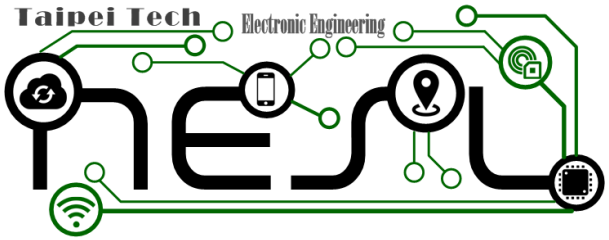
智慧整合感控系統概論
Introduction to Cyber-Physical Systems

LAB：OM2M + Postman

國立臺北科技大學電子工程系
授課教師：李昭賢 副教授
電子郵件：chlee@ntut.edu.tw
校內分機：2288

行動寬頻尖端技術跨校教學聯盟



http://www.cc.ntut.edu.tw/~chlee/

學習目標

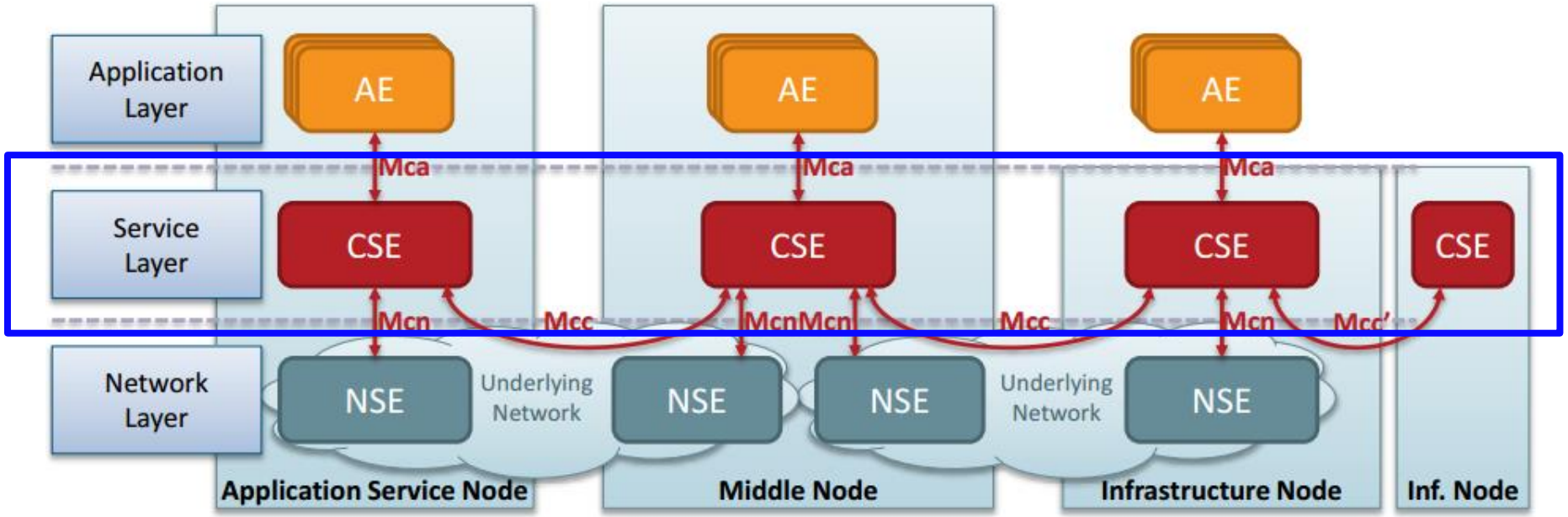
1. OM2M介紹
2. 利用Postman存取OM2M

OM2M Overview

- The Eclipse OM2M project, initiated by LAAS-CNRS, is an open source implementation of oneM2M and smartM2M standard.
  - It provides a horizontal Service Common Entity (CSE) that can be deployed in an M2M server, a gateway, or a device.
  - Each CSE provides Application Enablement, Security, Triggering, Notification, Persistency, Device Interworking, Device Management, etc.

