

# International Institute of Information Technology

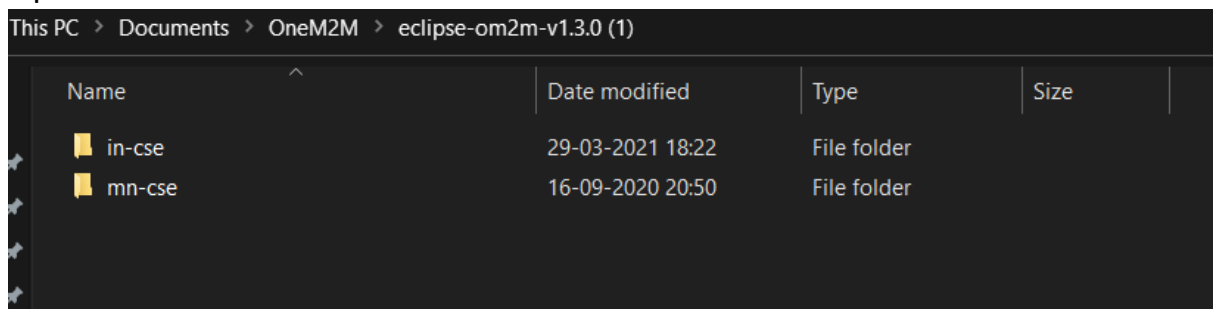
## Research Teaser Internship '21

### - Internet of Things

**Overview:** In this lab session, you will be creating a resource tree in OM2M using Postman. You are going to create an AE or Application Entity named test\_AE then create a CNT or Container named test\_container inside the AE. Inside test\_container you will be creating a container instance. Then fetch the latest data inside test\_container.

