FIWARE-EPCIS MEDIATION GATEWAY DEVELOPER GUIDE

Developer Guide

Abstract

This document includes how to use FIWARE-EPCIS mediation gateway. This document doesn’t include how to use EPCIS and FIWARE in detail. It assumes the user is familiar with each of them fairly.

Yalew kidane

yalewkidane@gmail.com/@kaist.ac.kr

Contents

Server Installed:

Wise-IoT-Server: 143.248.57.28

Installation

Use the following page to install FIWARE

https://github.com/telefonicaid/fiware-orion/blob/master/doc/manuals/admin/yum.md

Servers to be prepared

- Fiware server
- EPCIS server
- Mediation gateway server

How to run the mediation gateway

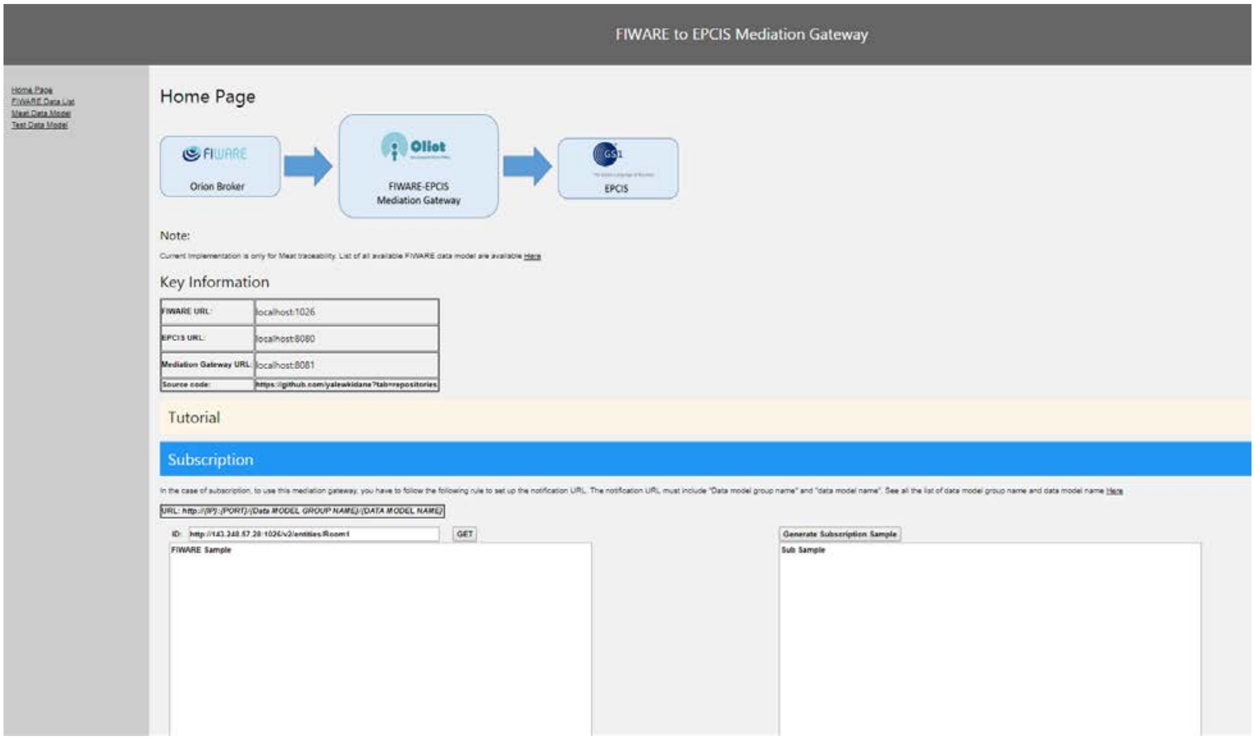
```
java -jar <path/to/my/jar> --server.port=8080
```

Enter Fiware :

Enter :

Main page

{IP}:{PORT} /home



Home Page

Note:

Current Implementation is only for Meat traceability. List of all available FIWARE data model are available Here

Key Information

FIWARE URL:	localhost:1026
EPCIS URL:	localhost:8080
Mediation Gateway URL:	localhost:8081
Source code:	https://github.com/yalewkidane?tab=repositories

Tutorial

Subscription

In the case of subscription, to use this mediation gateway, you have to follow the following rule to set up the notification URL. The notification URL must include "Data model group name" and "data model name". See all the list of data model group name and data model name Here

