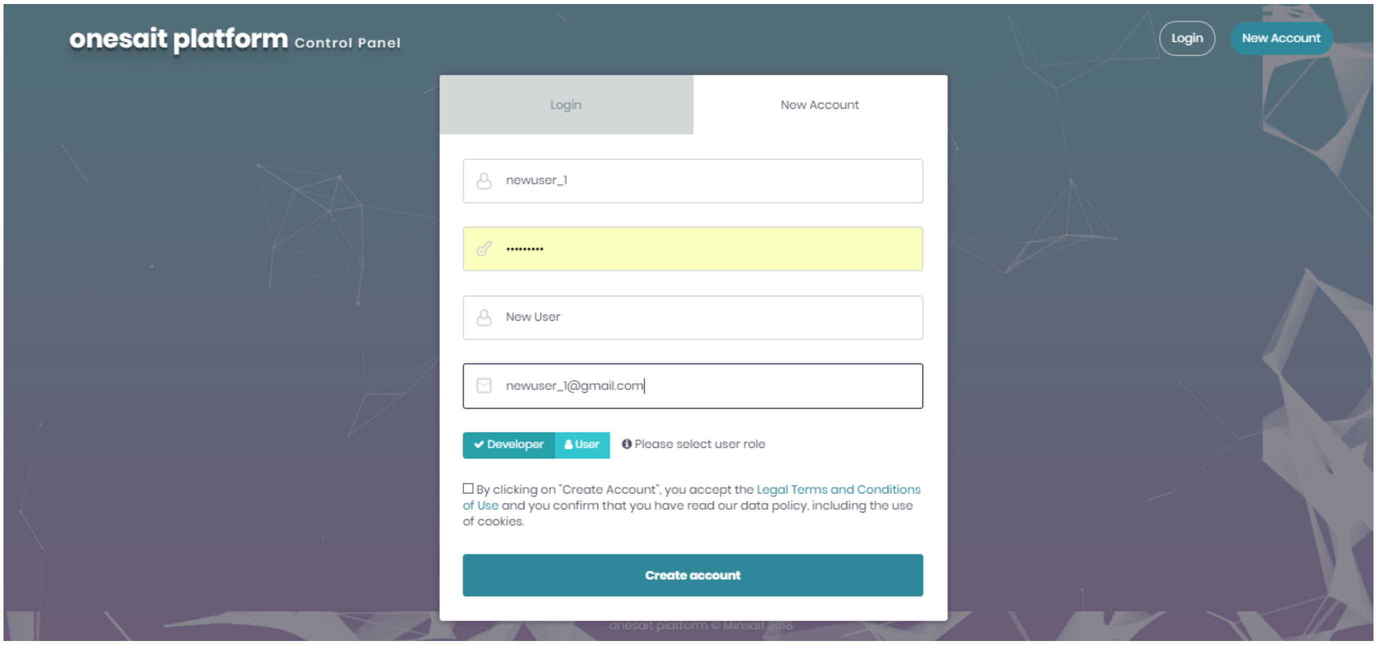
Getting Started with the onesait Cloud Platform

In this guide we are using onesait Platform CloudLab, this is a free installation of the platform offered in Cloud in which developers and potential users of the platform can probe the platform.

Access CloudLab

The main access is onesait Platform CloudLab (https://loadbalancerservice-onesait.ocp.52.233.186.149.nip.io/controlpanel)

First of all you have to create a User, in order to do it on the login page select **New Account** and enter id, password, name and mail for accesing the platform. If you want to accces the platform as a developer you should select **Developer** role:


onesait platform Control Panel
Login
New Account
Login
New Account
✓ Developer
User
Create account
onesait platform © Minsait 2018

newuser_1

•••••••••

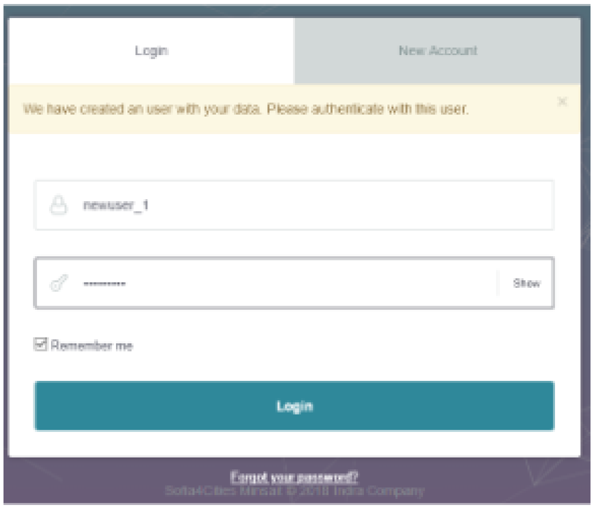
New User

newuser_1@gmail.com

Please select user role

□ By clicking on "Create Account", you accept the Legal Terms and Conditions of Use and you confirm that you have read our data policy, including the use of cookies.

Once you have created the account you can access with your user and password:



We have created an user with your data. Please authenticate with this user.

