FIWARE-ORION

ORION DOCUMENTATION

Entities v1

Types v1

Subscriptions v1

Entities v2

Types v2

Subscriptions v2

Entities

create context entity v1

`http://{{url}}:{{port-orion}}/v1/registry/registerContext`

Definition: Method to create a context of entity, an entity without values.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
    "contextRegistrations": [
        {
            "entities": [
                {
                    "type": "entity_type",
                    "isPattern": "false",
                    "id": "entity_id"
                }
            ],
            "attributes": [
                {
                    "name": "nombre_atributo",
                    "type": "attribute_type",
                    "isDomain": "false"
                }
            ],
            "providingApplication": "http://homard.hopu.eu:1026/v2/entities"
        }
    ],
    "duration": "P1M"
}
```

- `isPattern:` Currently is hasn't use. Always 'false'.
- `isDomain:` The attribute domains aren't supported. Always 'false'.
- `providingApplication:` The URL that represents the context information of the registered entities and attributes.
- `duration:` The duration of the element. In ISO 8601 standard format.

Return:

Returns a confirmation that the item has been created correctly.

```
{
    "duration": "P1M",
    "registrationId": "5a79812d777fc523840b8446"
}
```

retrieve context entity v1

`http://{{url}}:{{port-orion}}/v1/registry/discoverContextAvailability`

Definition: Method to retrieve a context of entity, an entity without values, depends on ID and type.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
    "entities": [
        {
            "type": "entity_type”,
            "isPattern": "false",
            "id": "entity_id"
        }
    ]
}
```

- `isPattern`: Currently is hasn’t use. Always ‘false’.

Return:

In case of find this element, it return the information.
In case of don’t find this element, it return a 404 error “`No context element found`”.

---

create/update entity v1

`http://{{url}}:{{port-orion}}/v1/updateContext`
Definition: Method to create or update an entity, depends on method’s body.
Method: POST
Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

- **Creation**

```
{
    "contextElements": [
        {
              "type": "entity_type",
              "isPattern": "false",
              "id": "entity_id ",
              "attributes": [
                     {
                     "name":"attribute_id ",
                     "type": "attribute_type",
                     "value": "attribute_value"
                     }
              ]
        }
    ],
    "updateAction": "APPEND"
}
```

  - `isPattern`: Currently is hasn’t use. Always ‘false’.
  - `updateAction`: Action to be taken (“APPEND” o “UPDATE”). In case of “APPEND” create the entity.

- **Updating**

```
{
    "contextElements": [
        {
              "type": "entity_type",
              "isPattern": "false",
              "id": "entity_id ",
              "attributes": [
                     {
                     "name":"attribute_id",
                     "type": "attribute_type",
                     "value": "attribute_value"
                     }
              ]
       }
    ],
    "updateAction": "UPDATE"
}
```

- `isPattern:` Currently is hasn't use. Always 'false'.
- `updateAction:` Action to be taken ("APPEND" o "UPDATE"). In case of "UPDATE" update the entity.

Return:
In case of performing the method correctly, it return 200 OK together with the information of the entity.
In UPDATE, in case of not find that entity, it return a 404 ERROR "`No context element found`".

---

create entity v1

`http://{{url}}:{{port-orion}}/v1/contextEntities/{{fiware-entity}}`

Definition: Method to create a new entity
Method: POST
Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be create of the service.

Body:

```
{
       "type": "entity_type",
       "attributes": [
              {
                     "name": "attribute_id",
                     "type": "attribute_type",
                     "value": "attribute_value"
              }
       ]
}
```

Return:
In case of performing the method correctly, it return 200 OK together with the information of the new entity.

---

retrieve entity standard v1

`http://{{url}}:{{port-orion}}/v1/contextEntities/{{fiware-entity}}`

Definition: Method to return a defined entity passed as parameter.
Method: GET
Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be retrieve of the service.

Return:
In case of performing the method correctly, it return 200 OK together with the information of the entity.
In case of not find that entity, it return a 404 ERROR “`No context element found`”.