OM2M Overview

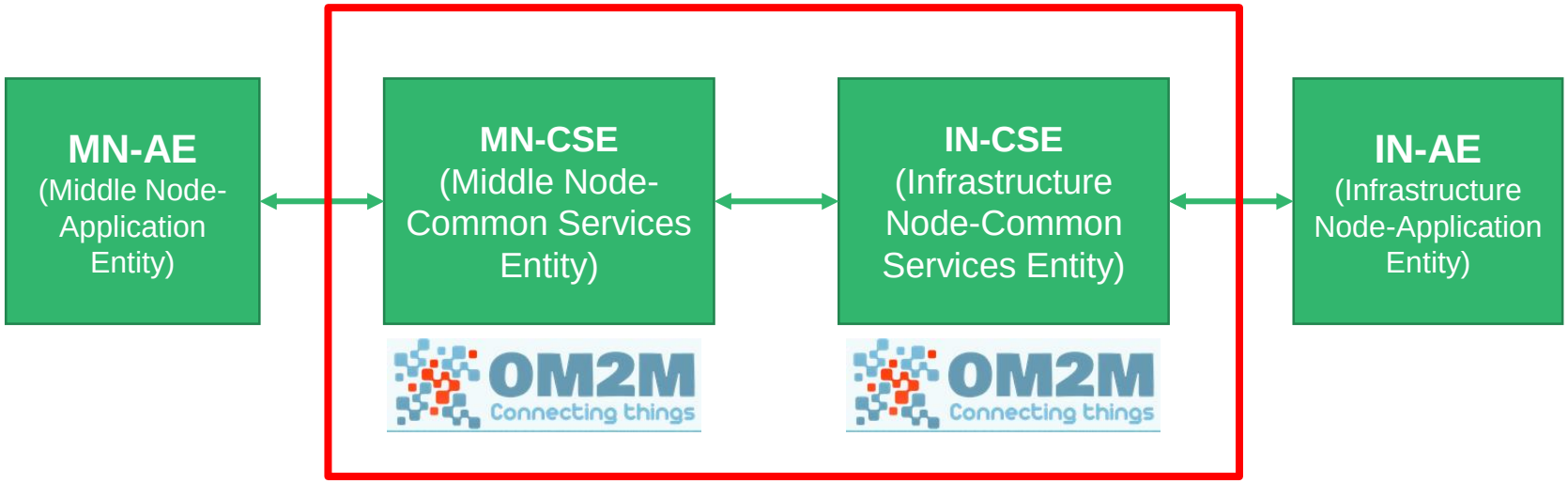
OneM2M



AE: Application Entity
CSE: Common Service Entity
NSE: Network Service Entity

OM2M Overview

OM2M



OM2M Overview

- Exposes a RESTful API providing primitive procedures
  - machines authentication
  - resources discovery
  - applications registration
  - containers management
  - synchronous and asynchronous communications
  - access rights authorization
  - groups organization
  - re-targeting

OM2M Overview

- OM2M is a Java implementation running on top of an OSGi Equinox runtime.
  - Making it highly extensible via plugins.
- OM2M is built as an Eclipse product using Maven and Tycho.
- Each plugin offers specific functionalities, and can be remotely installed, started, stopped, updated, and uninstalled without requiring a reboot.

OM2M VM

- Virtual Machine (VM) image download
  - https://goo.gl/wH2R7M
  - Password: iotclass

- This VM integrates Postman & NodeRED.
  - If you want to install by yourself, please refer to the following link.
    - https://wiki.eclipse.org/OM2M/one#Introduction

OM2M in-cse & mn-cse on different VM

- Set two VMs on **Virtual Box** ,one is for in-cse the other for mn-cse.

OM2M VM Network Setting

- Select Bridge options in network settings.

- If you want to use two VMs on different computers
  - The two different computers must be located in the same router.

Set the IP of M2M IN-CSE

- Open VM and find your VM IP :
  - $ ifconfig

```
iotclass@iotclass:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:96:3a:51
          inet addr:192.168.0.134  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::8409:a79c:75d4:88b2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:450803 errors:0 dropped:1 overruns:0 frame:0
          TX packets:106583 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:492955701 (492.9 MB)  TX bytes:6485711 (6.4 MB)
```

- IN-CSE configuration directory
  - $cd org.eclipse.om2m/org.eclipse.om2m.site.in-cse/target/products/in-cse/linux/gtk/x86_64/configuration

Set the IP of M2M IN-CSE

- Set the IP in the config.ini : $ nano config.ini
  - org.eclipse.om2m.cseBaseAddress = “ IP ”

```
 GNU nano 2.5.3                File: config.ini

#This configuration file was written by: org.eclipse.equinox.internal.framework$
#Fri Jul 01 16:56:47 CST 2016
log4j.configuration=file\:./log4j.configuration
org.eclipse.equinox.http.jetty.http.port=8080
org.eclipse.om2m.dbReset=true
org.eclipse.om2m.cseBaseContext=/
org.eclipse.om2m.globalContext=
osgi.bundles=reference\:file\:javax.servlet_3.1.0.v20140303-1611.jar@4,referenc$
org.eclipse.om2m.cseBaseProtocol.default=http
org.eclipse.om2m.cseBaseName=in-name
org.eclipse.om2m.cseBaseAddress=192.168.0.134
eclipse.p2.profile=DefaultProfile
org.eclipse.om2m.dbUrl=jdbc\:h2\:./database/indb
```

Set the IP of M2M MN-CSE

- Open the other VM and find your VM IP :
  - $ ifconfig

```
iotclass@iotclass:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:fa:78:c0
          inet addr:192.168.0.175  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::e9cc:f3ca:dc86:a3da/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6238 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1597 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6138909 (6.1 MB)  TX bytes:128668 (128.6 KB)
```

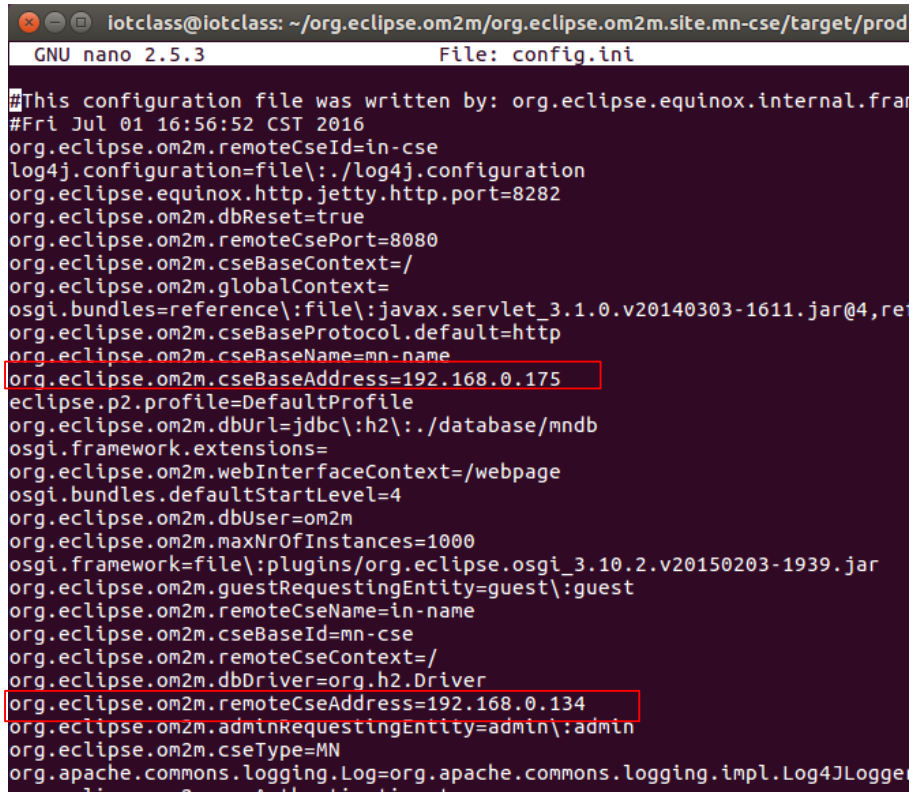
- MN-CSE configuration directory
  - $cd org.eclipse.om2m/org.eclipse.om2m.site.mn-cse/target/products/mn-cse/linux/gtk/x86_64/configuration