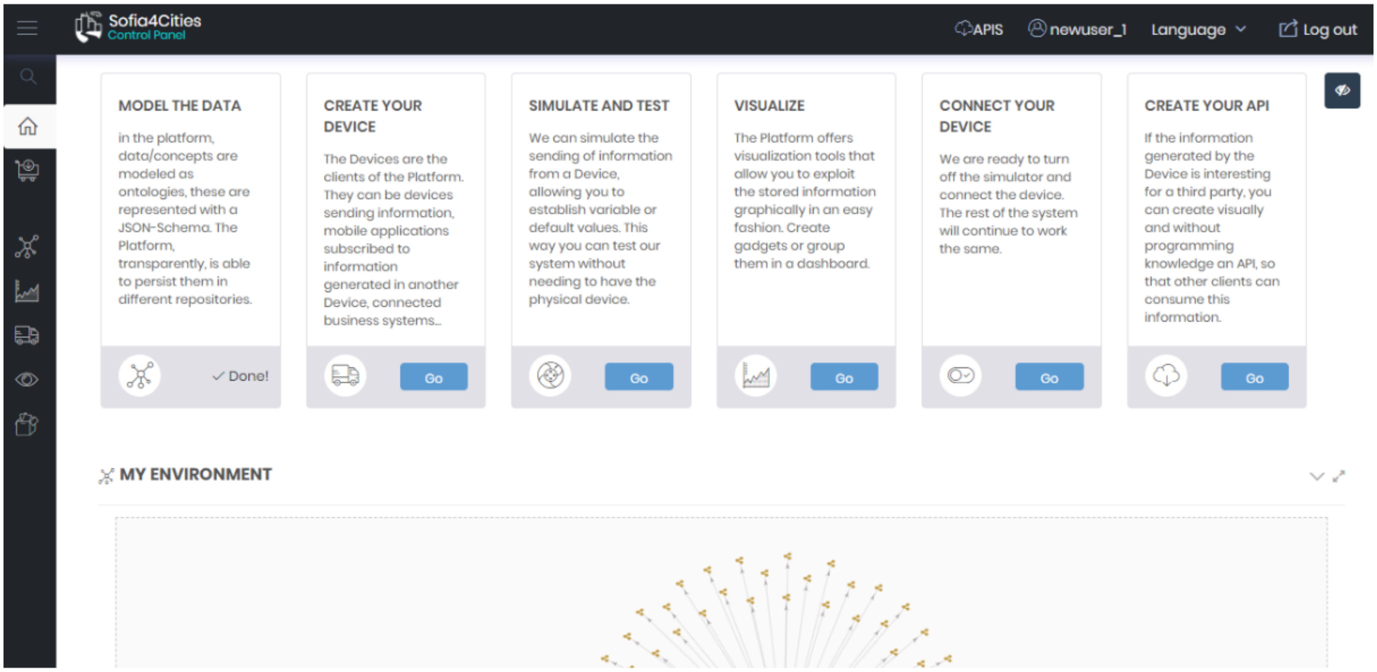
newuser_1

••••••••••

☑ Remember me

Main Page

When you access the ControlPanel with developer role you´ll see a page like this:



MODEL THE DATA

in the platform, data/concepts are modeled as ontologies, these are represented with a JSON-Schema. The Platform, transparently, is able to persist them in different repositories.

CREATE YOUR DEVICE

The Devices are the clients of the Platform. They can be devices sending information, mobile applications subscribed to information generated in another Device, connected business systems...

SIMULATE AND TEST

We can simulate the sending of information from a Device, allowing you to establish variable or default values. This way you can test our system without needing to have the physical device.

VISUALIZE

The Platform offers visualization tools that allow you to exploit the stored information graphically in an easy fashion. Create gadgets or group them in a dashboard.

CONNECT YOUR DEVICE

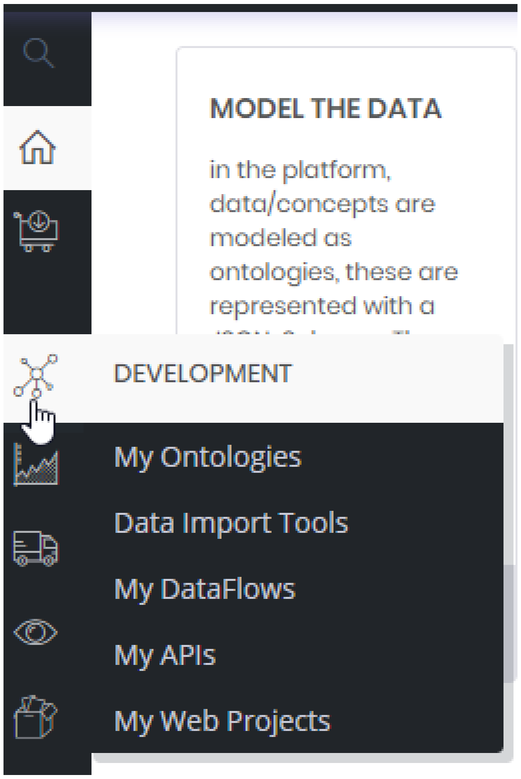
We are ready to turn off the simulator and connect the device. The rest of the system will continue to work the same.

CREATE YOUR API

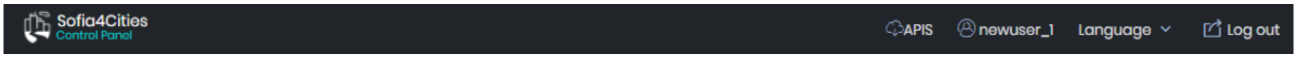
If the information generated by the Device is interesting for a third party, you can create visually and without programming knowledge an API, so that other clients can consume this information.

The main page show you:

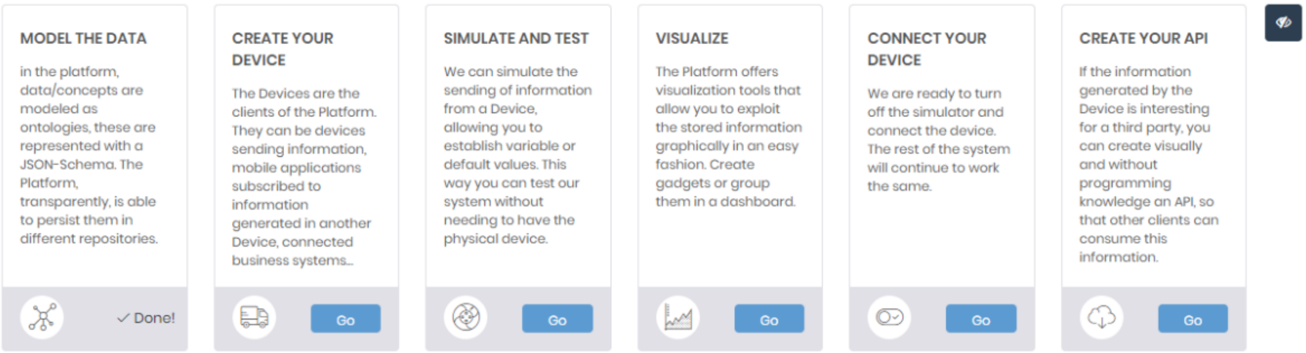
- A menu in the left side with all the options of the platform organized in submenus: SHARE&USE, DEVELOPMENT, VISUALIZATION, DEVICES, DIGITAL TWIN and TOOLS.