---

retrieve entity as object standard v1

```
http://{{url}}:{{port-orion}}/v1/registry/contextEntities/{{fiware-type}}?attributeFormat=object
```

Definition: Method to return a defined entity with object JSON format.
Method: GET
Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be retrieve of the service.

Return:
In case of performing the method correctly, it return 200 OK together with the information of the entity with object JSON format.
In case of not find that entity, it return a 404 ERROR “`No context element found`”.

---

retrieve entity convenience v1

```
http://{{url}}:{{port-orion}}/v1/queryContext
```

Definition: Method to returns a defined entity passed in the body of the method.
Method: POST
Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
    "entities": [
        {
            "type": "entity_type",
            "isPattern": "false",
            "id": "entity_id"
        }
    ]
}
```

Return:
In case of performing the method correctly, it return 200 OK together with the information of the entity.
In UPDATE, in case of not find that entity, it return a 404 ERROR “`No context element found`”.

---

retrieve entity as object convenience v1

`http://{{url}}:{{port-orion}}/v1/queryContext?attributeFormat=object`

Definition: Method to return a defined entity with object JSON format. The entity ID is passed in the body of method.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
    "entities": [
        {
            "type": "entity_type",
            "isPattern": "false",
            "id": "entity_id"
        }
    ]
}
```

Return:

In case of performing the method correctly, it return 200 OK together with the information of the entity with object JSON format.

In UPDATE, in case of not find that entity, it return a 404 ERROR “`No context element found`”.

---

retrieve entities v1

`http://{{url}}:{{port-orion}}/v1/contextEntities`

Definition: Method to return all entities.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Return:

Show all existing entities.

---

retrieve entities as object v1

`http://{{url}}:{{port-orion}}/v1/contextEntities?attributeFormat=object`

Definition: Method to return all entities with object JSON format.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Return:

Show all existing entities with object JSON format.

retrieve entities for type v1

`http://{{url}}:{{port-orion}}/v1/registry/contextEntityTypes/{{fiware-type}}`

Definition: Method to returns all entities of a concrete type.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-type: Type of entities that we want to receive.

Return:

In case of performing the method correctly, it return 200 OK together with the information of the all entities of this type.

In UPDATE, in case of not find that entity, it return a 404 ERROR "`No context element found`".

retrieve entities for type as object v1

`http://{{url}}:{{port-orion}}/v1/registry/contextEntityTypes/{{fiware-type}}?attributeFormat=object`

Definition: Method to return all entities of a concrete type with object JSON format.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-type: Type of entities that we want to receive.

Return:

In case of performing the method correctly, it return 200 OK together with the information of the all entities of this type with object JSON format.

In UPDATE, in case of not find that entity, it return a 404 ERROR "No context element found".

delete entity v1

`http://{{url}}:{{port-orion}}/v1/contextEntities/{{fiware-entity}}`

Definition: Method to delete a defined entity.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity we want to delete.

Return:

In case of performing the method correctly and delete the entity defined, it return 200 OK .

In case of not find that entity, it return a 404 ERROR "No context element found".

create/update entity attribute v1

`http://{{url}}:{{port-orion}}/v1/contextEntities/{{fiware-entity}}/attributes`

Definition: Method to update the values of one or many attributes of an entity passed as parameter. Also, with this method we can create new attributes for an entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be create of the service.

Body:

- Update an attribute. We passed an existed attribute with a new value.

```
{
        "attributes": [
                {
                "name": "attribute_id",
                "type": "attribute_type",
                "value": "attribute_value"
                }
        ]
}
```

- Create a new attribute. We passed a new attribute in the body of method:

```
{
        "attributes": [
                {
                "name": "attribute_id_nuevo",
                "type": "attribute_type_nuevo",
                "value": "attribute_value_nuevo"
                }
        ]
}
```

Return:

In case of performing the method correctly, it return 200 OK together with the information of the entity updating.

In case of not find that entity, it return a 404 ERROR “`No context element found`”.