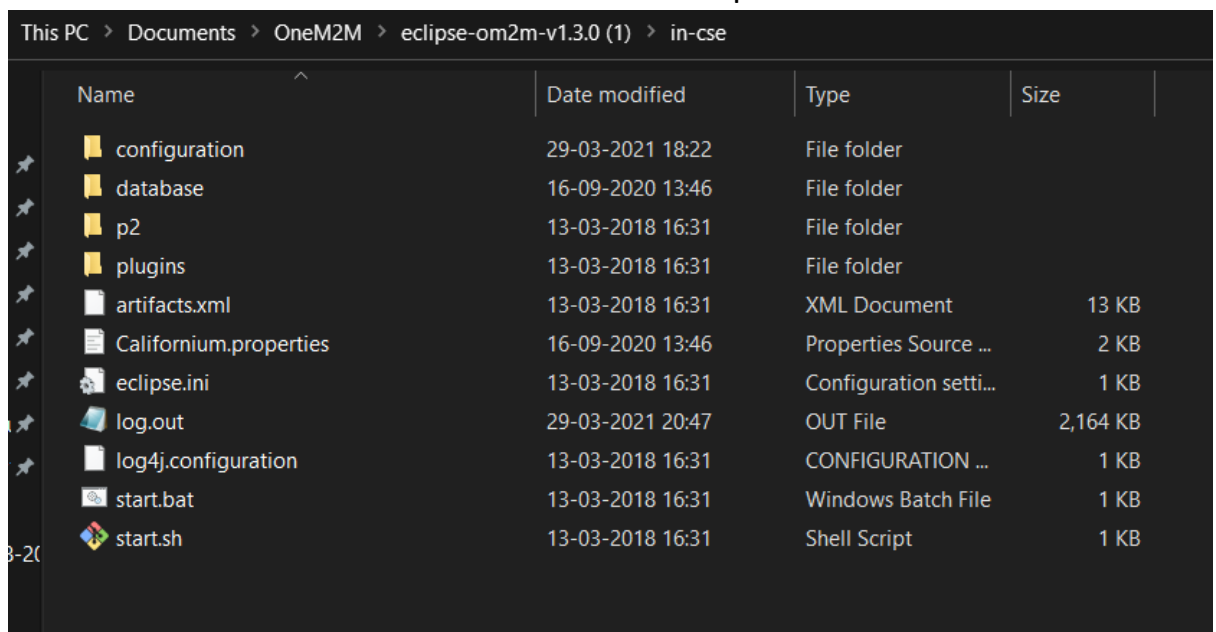
#### Part 1: Start the OM2M

1. Open the OM2M folder



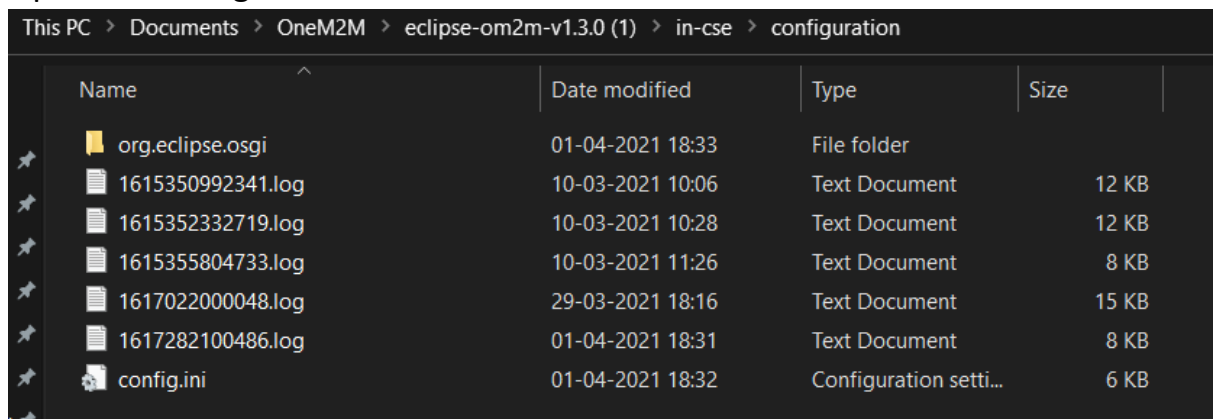
Name	Date modified	Type	Size
in-cse	29-03-2021 18:22	File folder	
mn-cse	16-09-2020 20:50	File folder	

2. There are two folders in-cse and mn-cse. Now open the folder in-cse



Name	Date modified	Type	Size
configuration	29-03-2021 18:22	File folder	
database	16-09-2020 13:46	File folder	
p2	13-03-2018 16:31	File folder	
plugins	13-03-2018 16:31	File folder	
artifacts.xml	13-03-2018 16:31	XML Document	13 KB
Californium.properties	16-09-2020 13:46	Properties Source ...	2 KB
eclipse.ini	13-03-2018 16:31	Configuration setti...	1 KB
log.out	29-03-2021 20:47	OUT File	2,164 KB
log4j.configuration	13-03-2018 16:31	CONFIGURATION ...	1 KB
start.bat	13-03-2018 16:31	Windows Batch File	1 KB
start.sh	13-03-2018 16:31	Shell Script	1 KB

### 3. Open the configuration folder



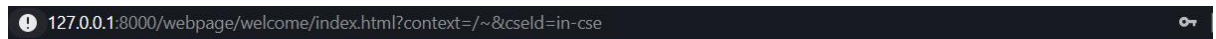
### 4. Open the config.ini and here you can change the port where you want host the OM2M by changing org.eclipse.equinox.http.jetty.http.port to your desired port.

```
#This configuration file was written by: org.eclipse.equinox.internal.frameworkadmin.equinox.EquinoxFwConfigFileParser
#Tue Mar 13 06:01:10 CDT 2018
log4j.configuration=file\:./log4j.configuration
org.eclipse.equinox.http.jetty.http.port=8000
org.eclipse.om2m.dbReset=false
org.eclipse.om2m.cseBaseContext=/  

```

### 5. Now open the start.bat for windows and start.sh for ubuntu/linux

### 6. Now open your browser and put in the url 127.0.0.1:<port number>/webpage For example in my case I set the port number as 8000 so in my browser I opened 127.0.0.1:8000/webpage



username:

password:

Login

### 7. The default username and password are admin and admin, respectively. Put username and password and then click login.