URL: http://{IP}:{PORT}/{Data MODEL GROUP NAME}/{DATA MODEL NAME}

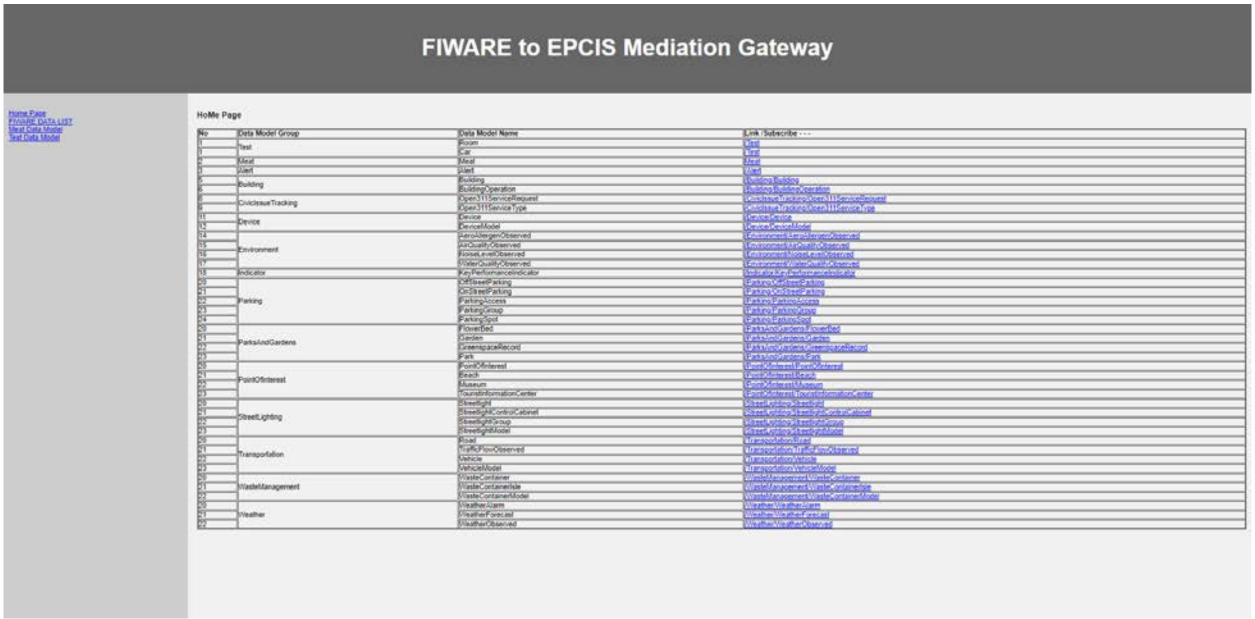
ID: http://143.248.57.28:1026/v2/entities/Room1

FIWARE Sample

Sub Sample

Fiware Data List

{IP}:{PORT} /FiwareDataModel



FIWARE to EPCIS Mediation Gateway

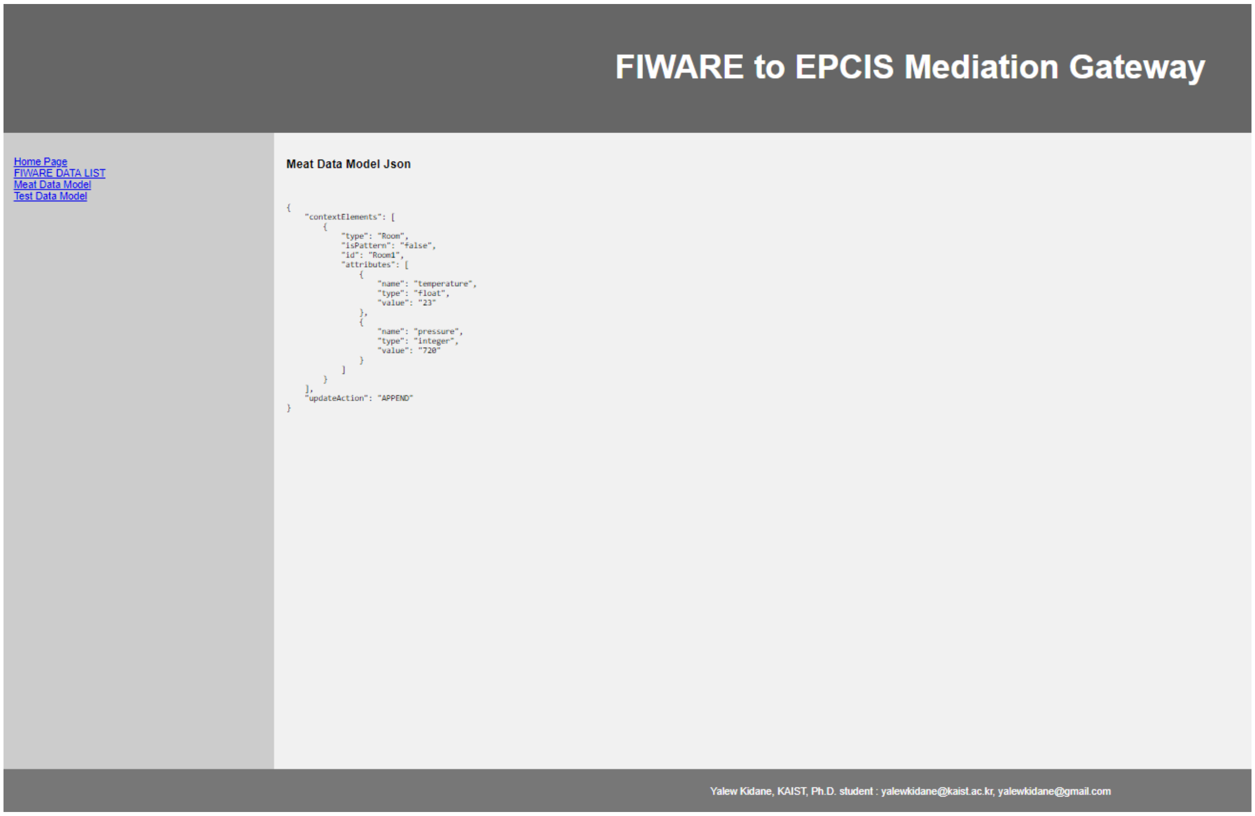
Home Page
FIWARE DATA LIST
Meat Data Model
Test Data Model