---

retrieve entity attribute v1

`http://{{url}}:{{port-orion}}/v1/contextEntities/{{fiware-entity}}/attributes/{{fiware-attr}}`

Definition: Method to retrieve the defined attribute depending on the ID of a concrete entity.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be retrieve of the service.
- fiware-attr: Attribute ID we want to receive.

Return:

In case of performing the method correctly, it return 200 OK together the attribute value defining by its ID for a concrete entity.

In case of not find that entity or attribute, it return 404 ERROR “`No context element found`”.

retrieve entity attribute as object v1

```
http://{{url}}:{{port-orion}}/v1/contextEntities/{{fiware-entity}}/attributes/{{fiware-attr}}?attributeFormat=object
```

Definition: Method to retrieve the defined attribute depending on the ID of a concrete entity with object JSON format.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be retrieve of the service.
- fiware-attr: Attribute ID we want to receive.

Return:

In case of performing the method correctly, it return 200 OK together the attribute value defining by its ID for a concrete entity with object JSON format.

In case of not find that entity or attribute, it return 404 ERROR “`No context element found`”.

retrieve entities attribute for type v1

```
http://{{url}}:{{port-orion}}/v1/registry/contextEntityTypes/{{fiware-
        type}}/attributes/{{fiware-attr}}
```

Definition: Method to return the defined attribute of all entities of a concrete type.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-type: Type of entities that we want to receive.
- fiware-attr: Attribute of type of entities that we want to receive

Return:

In case of performing the method correctly, it return 200 OK together with the information of defined attributes of all entities of this type.

In UPDATE, in case of not find that entity, it return a 404 ERROR “No context element found”.

retrieve entities attribute for type as object v1

```
http://{{url}}:{{port-orion}}/v1/registry/contextEntityTypes/{{fiware-type}}/attributes/{{fiware-attr}}?attributeFormat=object
```

Definition: Method to return the defined attribute of all entities of a concrete type with object JSON format.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-type: Type of entities that we want to receive.
- fiware-attr: Attribute of type of entitites that we want to receive.

Return:

In case of performing the method correctly, it return 200 OK together with the information of defined attributes of all entities of this type with object JSON format.
In UPDATE, in case of not find that entity, it return a 404 ERROR "`No context element found`".

---

delete entity attribute v1

`http://{{url}}:{{port-orion}}/v1/contextEntities/{{fiware-entity}}/attributes/{{fiware-attr}}`

Definition: Method to delete the defined attribute depending on the ID of a concrete entity.
Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be retrieve of the service.
- fiware-attr: Attribute ID we want to receive.

Return:

In case of performing the method correctly and delete the entity defined, it return 200 OK.
In case of not find that entity or attribute, it return 404 ERROR "`No context element found`".

Types

retrieve type v1

`http://{{url}}:{{port-orion}}/v1/contextTypes/{{fiware-type}}`

Definition: Method to receive an entity type concrete.
Method: GET
Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-type: Type of entities that we want to receive.

Return:

In case of performing the method correctly, it return 200 OK together the information of this entities type.
In case of not find that entities type, it return 404 ERROR "`No context element found`".

---

retrieve types v1

`http://{{url}}:{{port-orion}}/v1/contextTypes`

Definition: Method to receive the existing entities types.
Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Return:

In case of performing the method correctly, it return 200 OK together the information of all entities type.

In case of not find that entities type, it return 404 ERROR "`No context element found`".

Subscriptions

create context subscription standard v1

`http://{{url}}:{{port-orion}}/v1/subscribeContextAvailability`

Definition: Method to create a subscription to one or many context entities.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
       "entities": [
               {
              "type": "id_entity_type",
              "isPattern": "false",
              "id": ".*"
              }
       ],
       "attributes": [
              "id_attribute"
       ],
       "reference": "http://cygnus:5050/notify",
       "duration": "P1M"
}
```

- `isPattern:` Currently is hasn't use. Always 'false'.
- `id`: To which entity it wants to subscribe. In this case, to all entities of those type.
- `reference:` URL of client that it want to subscribe.
- `duration:` The duration of the subscription. In ISO 8601 standard format..

Return:

```
{
    "subscriptionId": "subscription_id",
    "duration": "P1M"
}
```

---

create context subscription convenience v1

`http://{{url}}:{{port-orion}}/v1/contextAvailabilitySubscriptions`

Definition: Method to create a subscription to one or many context entities.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
        "entities": [
                 {
                "type": "id_entity_type",
                "isPattern": "false",
                "id": ".*"
                }
        ],
        "attributes": [
                "id_attribute"
        ],
        "reference": "http://cygnus:5050/notify",
        "duration": "P1M"
}
```

- `isPattern:` Currently is hasn't use. Always 'false'.
- id: To which entity it wants to subscribe. In this case, to all entities of those type.
- `reference:` URL of client that it want to subscribe.
- `duration:` The duration of the subscription. In ISO 8601 standard format.

Return:

```
{
    "subscriptionId": "subscription_id",
    "duration": "P1M"
}
```

---

update context subscription standard v1

`http://{{url}}:{{port-orion}}/v1/updateContextAvailabilitySubscriptions`

Definition: Method to update a subscription of a context entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
        "entities": [
                 {
                "type": "id_entity_type",
                "isPattern": "false",
                "id": ".*"
                }
        ],
        "duration": "P1M"
        "subscriptionId": “subscription_id”
}
```

Return:

```
{
    "subscribeResponse": {
   "subscriptionId": "id_subscription",
        "id_parameter": "value_parameter"
    }
}
```

update context subscription convenience v1

```
http://{{url}}:{{port-
orion}}/v1/registry/contextAvailabilitySubscriptions/{{fiware-subscription}}
```

Definition: Method to update a subscription of a context entity.

Method: PUT

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-subscription: ID de la suscripción a actualizar.

Body:

```
{
       "entities": [
              {
             "type": "id_entity_type",
             "isPattern": "false",
             "id": ".*"
             }
       ],
       "duration": "P1M"
       "subscriptionId": “subscription_id”
}
```

Return:

```
{
  "subscribeResponse": {
        "subscriptionId": "id_subscription",
        "id_parameter": "value_parameter"
    }
}
```

delete context subscription standard v1

```
http://{{url}}:{{port-orion}}/v1/unsuscribeContextAvailability
```

Definition: Method to delete a subscription of a context entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
       "subscriptionId": "5a785788777fc523840b843e"
}
```

Return:

In case of performing the method correctly and delete the entity defined, it returns 200 OK together with the removed subscription ID.

In case of not find that subscription, it return 404 ERROR “No context element found”.

delete context subscription convenience v1

`http://{{url}}:{{port-orion}}/v1/contextAvailabilitySubscriptions/{{fiware-subscription}}`

Definition: Method to delete a subscription of a context entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-subscription: ID of subscription to remove.

Return:

In case of performing the method correctly and delete the entity defined, it returns 200 OK together with the removed subscription ID.

In case of not find that subscription, it return 404 ERROR “No context element found”.

---

create subscription standard v1

`http://{{url}}:{{port-orion}}/v1/suscribeContext`

Definition: Method to create a subscription to one or many entities.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
        "entities": [
         {
                "type": "entity_type",
                "isPattern": "false",
                "id": ".*"
        }
        ],
                "attributes": [
                        "attribute_id"
        ],
        "reference": "http://cygnus:5050/notify",
        "duration": "P1M"
        "notifyConditions": [
        {
                "type": "ONCHANGE",
                "condValues": [
                        "attribute_id"
                ]
        }
        ],
        "throttling": "PT5S"
}
```

- `isPattern:` Currently is hasn’t use. Always ‘false’.
- id: To which entity it wants to subscribe. In this case, to all entities of those type.
- `reference:` URL of client that it want to subscribe.
- `duration:` The duration of the subscription. In ISO 8601 standard format.
- `notifyConditions:` Define the launcher to notify the subscriptions. In this case, when change a value of attributes passed in the point “condValues” of an entity will launch a notification.

- `throttling:` Specify a minimum inter-notification arrival time. In this case, 5 seconds.

Return:

```
{
        "subscribeResponse": {
                "subscriptionId": "id_subscription",
                "duration": "P1M"
        }
}
```

create subscription convenience v1

`http://{{url}}:{{port-orion}}/v1/contextSubscriptions`

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
        "entities": [
         {
                "type": "entity_type",
                "isPattern": "false",
                "id": ".*"
        }
        ],
                "attributes": [
                        "attribute_id"
        ],
        "reference": "http://cygnus:5050/notify",
        "duration": "P1M"
        "notifyConditions": [
        {
                "type": "ONCHANGE",
                "condValues": [
                        "attribute_id"
                ]
        }
        ],
        "throttling": "PT5S"
}
```

- `isPattern:` Currently is hasn’t use. Always ‘false’.
- id: To which entity it wants to subscribe. In this case, to all entities of those type.
- `reference:` URL of client that it want to subscribe.
- `duration:` The duration of the subscription. In ISO 8601 standard format.
- `notifyConditions:` Define the launcher to notify the subscriptions. In this case, when change a value of attributes passed in the point “condValues” of an entity will launch a notification.
- `throttling:` Specify a minimum inter-notification arrival time. In this case, 5 seconds.

Return:

```
{
    "subscribeResponse": {
        "subscriptionId": "id_subscription",
        "duration": "P1M"
    }
}
```

update subscription standard v1

```
http://{{url}}:{{port-orion}}/v1/updateContextSubscription
```

Definition: Method to update a subscription of an entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
       "subscriptionId": "id_subscription",
       "id_parameter": "value_parameter"
}
```

Return:

```
{
    "subscribeResponse": {
        "subscriptionId": "id_subscription",
        "id_parameter": "value_parameter"
    }
}
```

update subscription convenience v1

```
http://{{url}}:{{port-orion}}/v1/contextSubscriptions/{{fiware-subscription}}
```

Definition: Method to update a subscription of an entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
       "subscriptionId": "id_subscription",
       "id_parameter": "value_parameter"
}
```

Return:

```
{
    "subscribeResponse": {
        "subscriptionId": "id_subscription",
        "id_parameter": "value_parameter"
    }
}
```

delete subscription v1

```
http://{{url}}:{{port-orion}}/v1/contextSubscriptions/{{fiware-subscription}}
```

Definition: Method to delete the subscription to an entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-subscription: ID of subscription to remove.

Return:

```
{
    "subscribeResponse": {
        "subscriptionId": "id_subscription",
        "id_parameter": "value_parameter"
    }
}
```

Entities v2

create entity v2

`http://{{url}}:{{port-orion}}/v2/entities/`

Definition: Method to put an entity in a service. For a successful creation must be insert in the body section the ID entity, type of the entity and all attributes of the entity.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Body:

```
{
  "id": "entity_id",
  "type": "entity_type",
   "attribute_id": {
            "type": "type_attribute",
            "value": "value_attribute",
            "metadata": {}
        }
    …
}
```

Return:

Can return an error message, if already exists an entity in the service with the same id.

---

retrieve entity v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}`

Definition: Method to obtain an entity of a service.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be retrieve from the service

Return:

If the search has been successful then the entity is retrieved and shows all their characteristics.

Else an error message is returned.

retrieve entity as data model v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}`

Definition: Method to obtain an entity of a service with data model format.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: ID of the entity that will be retrieve from the service

Return:

If the search has been successful then the entity is retrieved and shows their characteristics of compressed way.

Else an error message is returned.

retrieve entities v2

`http://{{url}}:{{port-orion}}/v2/entities`

Definition: Method to retrieve all entities of a service.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Return:

All entities of the service and for each entity shows their characteristics.

retrieve entities as data model v2

`http://{{url}}:{{port-orion}}/v2/entities?options=keyValues`

Definition: Method to retrieve all entities of a service with data model format.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Return:

All entities of the service and for each entity shows their characteristics of compressed way.

update entity v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}/attrs`

Definition: Method to update some attributes of the entity. For an update successful all attributes that going to be modified must be insert in body section.