- A header in which you can view your token, access your profile, select the language of the application and log out of the system:



- a first widget that shows you the typical flow for creating apps with the platform, each box explain the action of it.



MODEL THE DATA

in the platform, data/concepts are modeled as ontologies, these are represented with a JSON-Schema. The Platform, transparently, is able to persist them in different repositories.

✓ Done!

CREATE YOUR DEVICE

The Devices are the clients of the Platform. They can be devices sending information, mobile applications subscribed to information generated in another Device, connected business systems...

Go

SIMULATE AND TEST

We can simulate the sending of information from a Device, allowing you to establish variable or default values. This way you can test our system without needing to have the physical device.

Go

VISUALIZE

The Platform offers visualization tools that allow you to exploit the stored information graphically in an easy fashion. Create gadgets or group them in a dashboard.

Go

CONNECT YOUR DEVICE

We are ready to turn off the simulator and connect the device. The rest of the system will continue to work the same.

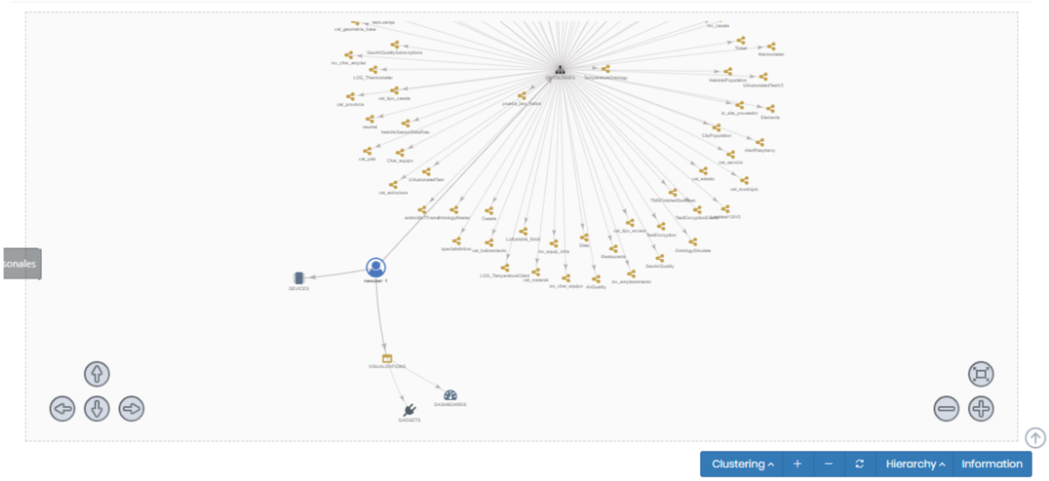
Go

CREATE YOUR API

If the information generated by the Device is interesting for a third party, you can create visually and without programming knowledge an API, so that other clients can consume this information.