HoMe Page

No	Data Model Group	Data Model Name	Link / Subscribe - - -
1	Test	Room	/Test
1		Car	/Test
2	Meat	Meat	/Meat
3	Alert	Alert	/Alert
5	Building	Building	/Building/Building
6		BuildingOperation	/Building/BuildingOperation
8	CivicIssueTracking	Open311ServiceRequest	/CivicIssueTracking/Open311ServiceRequest
9		Open311ServiceType	/CivicIssueTracking/Open311ServiceType
11	Device	Device	/Device/Device
12		DeviceModel	/Device/DeviceModel
14	Environment	AeroAllergenObserved	/Environment/AeroAllergenObserved
15		AirQualityObserved	/Environment/AirQualityObserved
16		NoiseLevelObserved	/Environment/NoiseLevelObserved
17		WaterQualityObserved	/Environment/WaterQualityObserved
18	Indicator	KeyPerformanceIndicator	/Indicator/KeyPerformanceIndicator
20	Parking	OffStreetParking	/Parking/OffStreetParking
21		OnStreetParking	/Parking/OnStreetParking
22		ParkingAccess	/Parking/ParkingAccess
23		ParkingGroup	/Parking/ParkingGroup
24		ParkingSpot	/Parking/ParkingSpot
20	ParksAndGardens	FlowerBed	/ParksAndGardens/FlowerBed
21		Garden	/ParksAndGardens/Garden
22		GreenspaceRecord	/ParksAndGardens/GreenspaceRecord
23		Park	/ParksAndGardens/Park
20	PointOfInterest	PointOfInterest	/PointOfInterest/PointOfInterest
21		Beach	/PointOfInterest/Beach
22		Museum	/PointOfInterest/Museum
23		TouristInformationCenter	/PointOfInterest/TouristInformationCenter
20	StreetLighting	Streetlight	/StreetLighting/Streetlight
21		StreetlightControlCabinet	/StreetLighting/StreetlightControlCabinet
22		StreetlightGroup	/StreetLighting/StreetlightGroup
23		StreetlightModel	/StreetLighting/StreetlightModel
20	Transportation	Road	/Transportation/Road
21		TrafficFlowObserved	/Transportation/TrafficFlowObserved
22		Vehicle	/Transportation/Vehicle
23		VehicleModel	/Transportation/VehicleModel
20	WasteManagement	WasteContainer	/WasteManagement/WasteContainer
21		WasteContainerIsle	/WasteManagement/WasteContainerIsle
22		WasteContainerModel	/WasteManagement/WasteContainerModel
20	Weather	WeatherAlarm	/Weather/WeatherAlarm
21		WeatherForecast	/Weather/WeatherForecast
22		WeatherObserved	/Weather/WeatherObserved

Meat Data List

{IP}:{PORT} /meat



FIWARE to EPCIS Mediation Gateway

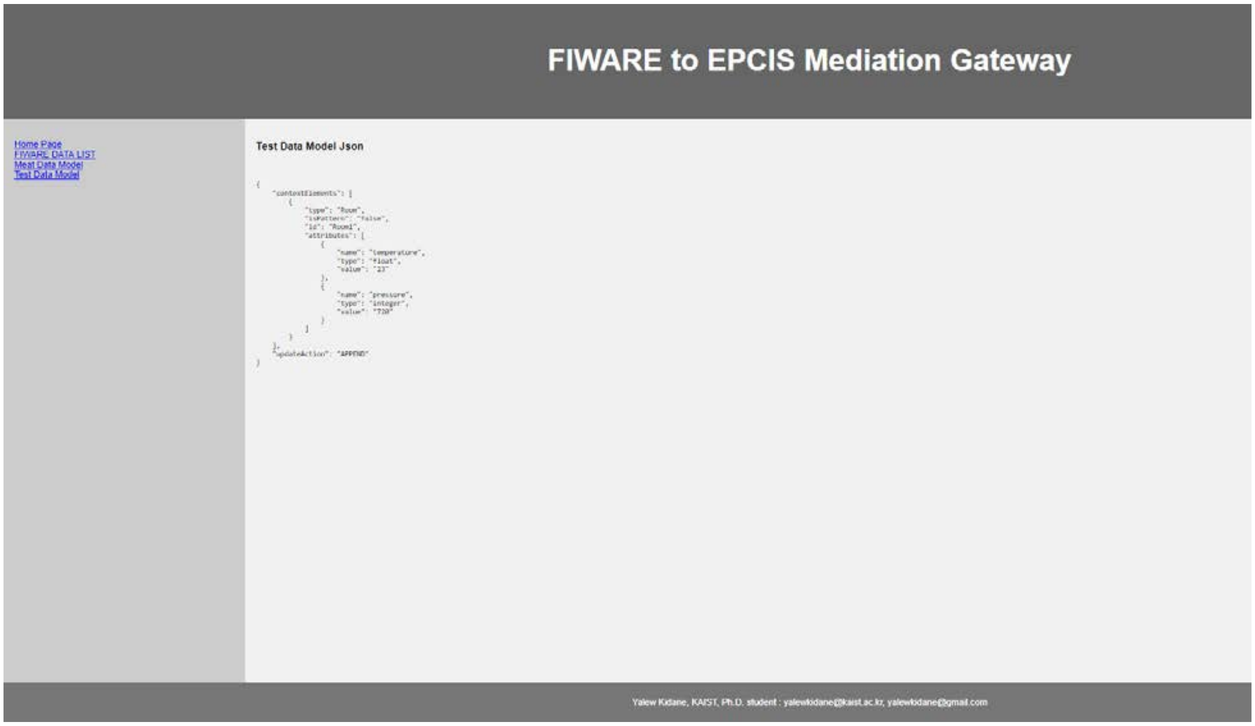
Meat Data Model Json

```
{
    "contextElements": [
        {
            "type": "Room",
            "isPattern": "false",
            "id": "Room1",
            "attributes": [
                {
                    "name": "temperature",
                    "type": "float",
                    "value": "23"
                },
                {
                    "name": "pressure",
                    "type": "integer",
                    "value": "720"
                }
            ]
        }
    ],
    "updateAction": "APPEND"
}
```