Logout

OM2M CSE Resource Tree

http://127.0.0.1:8000/~in-cse

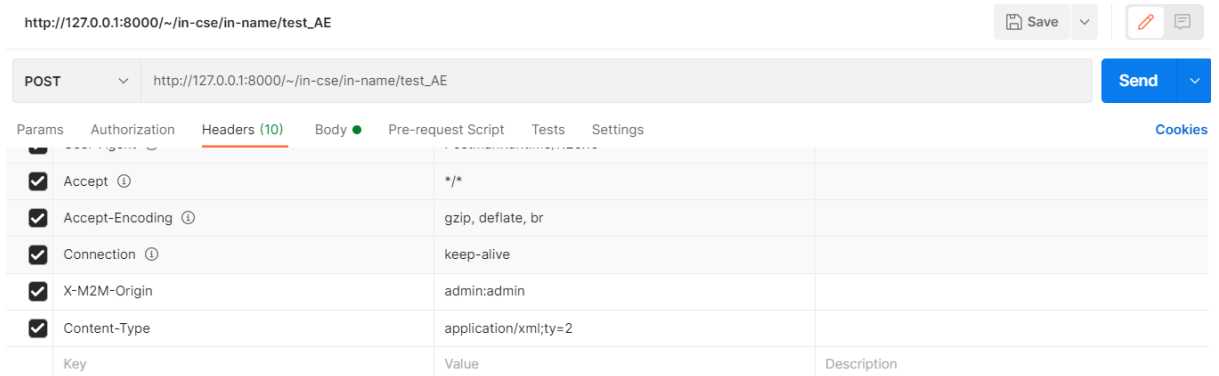
- in-name
  - acp\_admin



Attribute	Value
rn	in-name
ty	5
ri	/in-cse
ct	20200916T134625
lt	20200916T134625
acpi	<div>AccessControlPolicyIDs</div> <div>/in-cse/aco-473755009</div>

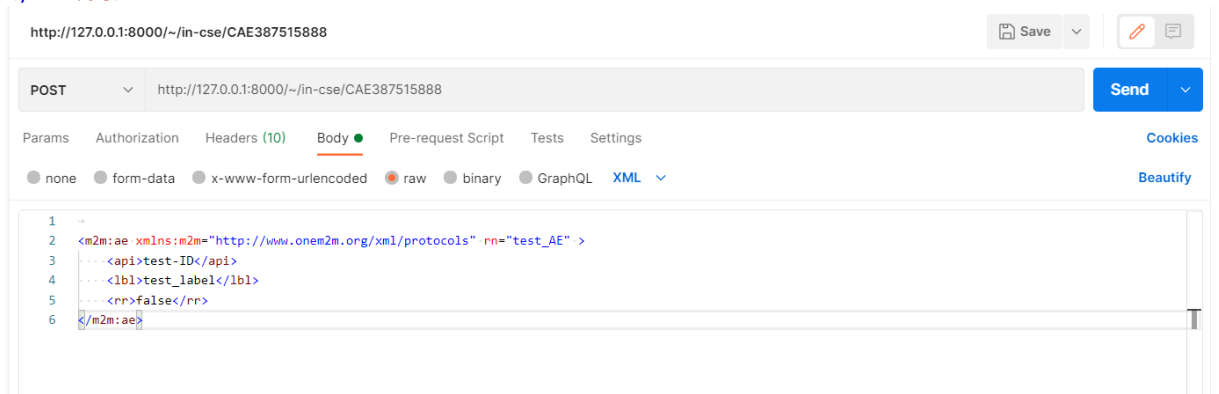
## Part 2: Creating AE

1. Start the Postman.
2. In URL put <http://127.0.0.1:8000/~in-cse/in-name>, In headers add X-M2M-Origin as key and admin:admin as value, then add another key Content-type and application/xml;ty=2



3. In body select raw and then add following xml text

```
<m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="test_AE" >
  <api>test-ID</api>
  <lbl>test_label</lbl>
  <rr>false</rr>
</m2m:ae>
```



“rn” stands for resource name and represents the name of AE.

“api” stands for application ID.

“rr” stands for resource reachability.

“lbl” stands for label.

4. Now change the method from GET to POST and then click Send.

5. If successful you will get an output at the bottom as shown below

http://127.0.0.1:8000/~in-cse/in-name

POST http://127.0.0.1:8000/~in-cse/in-name

Body

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="test_AE">
3   <api>test-ID</api>
4   <lbl>test_label</lbl>
```

Status: 201 Created Time: 43 ms Size: 675 B

Body

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <m2m:ae xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m.org/xml/protocols/homedomain" rn="test_AE">
3   <ty>2</ty>
4   <ri>/in-cse/CAE944871219</ri>
5   <pi>/in-cse</pi>
6   <ct>20210401T230326</ct>
7   <lt>20210401T230326</lt>
8   <lbl>test_label</lbl>
9   <acpi>/in-cse/acp-473755009</acpi>
10  <et>20220401T230326</et>
11  <api>test-ID</api>
12  <aei>CAE944871219</aei>
13  <rr>false</rr>
14 </m2m:ae>
```