Set the IP of M2M MN-CSE

- Set the IP in the config.ini : $ nano config.ini
  - org.eclipse.om2m.cseBaseAddress = " IP *MN-CSE* "
  - org.eclipse.om2m.remoteCseAdderss = " IP *IN-CSE* "

The IP of MN-CSE

The IP of IN-CSE (the other VM)



Start the M2M IN-CSE

- Start the OM2M IN-CSE
  - IN-CSE product directory :
    - org.eclipse.om2m/org.eclipse.om2m.site.in-cse/target/products/in-cse/linux/gtk/x86_64
  - Open a terminal, go to the product directory and input the command:
    - $ sh start.sh
  - After starting it successfully, you will see “CSE Started”

OM2M Resource Tree Visualizer Tool

- URL: http://localhost:8080/webpage
  - User name: admin
  - User password: admin

OM2M Resource Tree Visualizer Tool

- You can see the "in-cse" cseBase sub-resources and attributes.

Start the M2M MN-CSE

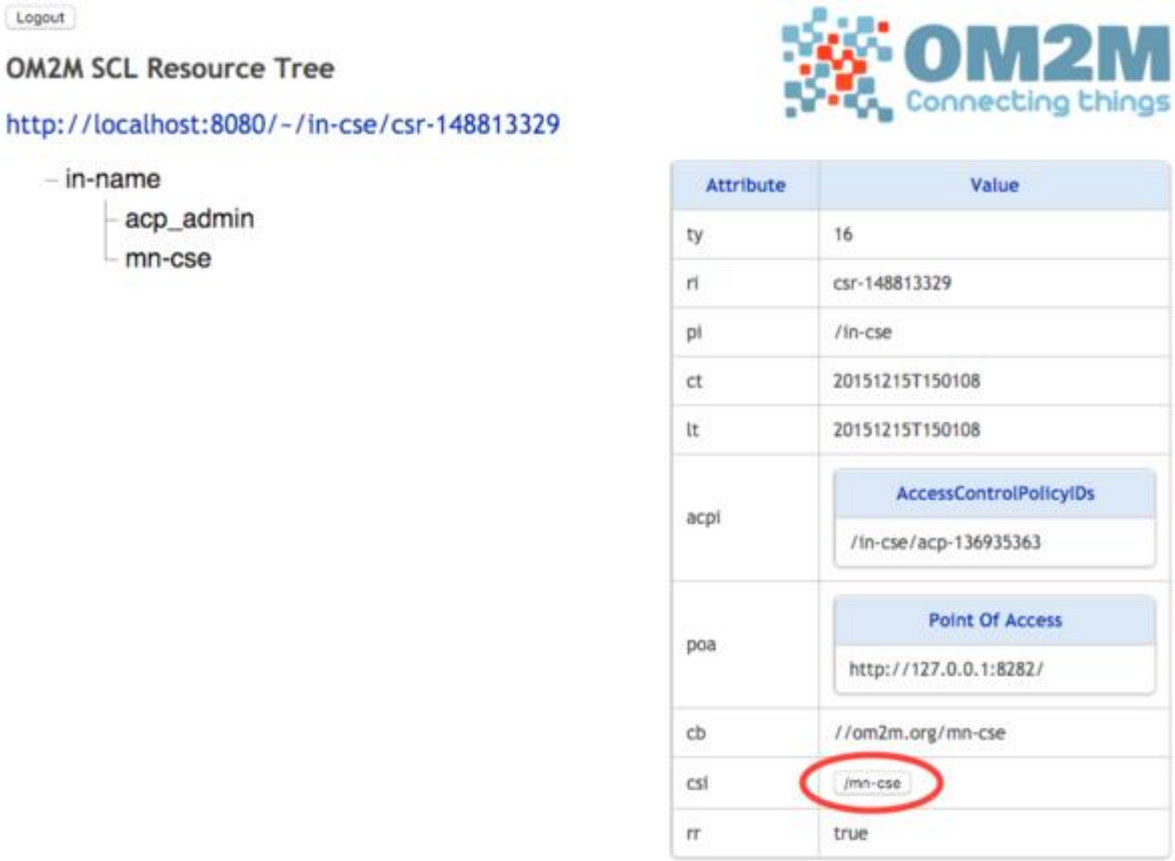
- Start the OM2M MN-CSE
  - MN-CSE product directory :
    - org.eclipse.om2m/org.eclipse.om2m.site.mn-cse/target/products/mn-cse/linux/gtk/x86_64
  - Open another terminal, go to the product directory and input the command:
    - $ sh start.sh
  - After starting it successfully, you will see “CSE Started”

OM2M Resource Tree Visualizer Tool

- After a successful authentication, the “mn-cse” resource is added to the in-cse resource tree, and respectively the “in-cse” resource is added to the mn-cse resource tree.
- You can now access the registered MN-CSE resource from the IN-CSE web interface under the “/in-cse/in-name/mn-cse” uri.