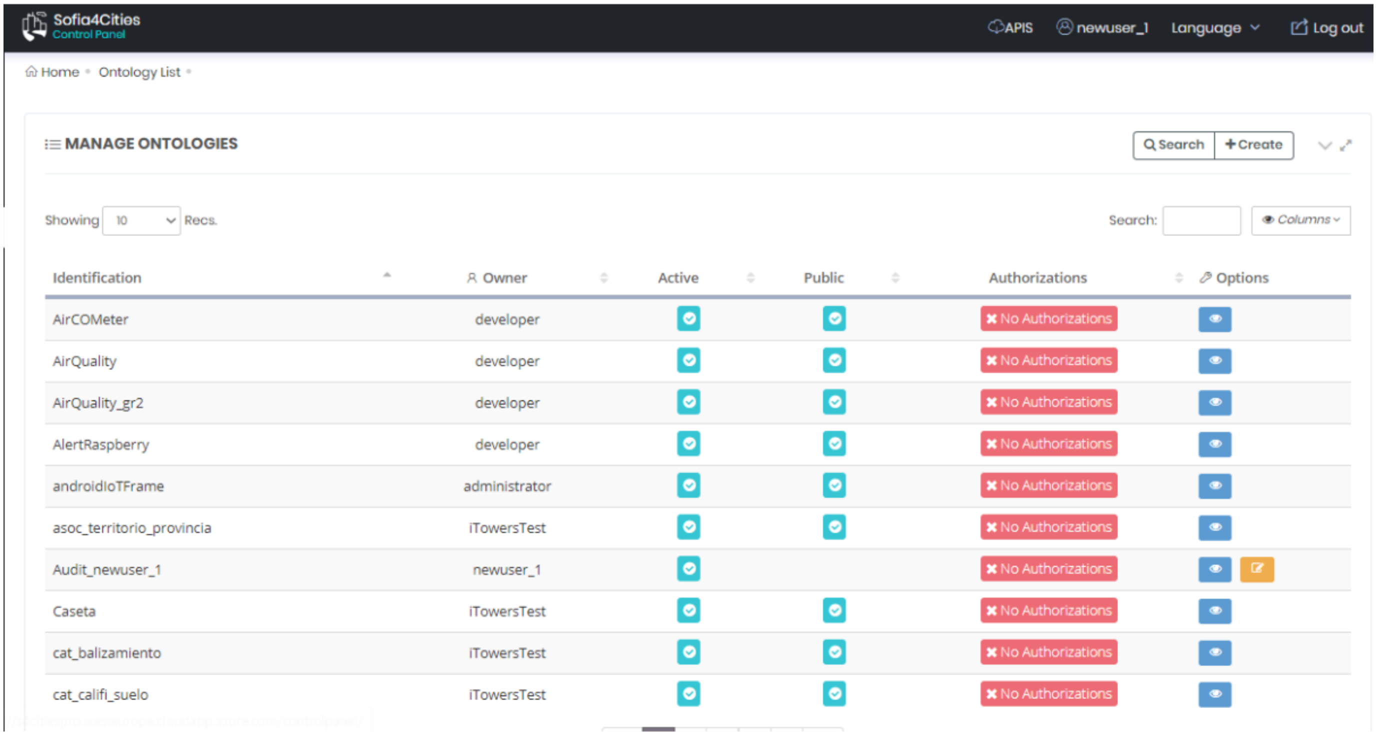
Go

- a second widget showing a graph of the components of the user. As some users have created public ontologies you can access them:



Creating an Ontology

The first step it to model the data that we want to use. To do that, go to the menú option **DEVELOPMENT -> My Ontologies**



MANAGE ONTOLOGIES

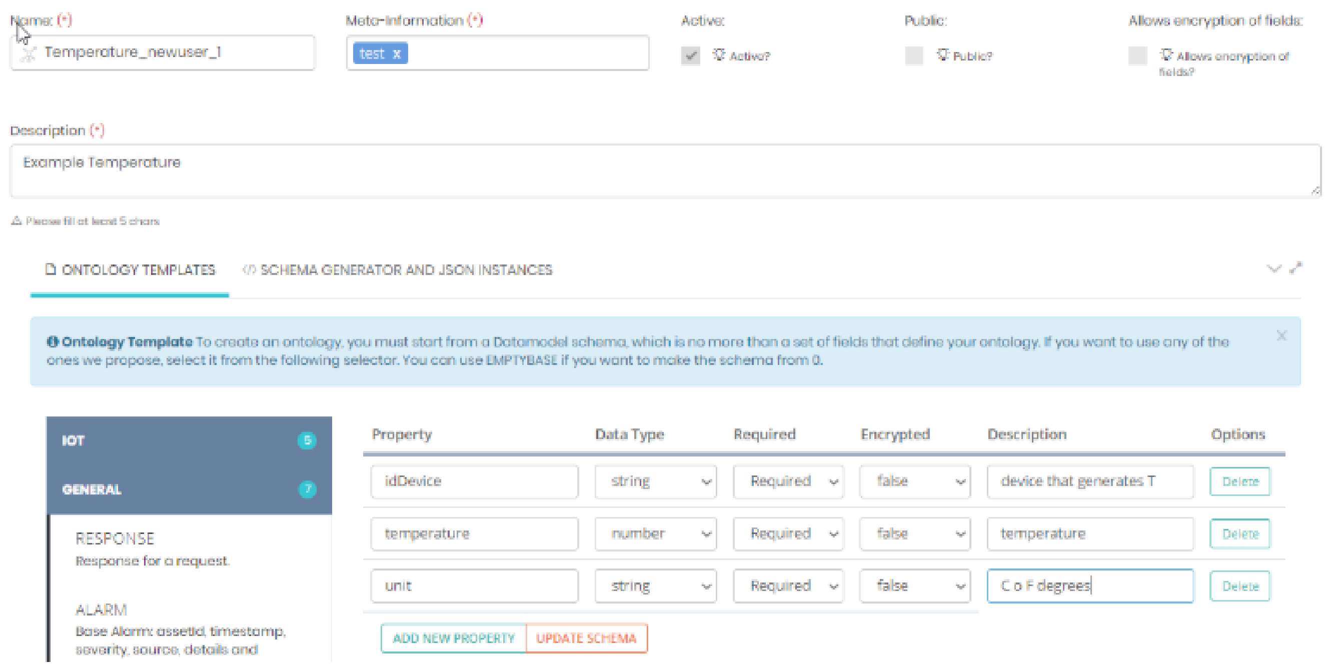
Showing 10 Recs. Search: Columns

Identification	Owner	Active	Public	Authorizations	Options
AirCOMeter	developer	✓	✓	✖ No Authorizations	👁
AirQuality	developer	✓	✓	✖ No Authorizations	👁
AirQuality_gr2	developer	✓	✓	✖ No Authorizations	👁
AlertRaspberry	developer	✓	✓	✖ No Authorizations	👁
androidIoTFrame	administrator	✓	✓	✖ No Authorizations	👁
asoc_territorio_provincia	iTowersTest	✓	✓	✖ No Authorizations	👁
Audit_newuser_1	newuser_1	✓		✖ No Authorizations	👁 ✎
Caseta	iTowersTest	✓	✓	✖ No Authorizations	👁
cat_balizamiento	iTowersTest	✓	✓	✖ No Authorizations	👁
cat_califi_suelo	iTowersTest	✓	✓	✖ No Authorizations	👁

This option shows the available ontologies. Use the **Create** option at the right upper corner to create a new Ontology.

An Ontology represents the data model of your data.

Complete the following information for your new Ontology:



Name: (*) Temperature_newuser_1

Meta-Information (*) test x

Active: ☑ Active?

Public: □ Public?

Allows encryption of fields: □ Allows encryption of fields?