6. Go to the OM2M tab in browser and then click on in-name you will see test\_AE in the tree

Logout

OM2M CSE Resource Tree

http://127.0.0.1:8000/~in-cse/CAE944871219

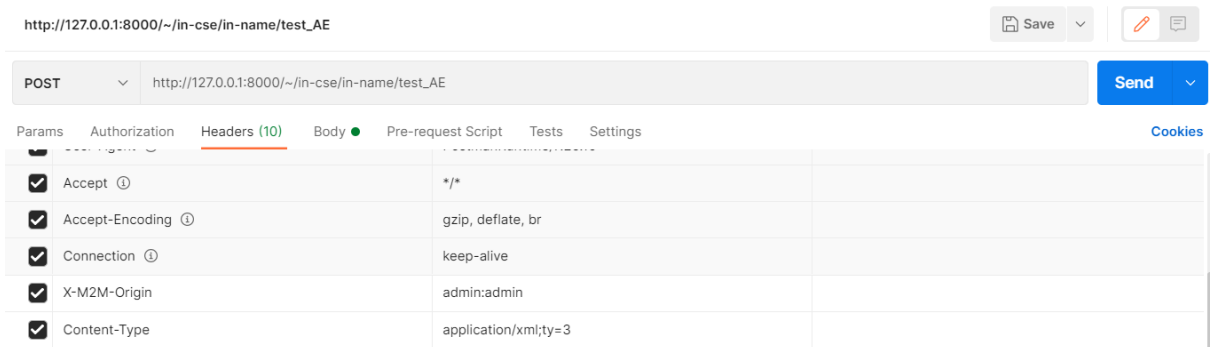
in-name

- acp\_admin
- test\_AE

Attribute	Value
rn	test_AE
ty	2
ri	/in-cse/CAE944871219
pi	/in-cse
ct	20210401T230326
lt	20210401T230326
lbl	• test_label

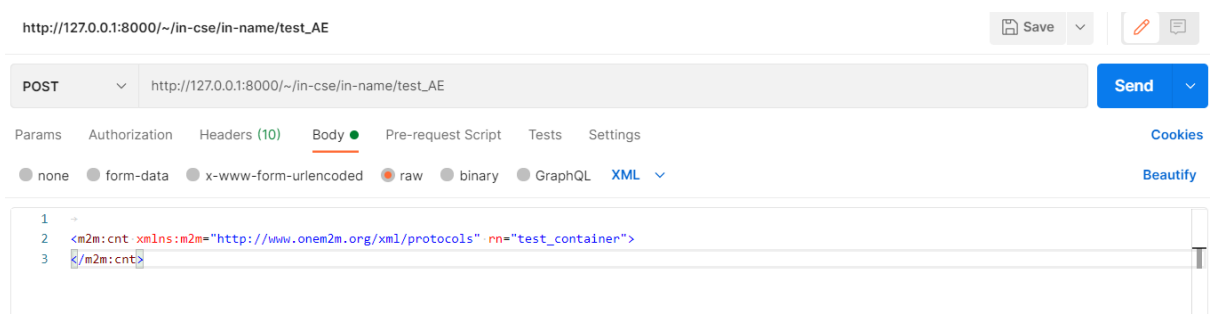
## Part 2: Creating CNT

1. In this section you are going to create a container (CNT) inside AE using postman
2. Change the url to [http://127.0.0.1:8000/~in-cse/in-name/test\\_AE](http://127.0.0.1:8000/~in-cse/in-name/test_AE), In headers change Content-type value to application/xml;ty=3



3. Change the text in body to the xml text mentioned below:

```
<m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="test_container">
</m2m:cnt>
```



“rn” stands for resource name and represents the name of CNT.

#### 4. Now click Send and if successful you will get an output as below

http://127.0.0.1:8000/~in-cse/in-name/test\_AE

POST http://127.0.0.1:8000/~in-cse/in-name/test\_AE

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL XML

```
1 <m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" rn="test_container">
2 </m2m:cnt>
```

Body Cookies Headers (6) Test Results Status: 201 Created Time: 28 ms Size: 818 B Save Response