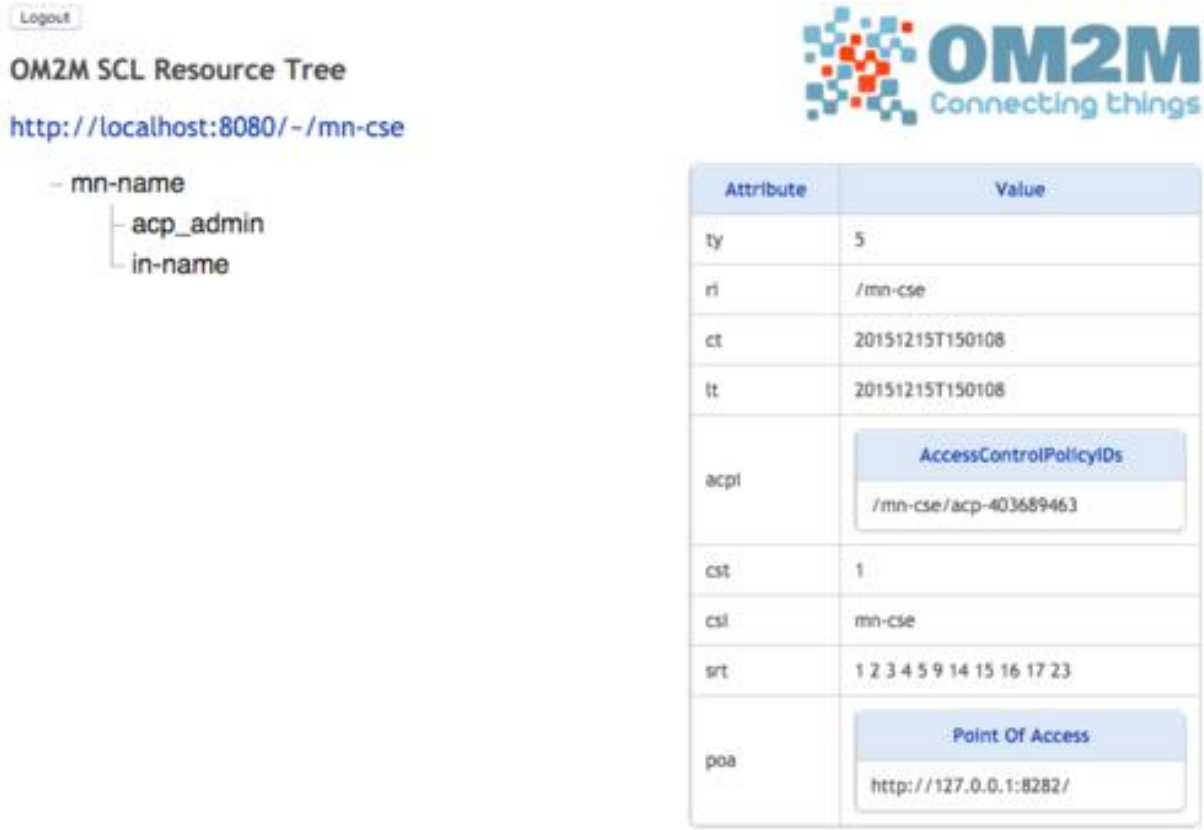
OM2M Resource Tree Visualizer Tool

- You notice the existence of one authenticated MN-CSE with id "mn-cse".



OM2M Resource Tree Visualizer Tool

- Click on the "mn-cse" resource to display remote MN-CSE sub-resources and attributes.



練習一：Retrieve a resource

- Postman
  - X-M2M-Origin: the authentication is handled using a specific oneM2M header field.

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse
Method	GET
Header	X-M2M-Origin : admin:admin
Body	(empty)

練習一：Retrieve a resource

Response

Field	Value
Status	200 OK
Body	<?xml version="1.0" encoding="UTF-8"?> <m2m:cb xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="mn-name"> <ty>5</ty> <ri>/mn-cse</ri> <ct>20160628T124737</ct> <lt>20160628T124737</lt> <acpi>/mn-cse/acp-832322075</acpi> <cst>1</cst> <csi>mn-cse</csi> <srt>1 2 3 4 5 9 14 15 16 17 23</srt> <poa>http://127.0.0.1:8282/</poa> </m2m:cb>

練習一：Retrieve a resource

- The feedback XML format is defined in oneM2M Specification: TS-0004 Service Layer Core Protocol.
  - http://www.onem2m.org/technical/published-documents
    - 對照xsd_v2_7_0_long & xsd_v2_7_0_short資料夾內的各個XML Schema (.xsd)即可得知全名與縮寫。

查核一：Retrieve a resource

- 請分別顯示Postman與OM2M Resource Tree Visualizer Tool的資料。
  - 請注意，對照後資料應該一致。

練習二：Create a "MY_SENSOR" Application

- Postman: create an AE
  - ty=2 for AE (Application Entity)
  - X-M2M-NM: the name of the resource

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse
Method	POST
Header	X-M2M-Origin: admin:admin Content-Type: application/xml;ty=2 X-M2M-NM: MY_SENSOR
Body	<om2m:ae xmlns:om2m="http://www.onem2m.org/xml/protocols"> <api>app-sensor</api> <lbl>Type/sensor Category/temperature Location/home</lbl> <rr>false</rr> </om2m:ae>

練習二：Create a "MY_SENSOR" Application

- The header parameters, e.g., ty and X-M2M-NM, are defined in oneM2M Specification: TS-0009 HTTP Protocol Binding.
  - http://www.onem2m.org/technical/published-documents
    - The value of “ty” is also defined in TS-0004 Service Layer Core Protocol.