Description (*) Example Temperature

⚠ Please fill at least 5 chars

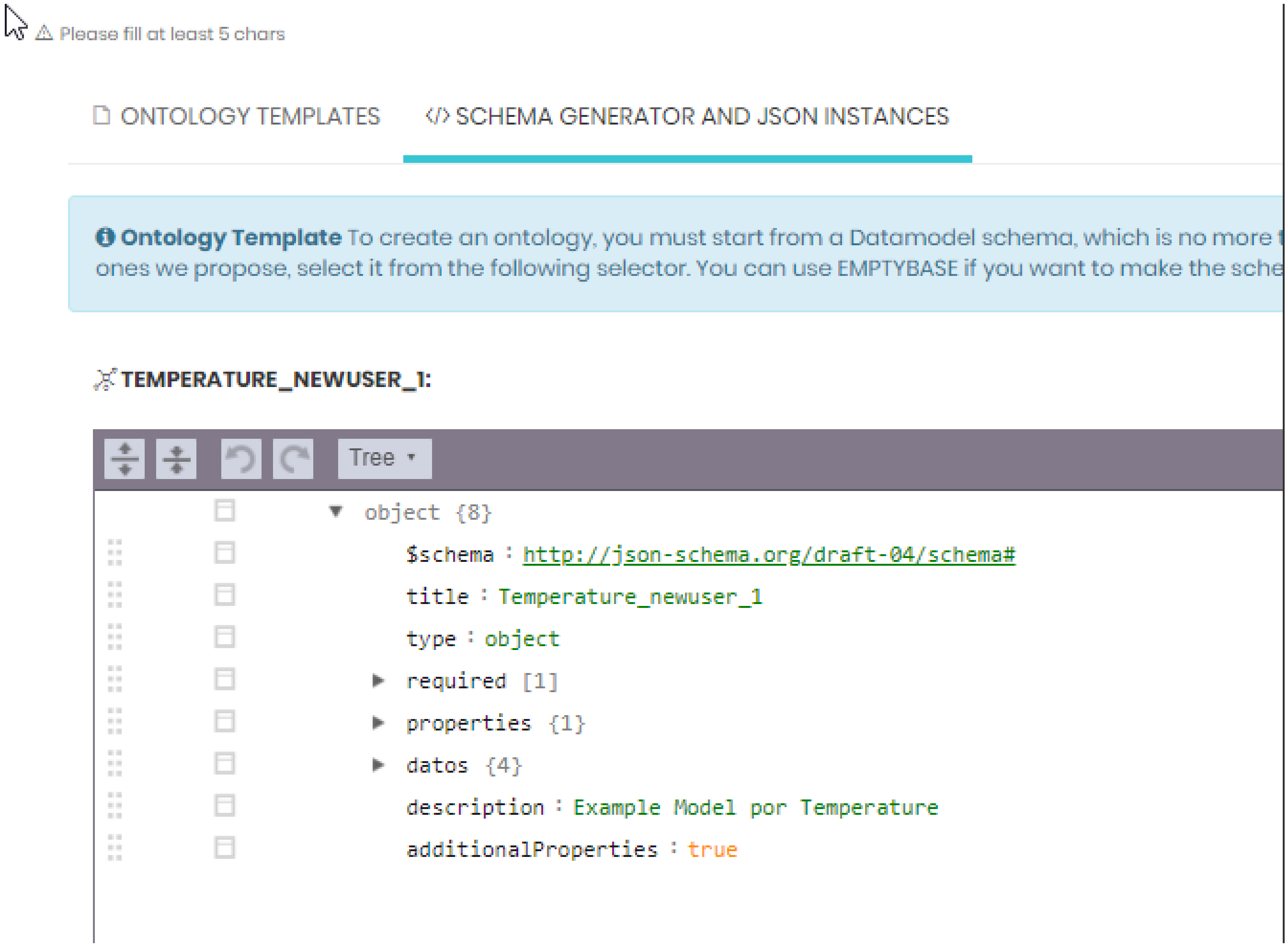
Ontology Template To create an ontology, you must start from a Datamodel schema, which is no more than a set of fields that define your ontology. If you want to use any of the ones we propose, select it from the following selector. You can use EMPTYBASE if you want to make the schema from 0.

IOT 5
GENERAL 7
RESPONSE
Response for a request.
ALARM
Base Alarm: assetId, timestamp, severity, source, details and

Property	Data Type	Required	Encrypted	Description	Options
idDevice	string	Required	false	device that generates T	Delete
temperature	number	Required	false	temperature	Delete
unit	string	Required	false	C o F degrees	Delete

- **Name**: provide a unique name of the Ontology. For example Temperature_newuser_1
- **Meta-Information**: insert keywords to classify Ontologies. Use commas to introduce several keywords.
- **Active**: indicate if the Ontology is active. It is not possible to use inactive Ontologies.
- **Public**: indicate if the Ontology is public. All users will be able to query the data stored in the public Ontologies.
- **Description**: provide a description that helps identify the goal of the Ontology.
- **Ontology Template**: select one of the available templates to create the ontology. The most basic template is General -> EmptyBase. After selecting the template, add all the properties that you need in the ontology using the ADD NEW PROPERTY button.

When you finish of adding properties use the **UPDATE SCHEMA** to add all the new properties to the Ontology. This shows the internal schema of the ontology (the platform validates all messages of this type with this schema):



⚠ Please fill at least 5 chars

Ontology Template To create an ontology, you must start from a Datamodel schema, which is no more t
ones we propose, select it from the following selector. You can use EMPTYBASE if you want to make the sche