Yalew Kidane, KAIST, Ph.D. student : yalewkidane@kaist.ac.kr, yalewkidane@gmail.com

Test Data List

{IP}:{PORT} /test



FIWARE to EPCIS Mediation Gateway

Home Page
FIWARE DATA LIST
Meat Data Model
Test Data Model

Test Data Model Json

```
{
    "contextElements": [
        {
            "type": "Room",
            "isPattern": "false",
            "id": "Room1",
            "attributes": [
                {
                    "name": "temperature",
                    "type": "float",
                    "value": "23"
                },
                {
                    "name": "pressure",
                    "type": "integer",
                    "value": "720"
                }
            ]
        }
    ],
    "updateAction": "APPEND"
}
```

Yalew Kidane, KAIST, Ph.D. student : yalewkidane@kaist.ac.kr, yalewkidane@gmail.com

MongoDB operation

//To start mongodb in terminal

```
$mongo
```

To switch to orion data base

```
$use orion
```

//To see all list of collections → csubs entities

```
$show collections
```

//To see all the subscriptions

```
db.csubs.find()
```

//TO see all the list of entities

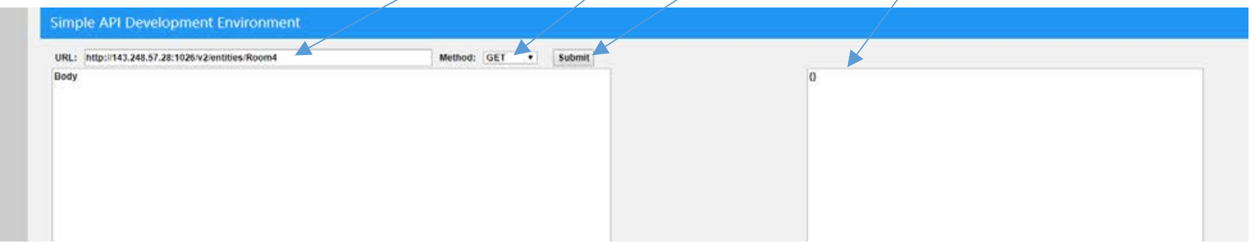
```
$ db. entities.find()
```

//To delete all subscriptions

```
$db.csubs.deleteMany({})
```

Simple Subscription example

1. Check for a specific entities (eg. Room4)



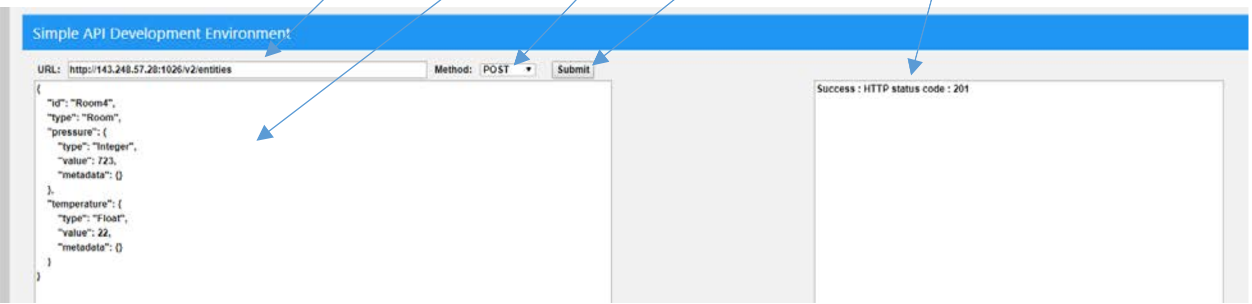
URL: http://143.248.57.28:1026/v2/entities/Room4