練習二：Create a "MY_SENSOR" Application

- Response
  - get Status 201 Created

OM2M CSE Resource Tree

http://127.0.0.1:8282/~/mn-cse

- mn-name
  - acp_admin
  - acpae-344339622
  - MY_SENSOR
  - in-name

Attribute	Value
ty	2
ri	/mn-cse/CAE344339622
pi	/mn-cse
ct	20160630T150207
lt	20160630T150207
lbl	Type/sensor Category/temperature Location/home
acpi	AccessControlPolicyIDs: /mn-cse/acp-146439930
et	20170630T150207
api	app-sensor
aei	CAE344339622
rr	false

練習二：Create a "MY_SENSOR" Application

- Postman : create a DESCRIPTOR Container
  - ty=3 for Container

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse/mn-name/MY_SENSOR
Method	POST
Header	X-M2M-Origin: admin:admin Content-Type: application/xml;ty=3 X-M2M-NM: DESCRIPTOR
Body	<om2m:cnt xmlns:om2m="http://www.onem2m.org/xml/protocols"> </om2m:cnt>

練習二：Create a "MY_SENSOR" Application

Response

OM2M CSE Resource Tree

http://127.0.0.1:8282/~/mn-cse/CAE344339622

- mn-name
  - acp_admin
  - acpae-344339622
  - MY_SENSOR
    - DESCRIPTOR
  - in-name

Attribute	Value
ty	3
ri	/mn-cse/cnt-393381338
pi	/mn-cse/CAE344339622
ct	20160630T150621
lt	20160630T150621
acpi	AccessControlPolicyIDs: /mn-cse/acp-146439930
et	20170630T150621
st	0
mni	1000
mbs	10000
mia	0
cni	0
cbs	0
ol	/mn-cse/mn-name/MY_SENSOR /DESCRIPTOR/ol
la	/mn-cse/mn-name/MY_SENSOR /DESCRIPTOR/la

練習二：Create a "MY_SENSOR" Application

- Postman: create a DESCRIPTOR ContentInstance
  - ty=4 for ContentInstance

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse/mn-name/MY_SENSOR/DESCRIPTOR
Method	POST
Header	X-M2M-Origin: admin:admin Content-Type: application/xml;ty=4
Body	<om2m:cin xmlns:om2m="http://www.onem2m.org/xml/protocols"> <cnf>message</cnf> <con> <obj> <str name="type" val="Temperature_Sensor"/> <str name="location" val="Home"/> <str name="appId" val="MY_SENSOR"/> <op name="getValue" href="/mn-cse/mn-name/MY_SENSOR/DATA/la" in="obix:Nil" out="obix:Nil" is="retrieve"/> </obj> </con> </om2m:cin>

練習二：Create a "MY_SENSOR" Application

Response

OM2M CSE Resource Tree

http://127.0.0.1:8282/~/mn-cse/cin-533260925

- mn-name
  - acp_admin
  - acpae-344339622
  - MY_SENSOR
    - DESCRIPTOR
      - cin_533260925
  - in-name

Attribute	Value
ty	4
ri	/mn-cse/cin-50126502
pi	/mn-cse/cnt-113144776
ct	20160808T225649
lt	20160808T225649
st	0
cnf	message
cs	312
con	

Attribute	Value
type	Temperature_Sensor
location	Home
appId	MY_SENSOR
getValue	/mn-cse/mn-name/MY_SENSOR/DATA/la

練習二：Create a "MY_SENSOR" Application

- Postman: create a DATA Container

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse/mn-name/MY_SENSOR
Method	POST
Header	X-M2M-Origin: admin:admin Content-Type: application/xml;ty=3 X-M2M-NM: DATA
Body	<om2m:cnt xmlns:om2m="http://www.onem2m.org/xml/protocols"> </om2m:cnt>

練習二：Create a "MY_SENSOR" Application

Response

OM2M CSE Resource Tree

http://127.0.0.1:8282/~/mn-cse/cnt-474692093

- mn-name
  - acp_admin
  - acpae-344339622
  - MY_SENSOR
    - DESCRIPTOR
    - DATA
  - in-name

Attribute	Value
ty	3
ri	/mn-cse/cnt-474692093
pi	/mn-cse/CAE344339622
ct	20160630T150935
lt	20160630T150935
acpi	AccessControlPolicyIDs: /mn-cse/acp-146439930
et	20170630T150935
st	0
mni	1000
mbs	10000
mia	0
cni	0
cbs	0
ol	/mn-cse/mn-name/MY_SENSOR /DATA/ol
la	/mn-cse/mn-name/MY_SENSOR /DATA/la

練習二：Create a "MY_SENSOR" Application

- Postman: create a DATA ContentInstance

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse/mn-name/MY_SENSOR/DATA
Method	POST
Header	X-M2M-Origin: admin:admin Content-Type: application/xml;ty=4
Body	<om2m:cin xmlns:om2m="http://www.onem2m.org/xml/protocols"> <cnf>message</cnf> <con> <obj> <str name="appId" val="MY_SENSOR"/> <str name="category" val="temperature "/> <int name="data" val="27"/> <int name="unit" val="celsius"/> </obj> </con> </om2m:cin>

練習二：Create a "MY_SENSOR" Application

Response

OM2M CSE Resource Tree

http://127.0.0.1:8282/~/mn-cse/cin-959617898

- mn-name
  - acp_admin
  - acpae-344339622
  - MY_SENSOR
    - DESCRIPTOR
    - DATA
      - cin_959617898
  - in-name

Attribute	Value
ty	4
ri	/mn-cse/cin-252851262
pi	/mn-cse/cnt-485916153
ct	20160808T225745
lt	20160808T225745
st	0
cnf	message
cs	202
con	