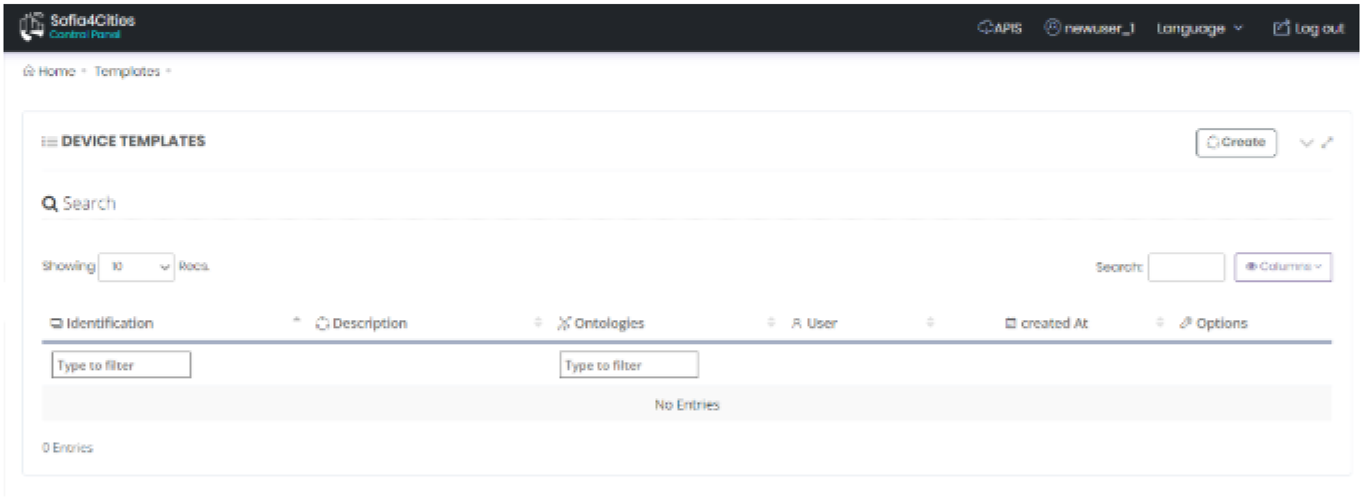
TEMPERATURE_NEWUSER_1:

```
▼ object {8}
    $schema : http://json-schema.org/draft-04/schema#
    title : Temperature_newuser_1
    type : object
  ▶ required [1]
  ▶ properties {1}
  ▶ datos {4}
    description : Example Model por Temperature
    additionalProperties : true
```

Finally, you save the Ontology using the New Button at the end of the page:

Creating a Device for the ontology

After defining the ontology it is possible to define Devices that work with the ontology. A **Device** is the representation of a physical device or system connected to the platform.
These Devices will generate or consume data. To do that, go to the D**EVICES -> Devices Templates** menu option.



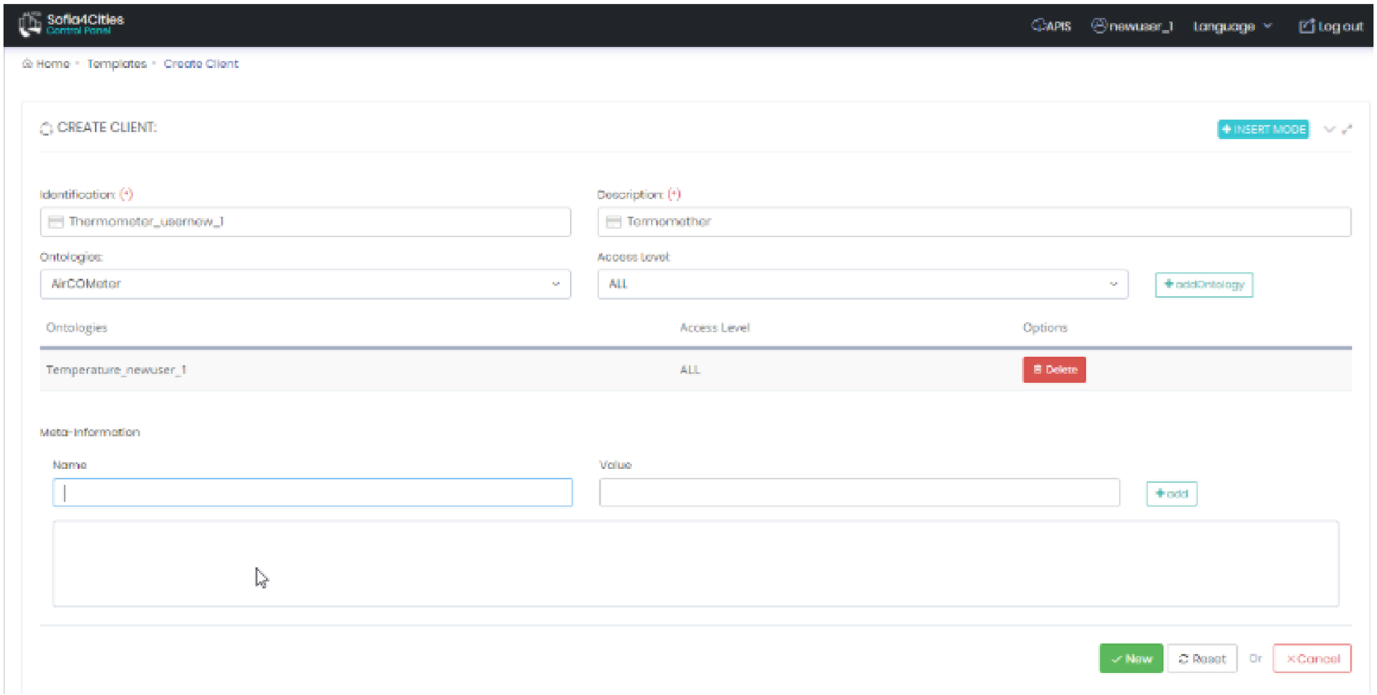
DEVICE TEMPLATES

Identification	Description	Ontologies	User	created At	Options
No Entries					

0 Entries

Use the Create button placed in the right upper corner to define a new Device.