Method: GET

Body

{}

2. Add an entity (eg. Room4)



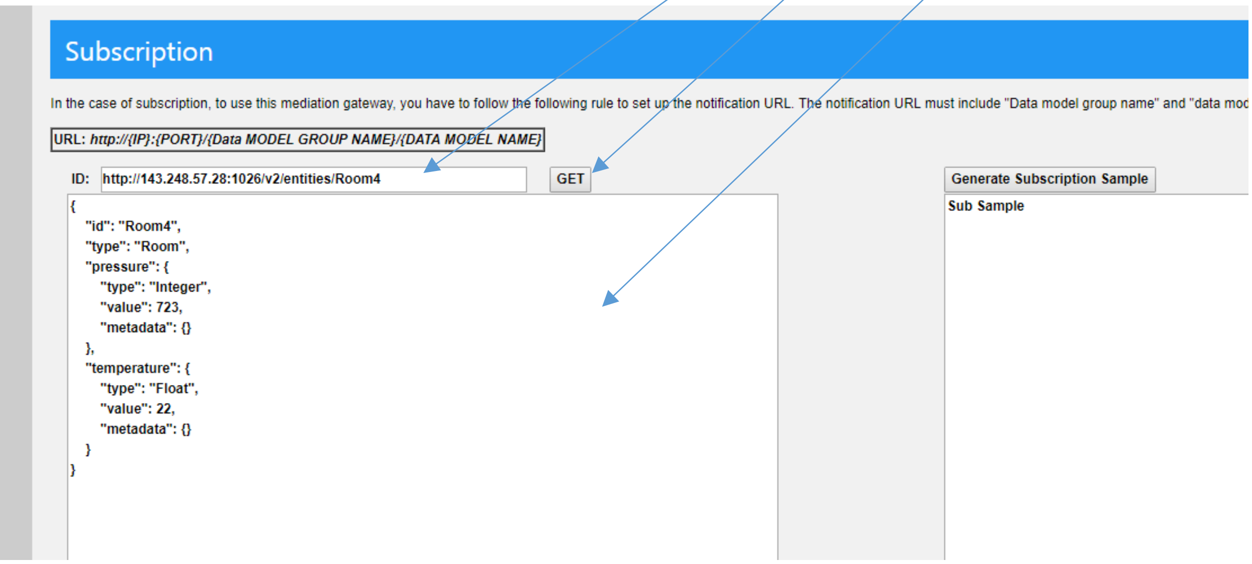
URL: http://143.248.57.28:1026/v2/entities

Method: POST

```
{
  "id": "Room4",
  "type": "Room",
  "pressure": {
    "type": "Integer",
    "value": 723,
    "metadata": {}
  },
  "temperature": {
    "type": "Float",
    "value": 22,
    "metadata": {}
  }
}
```

Success : HTTP status code : 201

3. Check the entity



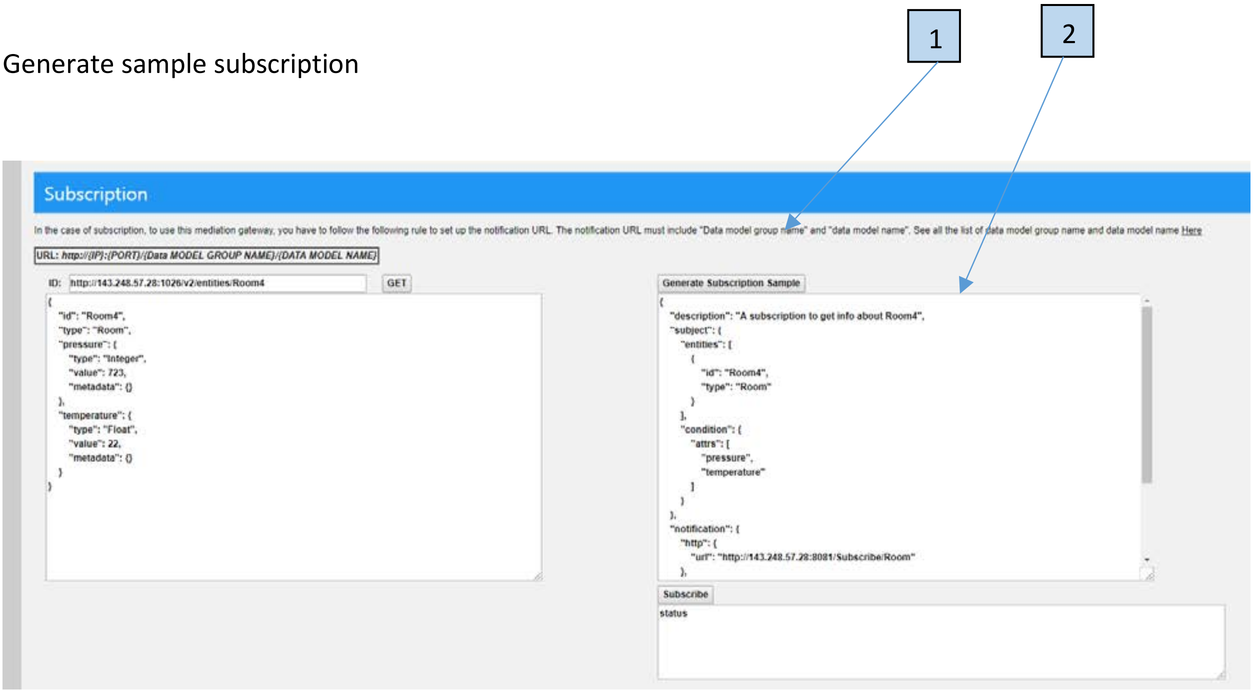
In the case of subscription, to use this mediation gateway, you have to follow the following rule to set up the notification URL. The notification URL must include "Data model group name" and "data mod

