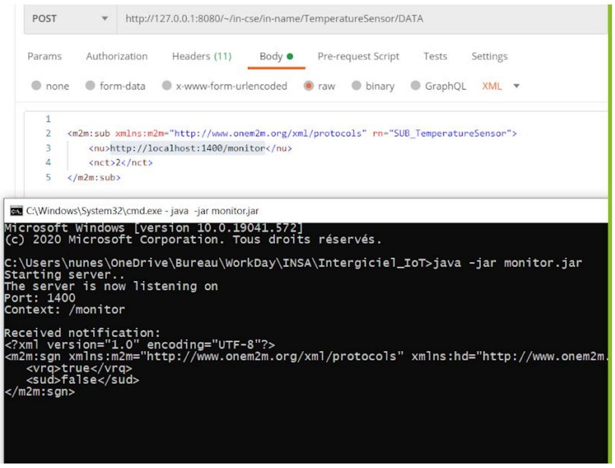
When each sensor had a DESCRIPTOR and DATA container, we started monitoring the various sensors by subscripting to each one. Here is an example :



```
– in-name
    acp_admin
    ACP_Device_Admin_1604482525
    SDT_IPE
    MY_SENSOR
    LuminositySensor
    SmartMeter
    TemperatureSensor
        DESCRIPTOR
        DATA
            cin_684737746
            SUB_TemperatureSens
    mn-name2
    mn-name
    mn-name1
```



Attribute	Value
rn	SUB_TemperatureSensor
ty	23
ri	/in-cse/sub-497944602
pi	/in-cse/cnt-896600259
ct	20201104T120412
lt	20201104T120412
acpi	AccessControlPolicyIDs /in-cse/acp-638681875
nu	• http://localhost:1400/monitor
nct	2



POST http://127.0.0.1:8080/~/in-cse/in-name/TemperatureSensor/DATA

Params
Authorization
Headers (11)
Body ●
Pre-request Script
Tests
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL XML

```

<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="SUB_TemperatureSensor">
    <nu>http://localhost:1400/monitor</nu>
    <nct>2</nct>
</m2m:sub>
```

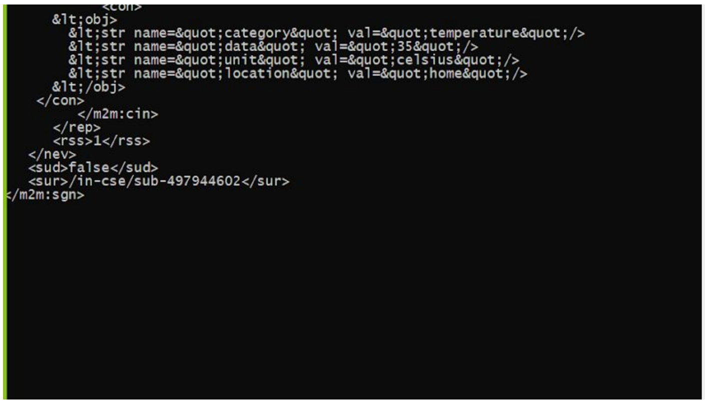
C:\Windows\System32\cmd.exe - java -jar monitor.jar

```
Microsoft Windows [version 10.0.19041.572]
(c) 2020 Microsoft Corporation. Tous droits réservés.

C:\Users\nunes\OneDrive\Bureau\WorkDay\INSA\Intergiciel_IoT>java -jar monitor.jar
Starting server..
The server is now listening on
Port: 1400
Context: /monitor

Received notification:
<?xml version="1.0" encoding="UTF-8"?>
<m2m:sgn xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m.
   <vrq>true</vrq>
   <sud>false</sud>
</m2m:sgn>
```

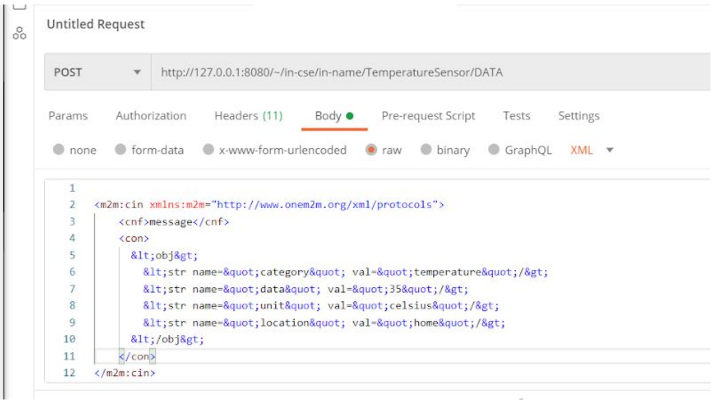
But the final goal of this Lab is to have a monitor AE redirecting every subscription message to the monitor app listening on port 1400. To do so we created a fourth AE, named monitor with two particular attributes :

Request reachability attribute -> rr
Point of access -> 127.0.0.1:1400
By creating an AE with this poa we are redirecting the subscription notification to the port 1400, and so we can be notified when new data is sent. Of course this only works if the Monitor AE has subscribed to each sensor DATA container.

As we can see in the following image, the system works perfectly. When a new data is sent by a sensor, the monitor application running on the computer gets a notification that has been routed by the monitor AE.



```
           <con>
     &lt;obj>
       &lt;str name=&quot;category&quot; val=&quot;temperature&quot;/>
       &lt;str name=&quot;data&quot; val=&quot;35&quot;/>
       &lt;str name=&quot;unit&quot; val=&quot;celsius&quot;/>
       &lt;str name=&quot;location&quot; val=&quot;home&quot;/>
     &lt;/obj>
   </con>
         </m2m:cin>
     </rep>
     <rss>1</rss>
  </nev>
  <sud>false</sud>
  <sur>/in-cse/sub-497944602</sur>
</m2m:sgn>
```



Untitled Request

POST http://127.0.0.1:8080/~/in-cse/in-name/TemperatureSensor/DATA

```

<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
    <cnf>message</cnf>
    <con>
      &lt;obj&gt;
        &lt;str name=&quot;category&quot; val=&quot;temperature&quot;/&gt;
        &lt;str name=&quot;data&quot; val=&quot;35&quot;/&gt;
        &lt;str name=&quot;unit&quot; val=&quot;celsius&quot;/&gt;
        &lt;str name=&quot;location&quot; val=&quot;home&quot;/&gt;
      &lt;/obj&gt;
    </con>
</m2m:cin>
```