Complete the following information for the new Device:



CREATE CLIENT:

Identification: (*) Thermometer_usernew_1

Description: (*) Termomether

Ontologies: AirCOMeter

Access Level: ALL

Ontologies	Access Level	Options
Temperature_newuser_1	ALL	Delete

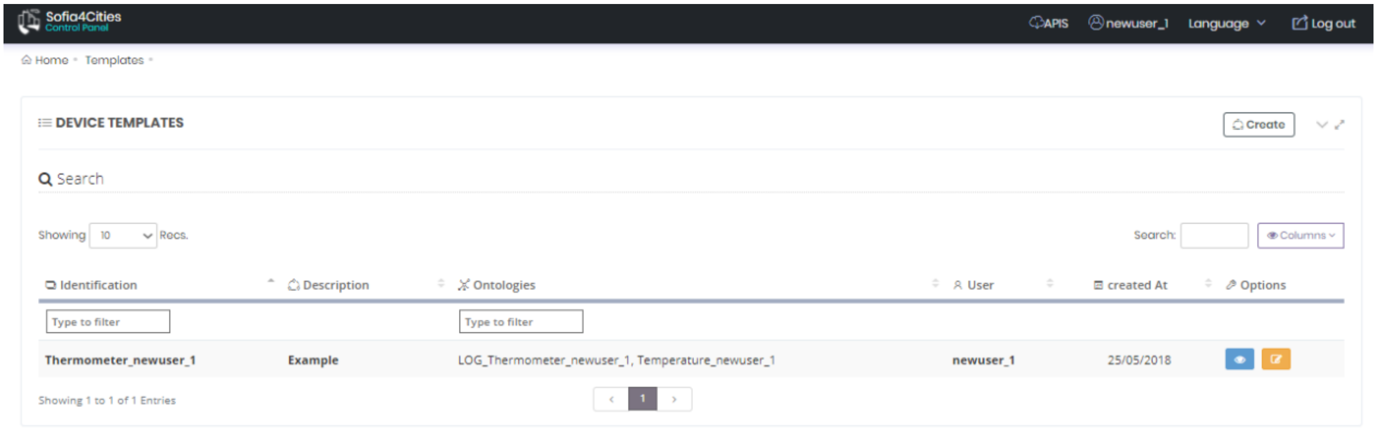
Meta-Information

Name

Value

- **Identification**: provide a unique name for the Device. For example Thermometer_newuser_1
- **Description**: add a description for the Device.
- **Ontologies**: it is possible to configure access to several ontologies. In this case we select the previously created ontology Temperature_newuser_1
    - **Ontology**: ontology identification.
    - **Access Level:** the type of access configured.
- **Meta Information**: Additional information about the device. The information will be added as key values pairs.
    - **name**: the name of the key.
    - **value**: the value of the property.

Now you can save your Device with **New** button. In the list of Devices you can view:



DEVICE TEMPLATES

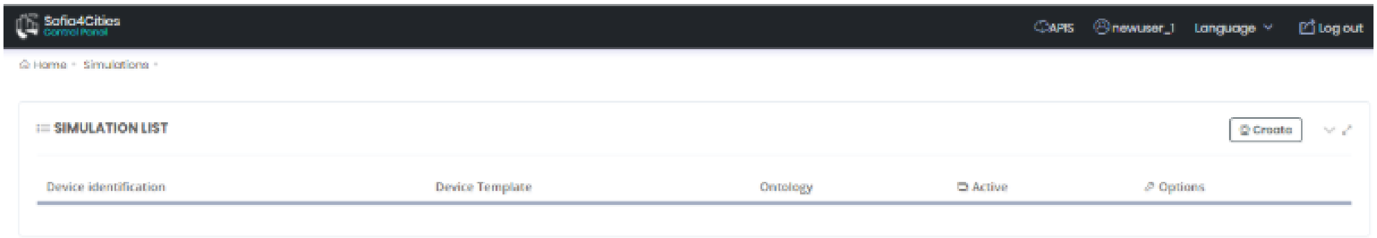
Showing 10 Recs.

Identification	Description	Ontologies	User	created At	Options
Thermometer_newuser_1	Example	LOG_Thermometer_newuser_1, Temperature_newuser_1	newuser_1	25/05/2018	

Showing 1 to 1 of 1 Entries

Creating a Simulator for the device

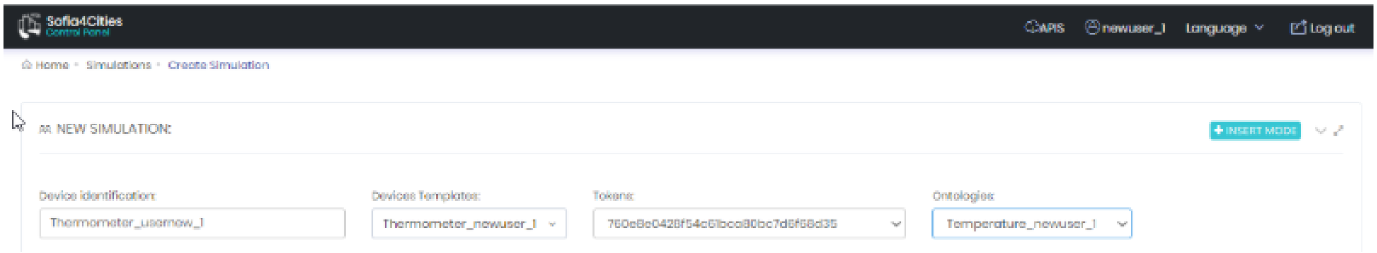
It is possible de define Device Simulators, this concept simulates a device sending information (Ontology instances) to the platform.
To do that, go to the **DEVICES -> Devices Simulator** menu option and use the **Create button placed in the right upper corner to define a new Device Simulator.