URL: *http://{IP}:{PORT}/{Data MODEL GROUP NAME}/{DATA MODEL NAME}*

ID: http://143.248.57.28:1026/v2/entities/Room4

```
{
  "id": "Room4",
  "type": "Room",
  "pressure": {
    "type": "Integer",
    "value": 723,
    "metadata": {}
  },
  "temperature": {
    "type": "Float",
    "value": 22,
    "metadata": {}
  }
}
```

4. Generate sample subscription



In the case of subscription, to use this mediation gateway, you have to follow the following rule to set up the notification URL. The notification URL must include "Data model group name" and "data model name". See all the list of data model group name and data model name Here

URL: *http://{IP}:{PORT}/{Data MODEL GROUP NAME}/{DATA MODEL NAME}*

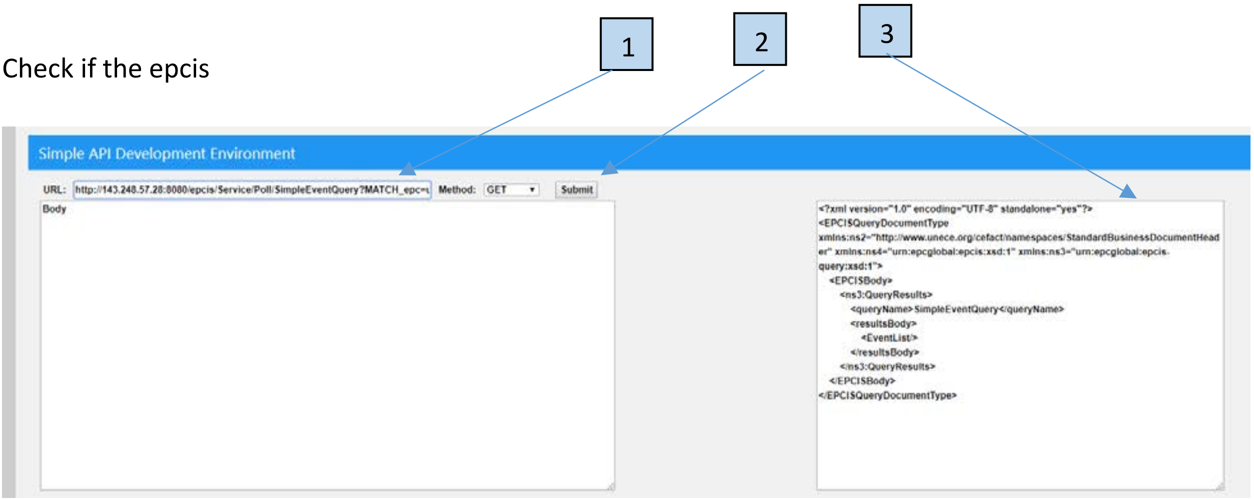
ID: http://143.248.57.28:1026/v2/entities/Room4

```
{
  "id": "Room4",
  "type": "Room",
  "pressure": {
    "type": "Integer",
    "value": 723,
    "metadata": {}
  },
  "temperature": {
    "type": "Float",
    "value": 22,
    "metadata": {}
  }
}
```

```
{
  "description": "A subscription to get info about Room4",
  "subject": {
    "entities": [
      {
        "id": "Room4",
        "type": "Room"
      }
    ],
    "condition": {
      "attrs": [
        "pressure",
        "temperature"
      ]
    }
  },
  "notification": {
    "http": {
      "url": "http://143.248.57.28:8081/Subscribe/Room"
    },
```