Attribute	Value
appId	MY_SENSOR
category	temperature
data	777
unit	celsius

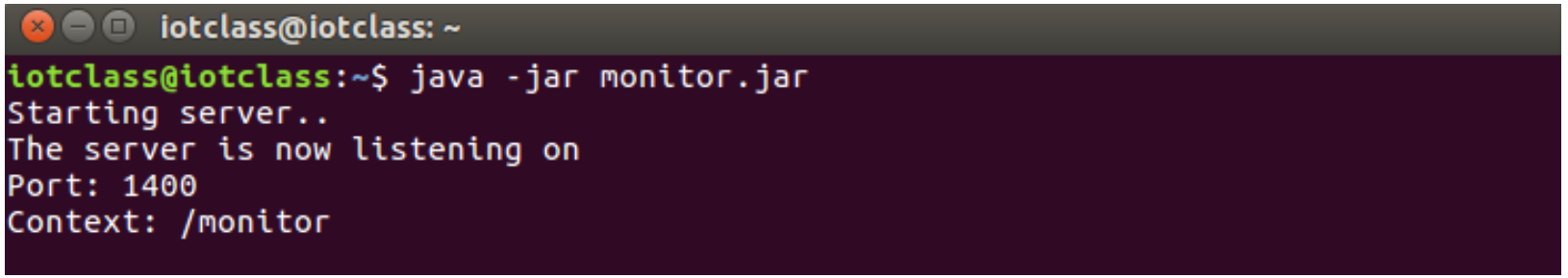
查核二：Create a "MY_SENSOR" Application

- 請利用OM2M Resource Tree Visualizer Tool顯示練習二的結果。

練習三：Subscribe to Data

- Monitor is a Web Application that listens for HTTP Post requests at port=1400 and context=/monitor.
  1. Open a terminal window.
  2. Move to monitor folder : /home/iotclass
  3. Start the Monitor server using the following command:

     $ java –jar monitor.jar



練習三：Subscribe to Data

- Postman: subscribe to MY_SENSOR Data
  - ty=23 for Subscription
  - nu: indicates the subscription representation, i.e., the monitor server (http://127.0.0.1:1400/monitor)

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse/mn-name/MY_SENSOR/DATA
Method	POST
Header	X-M2M-Origin: admin:admin Content-Type: application/xml;ty=23 X-M2M-NM: SUB_MY_SENSOR
Body	<m2m:sub xmlns:m2m="http://www.onem2m.org/xml/protocols"> <nu>http://localhost:1400/monitor</nu> <nct>2</nct> </m2m:sub>

練習三：Subscribe to Data

Response

OM2M CSE Resource Tree

http://127.0.0.1:8282/~/mn-cse/sub-647980870

- mn-name
  - acp_admin
  - acpae-344339622
  - MY_SENSOR
    - DESCRIPTOR
    - DATA
      - cin_959617898
      - SUB_MY_SENSOR
  - in-name

Attribute	Value
ty	23
ri	/mn-cse/sub-647980870
pi	/mn-cse/cnt-474692093
ct	20160630T151240
lt	20160630T151240
acpi	AccessControlPolicyIDs: /mn-cse/acp-146439930
nu	http://localhost:1400/monitor
nct	2

練習三：Subscribe to Data

- Postman: update MY_SENSOR Data
  - 請仿效練習二，再次新增一筆DATA ContentInstance
  - 記得要修改其中DATA數值，否則分辨不清楚

Field	Value
URL	http://127.0.0.1:8282/~/mn-cse/mn-name/MY_SENSOR/DATA
Method	POST
Header	X-M2M-Origin: admin:admin Content-Type: application/xml;ty=4
Body	<om2m:cin xmlns:om2m="http://www.onem2m.org/xml/protocols"> <cnf>message</cnf> <con> <obj> <str name="appId" val="MY_SENSOR"/> <str name="category" val="temperature "/> <int name="data" val="27"/> <int name="unit" val="celsius"/> </obj> </con> </om2m:cin>

練習三：Subscribe to Data

- 回到執行Monitor的Terminal，將會看到接收到資料更新

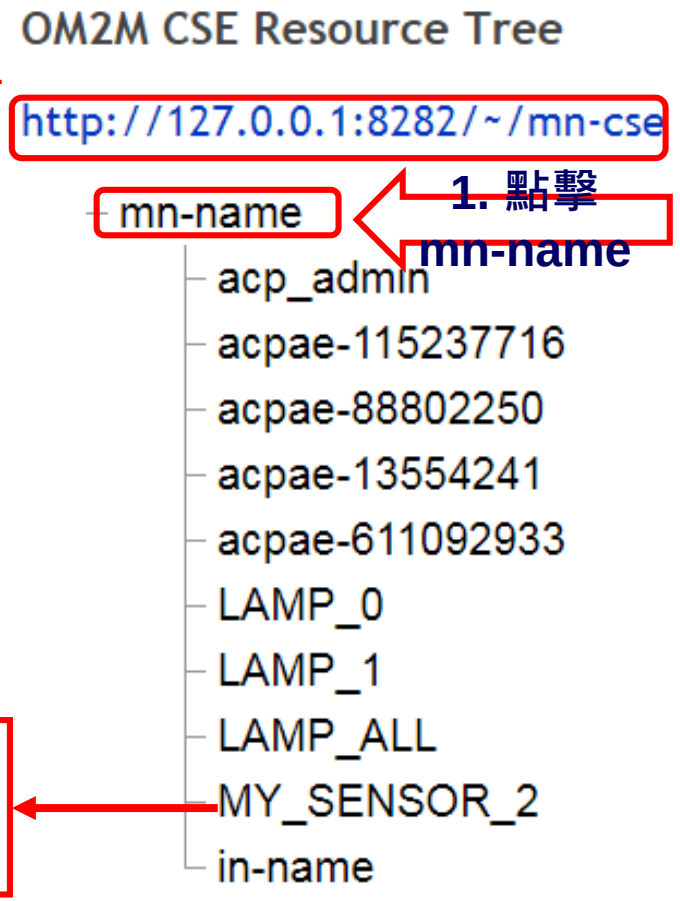
```
Received notification:
<?xml version="1.0" encoding="UTF-8"?>
<m2m:sgn xmlns:m2m="http://www.onem2m.org/xml/protocols">
   <nev>
      <rep rn="cin_693032208">
         <ty>4</ty>
         <ri>/mn-cse/cin-693032208</ri>
         <pi>/mn-cse/cnt-340979605</pi>
         <ct>20160809T132308</ct>
         <lt>20160809T132308</lt>
         <st>0</st>
         <cnf>message</cnf>
         <cs>201</cs>
         <con>
      &lt;obj>
        &lt;str name=&quot;appId&quot; val=&quot;MY_SENSOR&quot;/>
        &lt;str name=&quot;category&quot; val=&quot;temperature &quot;/>
        &lt;int name=&quot;data&quot; val=&quot;33&quot;/>
        &lt;int name=&quot;unit&quot; val=&quot;celsius&quot;/>
      &lt;/obj>
    </con>
      </rep>
      <rss>1</rss>
   </nev>
   <sud>false</sud>
   <sur>/mn-cse/mn-name/MY_SENSOR/DATA/SUB_MY_SENSOR</sur>
</m2m:sgn>
```