Pretty Raw Preview Visualize XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <m2m:cnt xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m.org/xml/protocols/homedomain" rn="test_container">
3   <ty>3</ty>
4   <ri>/in-cse/cnt-803275457</ri>
5   <pi>/in-cse/CAE245603325</pi>
6   <ct>20210401T233031</ct>
7   <lt>20210401T233031</lt>
8   <acpi>/in-cse/acp-473755009</acpi>
9   <et>20220401T233031</et>
10  <st>0</st>
11  <mni>10</mni>
12  <mbs>10000</mbs>
13  <mia>0</mia>
14  <cni>0</cni>
15  <cbs>0</cbs>
16  <ol>/in-cse/in-name/test_AE/test_container/ol</ol>
17  <la>/in-cse/in-name/test_AE/test_container/la</la>
18 </m2m:cnt>
```

#### 5. Go to the OM2M tab in browser and then click on test\_AE you will see test\_container inside test\_AE

Logout

OM2M CSE Resource Tree

http://127.0.0.1:8000/~in-cse/cnt-803275457

- in-name
  - acp\_admin
  - test\_AE
    - test\_container



Attribute	Value
rn	test_container
ty	3
ri	/in-cse/cnt-803275457
pi	/in-cse/CAE245603325
ct	20210401T233031
lt	20210401T233031
acpi	<div>AccessControlPolicyIDs</div> <div>/in-cse/acp-473755009</div>
et	20220401T233031
st	0
mni	10

### Part 3: Creating a content instance

1. In this section you are going to create a content instance inside test\_container
2. Change the url to [http://127.0.0.1:8000/~in-cse/in-name/test\\_AE/test\\_container](http://127.0.0.1:8000/~in-cse/in-name/test_AE/test_container), In headers change Content-type value to application/xml;ty=4

The screenshot shows a REST client interface with the URL `http://127.0.0.1:8000/~in-cse/in-name/test_AE/test_container`. The method is set to **POST**. The **Headers** tab is selected, showing the following headers:

Header	Value
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
X-M2M-Origin	admin:admin
Content-Type	application/xml;ty=4

3. Change the text in body to the xml text mentioned below:

```
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
  <cnf>message</cnf>
  <con>test instance</con>
</m2m:cin>
```

The screenshot shows the same REST client interface, but now the **Body** tab is selected. The body is configured as **raw** XML. The XML content is:

```
1 <m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols">
2   ...<cnf>message</cnf>
3   ...<con>test instance</con>
4 </m2m:cin>
```

“cnf” stands for confirmation.

“con” stands for content.

4. Now click Send and if successful you will get an output as below

The screenshot shows the REST client interface after the request has been sent. The **Test Results** tab is selected, showing the response body in **XML** format. The response is a 201 Created status with the following XML content:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m.org/xml/protocols/homedomain" rn="cin_394993463">
3   <ty>4</ty>
4   <ri>/in-cse/cin-394993463</ri>
5   <pi>/in-cse/cnt-803275457</pi>
6   <ct>20210401T234755</ct>
7   <lt>20210401T234755</lt>
8   <st>0</st>
9   <cnf>message</cnf>
10  <cs>13</cs>
11  <con>test instance</con>
12 </m2m:cin>
```

5. Go to the OM2M tab in browser and then click on test\_container you will see a content instance inside it

[Logout](#)

OM2M CSE Resource Tree

<http://127.0.0.1:8000/~in-cse/cin-394993463>

```
- in-name
  - acp_admin
  - test_AE
    - test_container
      - cin_394993463
```



Attribute	Value
rn	cin_394993463
ty	4
ri	/in-cse/cin-394993463
pi	/in-cse/cnt-803275457
ct	20210401T234755
lt	20210401T234755
st	0
cnf	message
cs	13
con	test instance



## Part 4: Get latest Data from test\_container

1. Change the POST method to GET method and url to [http://127.0.0.1:8000/~in-cse/in-name/test\\_AE/test\\_container/la](http://127.0.0.1:8000/~in-cse/in-name/test_AE/test_container/la)
2. Change Content-type in header to application/xml

The screenshot shows a REST client interface with the URL `http://127.0.0.1:8000/~in-cse/in-name/test_AE/test_container/la`. The method is set to GET. The Headers tab is active, showing a table with the following content:

Key	Value	Description
<input checked="" type="checkbox"/> Connection	keep-alive	
<input checked="" type="checkbox"/> X-M2M-Origin	admin:admin	
<input checked="" type="checkbox"/> Content-Type	application/xml	

3. Change the radio button in body to none

The screenshot shows the same REST client interface, but the Body tab is now active. It displays radio buttons for different body types: none (selected), form-data, x-www-form-urlencoded, raw, binary, and GraphQL. Below the radio buttons, it says "This request does not have a body".

4. Now click on Send and you should get a output as shown below

The screenshot shows the REST client interface after sending the request. The Status bar indicates "Status: 200 OK Time: 34 ms Size: 572 B". The Body tab is active, showing the response in XML format. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<m2m:cin xmlns:m2m="http://www.onem2m.org/xml/protocols" xmlns:hd="http://www.onem2m.org/xml/protocols/homedomain" rn="cin_394993463">
  <ty>4</ty>
  <ri>/in-cse/cin-394993463</ri>
  <pi>/in-cse/cnt-803275457</pi>
  <ct>20210401T234755</ct>
  <lt>20210401T234755</lt>
  <st>0</st>
  <cnf>message</cnf>
  <cs>13</cs>
  <con>test instance</con>
</m2m:cin>
```

You can see that you get the data inside <con> tag