status

5. Check if the epcis

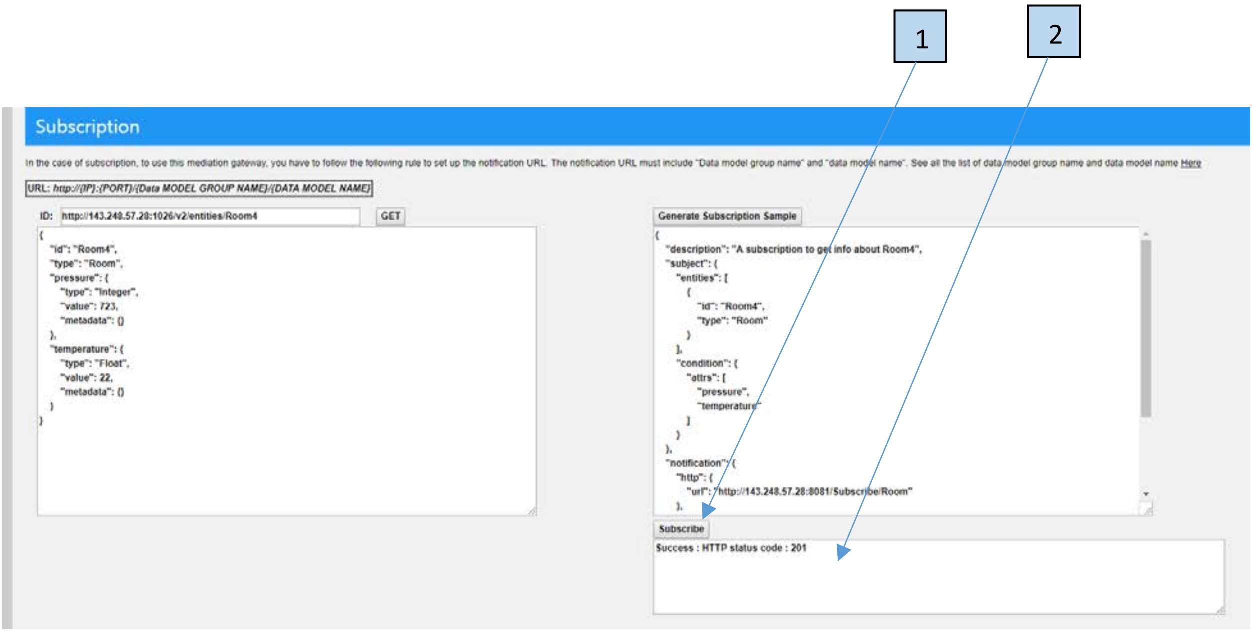


URL: http://143.248.57.28:8080/epcis/Service/Poll/SimpleEventQuery?MATCH_epc=u Method: GET

Body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EPCISQueryDocumentType
xmlns:ns2="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHead
er" xmlns:ns4="urn:epcglobal:epcis:xsd:1" xmlns:ns3="urn:epcglobal:epcis-
query:xsd:1">
  <EPCISBody>
    <ns3:QueryResults>
      <queryName>SimpleEventQuery</queryName>
      <resultsBody>
        <EventList/>
      </resultsBody>
    </ns3:QueryResults>
  </EPCISBody>
</EPCISQueryDocumentType>
```

6. Subscribe



In the case of subscription, to use this mediation gateway, you have to follow the following rule to set up the notification URL. The notification URL must include "Data model group name" and "data model name". See all the list of data model group name and data model name Here

URL: http://{IP}:{PORT}/{Data MODEL GROUP NAME}/{DATA MODEL NAME}

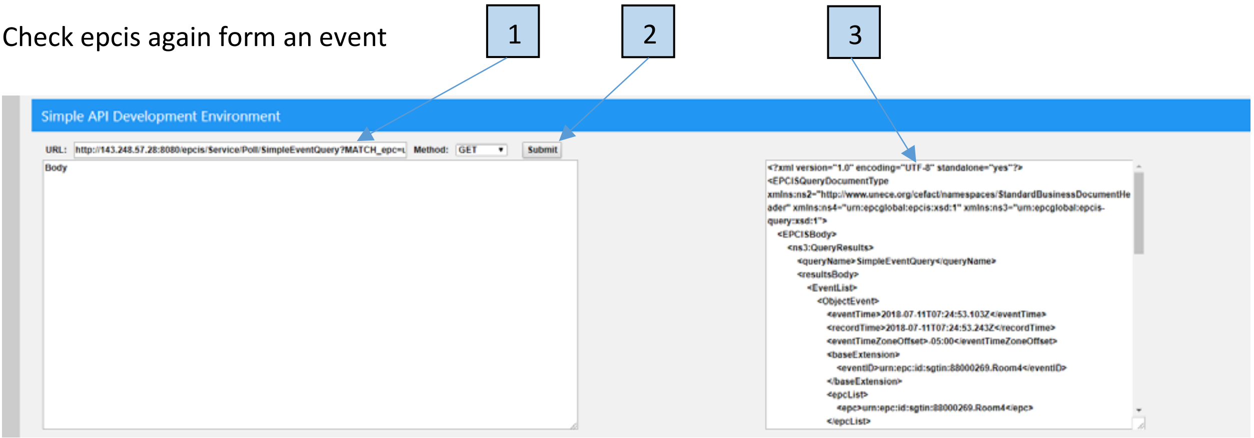
ID: http://143.248.57.28:1026/v2/entities/Room4

```
{
  "id": "Room4",
  "type": "Room",
  "pressure": {
    "type": "Integer",
    "value": 723,
    "metadata": {}
  },
  "temperature": {
    "type": "Float",
    "value": 22,
    "metadata": {}
  }
}
```

```
{
  "description": "A subscription to get info about Room4",
  "subject": {
    "entities": [
      {
        "id": "Room4",
        "type": "Room"
      }
    ],
    "condition": {
      "attrs": [
        "pressure",
        "temperature"
      ]
    }
  },
  "notification": {
    "http": {
      "url": "http://143.248.57.28:8081/Subscribe/Room"
    },
```