查核三：Subscribe to Data

- 請顯示Monitor接收資料更新的畫面。

練習四：Delete

- 查詢URL
  - 方法一：使用 資源名稱
    1. 點擊mn-name可看到CSEBase對應之URL
    2. 從CSEBase開始，就可由Resource Tree推每個項目的URL

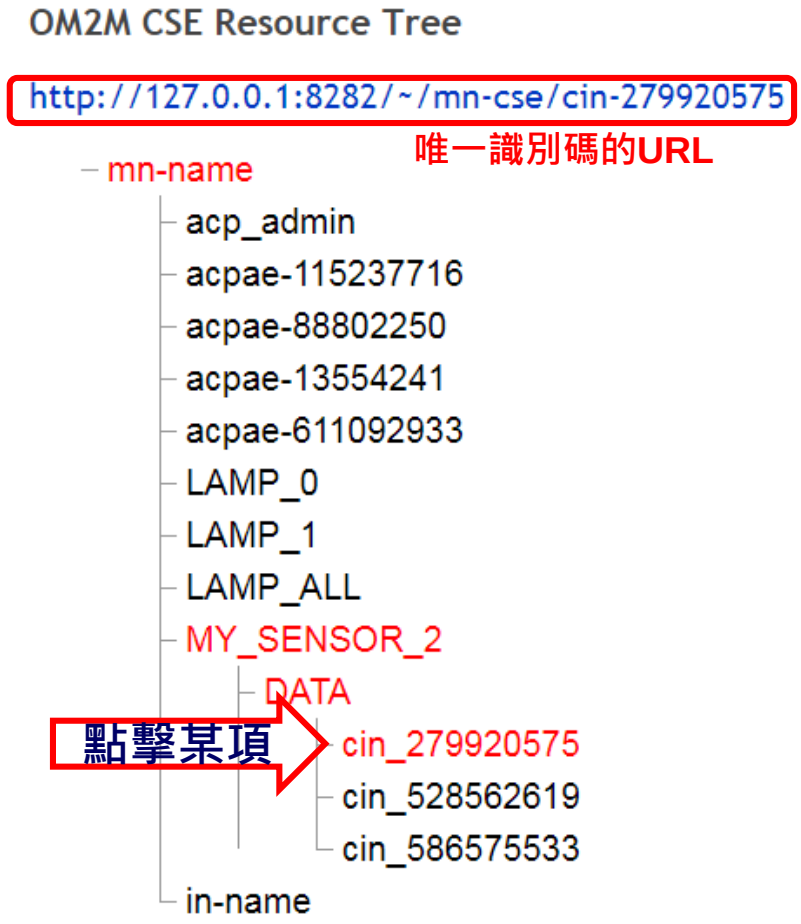


例如：想取得mn-name下的MY_SENSOR_2的url

http://localhost:8282/~/mn-cse/mn-name/MY_SENSOR_2

練習四：Delete

- 查詢URL
  - 方法二：使用 唯一識別碼UID
    - 直接點選某項資源取得其URL



練習四：Delete

- **使用 資源名稱 指定URL**
  - **Application Url (**MY_SENSOR_2**)**
    - http://localhost:8282/~/mn-cse/mn-name/MY_SENSOR_2
  - **Container Url (**MY_SENSOR_2/DATA**)**
    - http://localhost:8282/~/mn-cse/mn-name/MY_SENSOR_2/DATA
  - **ContentInstance Url (**MY_SENSOR_2/DATA/cin_279920575**)**
    - http://localhost:8282/~/mn-cse/mn-name/MY_SENSOR_2/DATA/cin_279920575

- **使用 唯一識別碼UID 指定URL**
  - **Application Url (**MY_SENSOR_2**)**
    - http://localhost:8282/~/mn-cse/CAE611092933
  - **Container Url (**MY_SENSOR_2/DATA**)**
    - http://localhost:8282/~/mn-cse/cnt-881591379
  - **ContentInstance Url (**MY_SENSOR_2/DATA/cin_279920575**)**
    - http://localhost:8282/~/mn-cse/cin-279920575