Method: PATCH

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: id of the entity where their attributes will be updated

Body:

```
{
   "attribute_id": {
            "type": "type_attribute",
            "value": "value_attribute",
        }
    …
}
```

Return:

All entities of the service and for each entity shows their characteristics of compressed way.

---

delete entity v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}`

Definition: Method to delete an entity from a service.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: id of the entity that will be deleted.

---

create attribute for entity v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}/attrs`

Definition: Method to add a new attribute to the entity. For add a new attribute to the entity must insert the new attribute/s in section body.

Method: POST

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: id of the entity that will have new attributes.

Body:

```
{
   "attribute_id": {
            "type": "type_attribute",
            "value": "value_attribute",
        }
    …
}
```

---

retrieve entity attribute v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}/attrs/{{fiware-attr}}`

Definition: Method to retrieve an attribute of an entity.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: Id of the entity where is the attribute.
- fiware-attr: Id of the attribute.

Return:

If the search has been successful then the attribute is retrieved and shows their characteristics.
Else an error message is returned.

---

retrieve entity attribute as data model v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}/attrs/{{fiware-attr}}?options=keyValues`

Definition: Method to retrieve an attribute of an entity.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: Id of the entity where is the attribute.
- fiware-attr: Id of the attribute.

Return:

If the search has been successful then the attribute is retrieved and shows their characteristics of compressed way.
Else an error message is returned.

---

delete attribute for entity v2

`http://{{url}}:{{port-orion}}/v2/entities/{{fiware-entity}}/attrs/{{fiware-attr}}`

Definition: Method to delete an attribute of an entity.

Method: DELETE

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.
- fiware-entity: id of the entity that will delete attributes.
- fiware-attr: id of the attribute that will delete.

Types

retrieve type v2

`http://{{url}}:{{port-orion}}/v2/types/{{fiware-type}}`

Definition: Method to retrieve an entity type of the service.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Return:

If the search has been successful then type is retrieved and show all their characteristics.

Else an error message is returned.

---

retrieve types v2

`http://{{url}}:{{port-orion}}/v2/types/`

Definition: Method to retrieve all Entity types existing in the service.

Method: GET

Parameters:

- url: Link to the service that will be consulted.
- port-orion: Port to connect with the service.

Return:

The different Entity types in the service and their attributes defining their types

Subcriptions

create subscription v2

`http://{{url}}:{{port-orion}}/v2/subscriptions`

Definition: Method to create a subscription to one or many entities

Method: POST

Parameters

- url: Link to the service that will be consulted
- port-orion: port to connect with the service

Body

```
{
      “description”:  “Definition of the subscription”,
      “subject”: {
       “entinties”: [
         {
             “id”: “Room1”
             “type: “Room”
         }
       ],
       “condition”: {
            “attrs”: [
                 “pressure”
       ]
            }
      },
      “notification”: {
      “http”: {
           “url”: “http://localhost:1028/accumulate”
       },
       “attrs”: [
            “temperature”
       ]
      },
      “expires”:  “2040-01-01T14:00:00.00Z”,
     “throttling”: “5”
}
```

- condition: define the "trigger" for the subscription.
- url: URL where to send notifications
- Throttling: used to specify a minimum inter-notification arrival time.
- Notification.attr: Attributes which you will received in a notification when “condition.attr” changes.

---

retrieve subscriptions v2

`http://{{url}}:{{port-orion}}/v2/subscriptions`

Definition: Method to retrieve all subscriptions of a service.

Method: GET

Parameters:

- url: Link to the service that will be consulted
- port-orion: port to connect with the service

Return:

All existing subscriptions of the service and for each subscription show their information.

---

remove subscription v2

`http://{{url}}:{{port-orion}}/v2/subscriptions/{{fiware-subscription}}`

Definition: Method to delete a subscription of a service

Method: DELETE

Parameters:

- url: Link to the service that will be consulted
- port-orion: port to connect with the service
- fiware-subscription: id of the subscription will be deleted.