```
Success : HTTP status code : 201
```

7. Check epcis again form an event

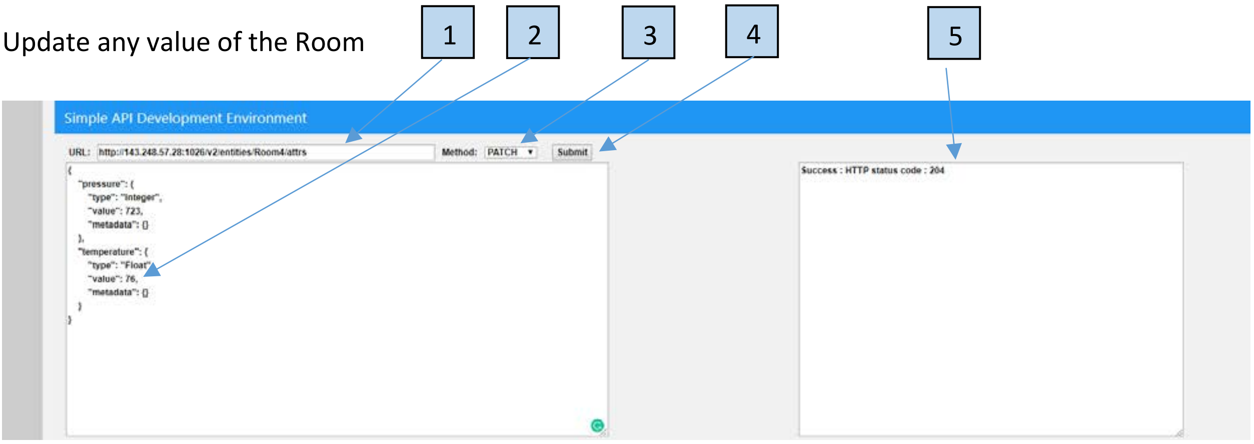


URL: http://143.248.57.28:8080/epcis/Service/Poll/SimpleEventQuery?MATCH_epc=u Method: GET

Body

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EPCISQueryDocumentType
xmlns:ns2="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHe
ader" xmlns:ns4="urn:epcglobal:epcis:xsd:1" xmlns:ns3="urn:epcglobal:epcis-
query:xsd:1">
  <EPCISBody>
    <ns3:QueryResults>
      <queryName>SimpleEventQuery</queryName>
      <resultsBody>
        <EventList>
          <ObjectEvent>
            <eventTime>2018-07-11T07:24:53.103Z</eventTime>
            <recordTime>2018-07-11T07:24:53.243Z</recordTime>
            <eventTimeZoneOffset>-05:00</eventTimeZoneOffset>
            <baseExtension>
              <eventID>urn:epc:id:sgtin:88000269.Room4</eventID>
            </baseExtension>
            <epcList>
              <epc>urn:epc:id:sgtin:88000269.Room4</epc>
            </epcList>
```

8. Update any value of the Room

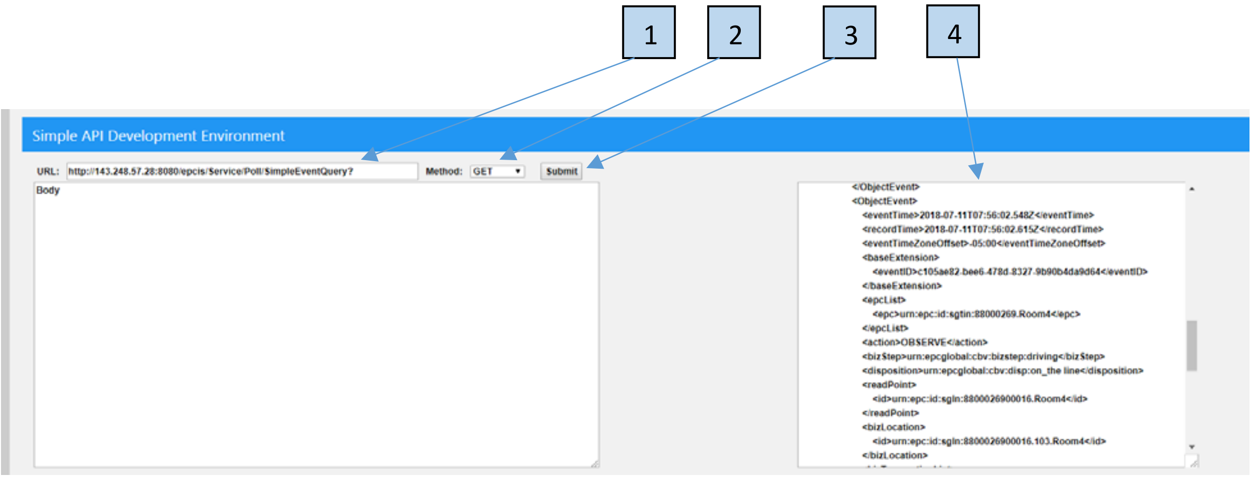


URL: http://143.248.57.28:1026/v2/entities/Room4/attrs Method: PATCH

```
{
  "pressure": {
    "type": "Integer",
    "value": 723,
    "metadata": {}
  },
  "temperature": {
    "type": "Float",
    "value": 76,
    "metadata": {}
  }
}
```

```
Success : HTTP status code : 204
```

9. Check epcis if there is two Room4 object



URL: http://143.248.57.28:8080/epcis/Service/Poll/SimpleEventQuery?

Method: GET

```
</ObjectEvent>
<ObjectEvent>
   <eventTime>2018-07-11T07:56:02.548Z</eventTime>
   <recordTime>2018-07-11T07:56:02.615Z</recordTime>
   <eventTimeZoneOffset>-05:00</eventTimeZoneOffset>
   <baseExtension>
      <eventID>c105ae82-bee6-478d-8327-9b90b4da9d64</eventID>
   </baseExtension>
   <epcList>
      <epc>urn:epc:id:sgtin:88000269.Room4</epc>
   </epcList>
   <action>OBSERVE</action>
   <bizStep>urn:epcglobal:cbv:bizstep:driving</bizStep>
   <disposition>urn:epcglobal:cbv:disp:on_the line</disposition>
   <readPoint>
      <id>urn:epc:id:sgln:880002690001 6.Room4</id>
   </readPoint>
   <bizLocation>
      <id>urn:epc:id:sgln:8800026900016.103.Room4</id>
   </bizLocation>
```

10.