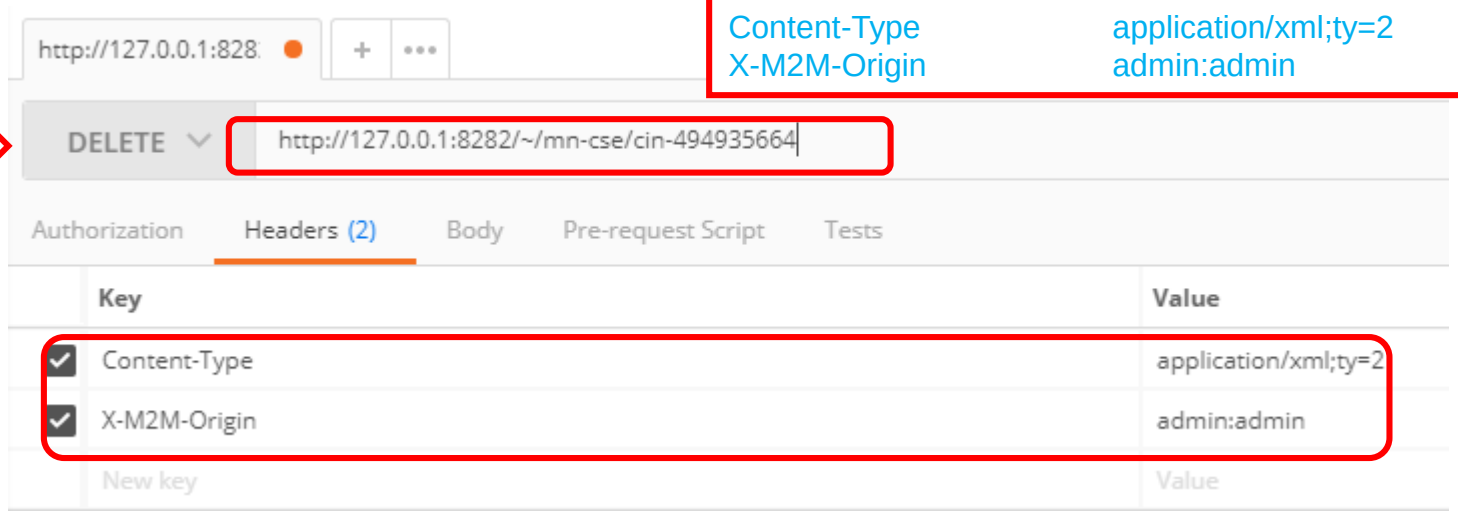
練習四：Delete

- Postman: delete ANYTHING
  1. 選擇DELETE
  2. 輸入URL (使用前面方法得知預刪除的URL)
  3. 設定Header

Url
該資源的url
Headers

Content-Type	application/xml;ty=2
X-M2M-Origin	admin:admin

選擇 DELETE



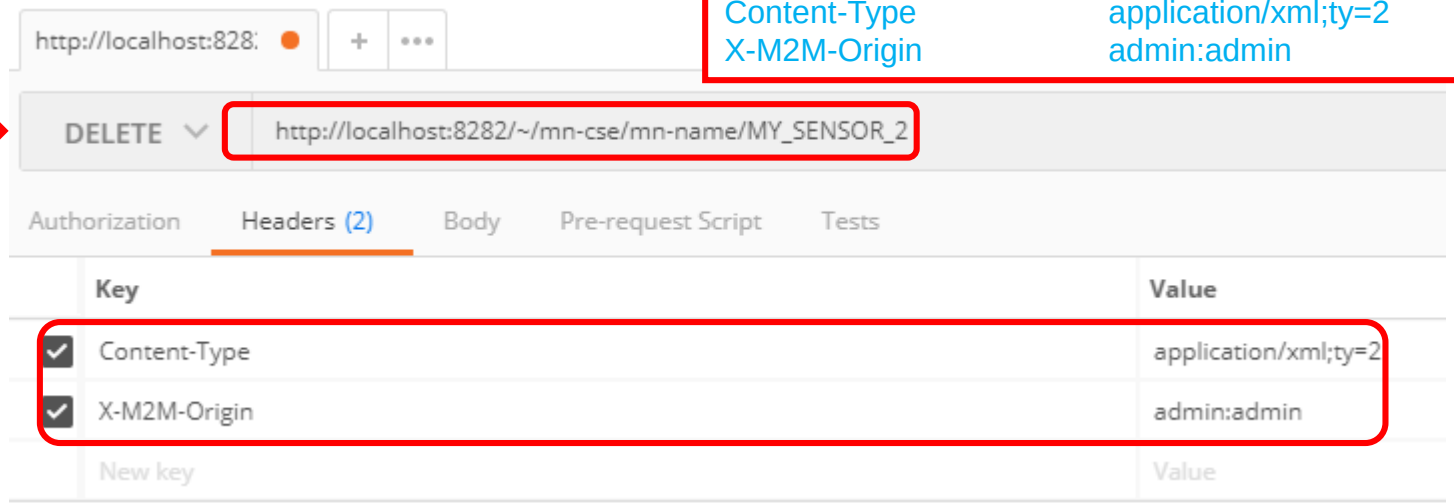
練習四：Delete

- Postman: delete MY_SENSOR_2
  1. 選擇DELETE
  2. 輸入MY_SENSOR_2's URL
  3. 設定Header

Application Url (MY_SENSOR_2**)**
http://localhost:8282/~/mn-cse/mn-name/MY_SENSOR_2
Headers

Content-Type	application/xml;ty=2
X-M2M-Origin	admin:admin

選擇 DELETE



總結

- The aim of OM2M is to facilitate the deployment of vertical applications and heterogeneous devices.

- This LAB shows how to use REST API for applications registration, containers management, and data access.
  - More details: https://wiki.eclipse.org/OM2M/one/REST_API