SIMULATION LIST

Device identification	Device Template	Ontology	Active	Options

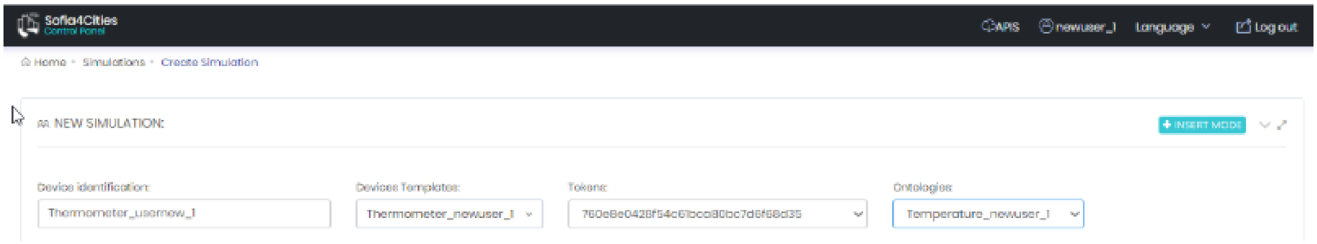
Complete the following information for the new Device Simulator:



NEW SIMULATION:

Device identification: Thermometer_usernew_1
Devices Templates: Thermometer_newuser_1
Tokens: 760e8e0428f54c61bca80bc7d6f58d35
Ontologies: Temperature_newuser_1

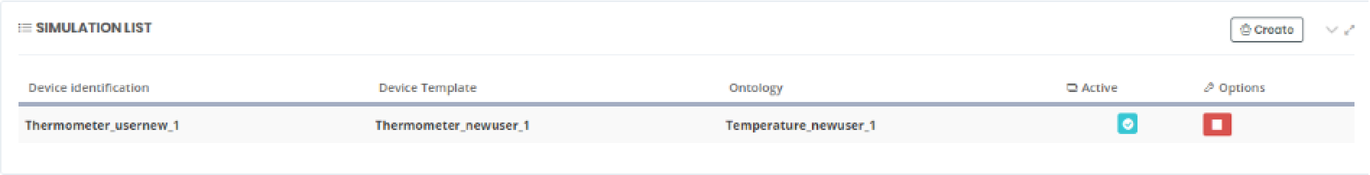
- **Device Identification**: Choose a unique name for the Device Simulator. For example ** Thermometer_usernew_1
- **Device Template:** choose the available *Device Thermometer_newuser_1 *
- **Token**: choose one of the tokens defined in the Device.
- **Ontology**: choose one of the ontologies in which the device has insert authorization. In this case Temperature_newuser_1
- **Time between inserts**: indicate the time in seconds between each insert in the ontology. For example 3
- **Ontology Properties**: fulfil the information required to generate values for each field of the ontology.



NEW SIMULATION:

Device identification: Thermometer_usernew_1
Devices Templates: Thermometer_newuser_1
Tokens: 760e8e0428f54c61bca80bc7d6f58d35
Ontologies: Temperature_newuser_1

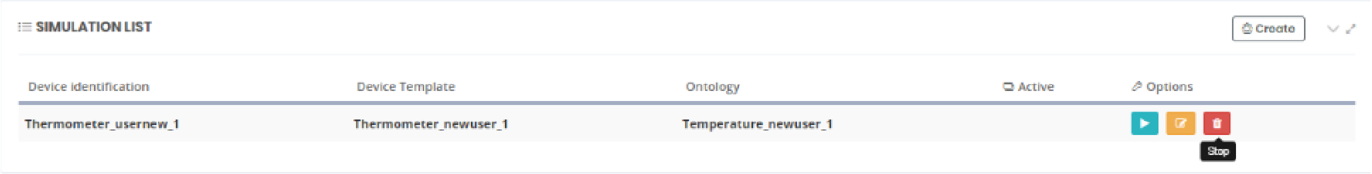
Finally, it is necessary to save the simulator by using the New button:



Device identification	Device Template	Ontology	Active	Options
Thermometer_usernew_1	Thermometer_newuser_1	Temperature_newuser_1	✓	■

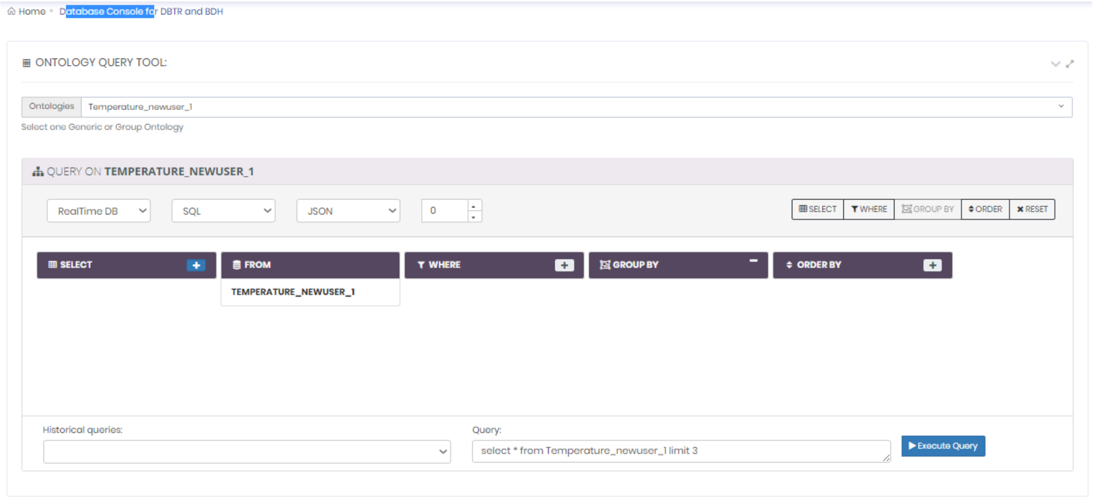
When you have created the simulator automatically it starts to generate data as you can see in the list:

You can stop in the STOP button and then you´ll see:



Device identification	Device Template	Ontology	Active	Options
Thermometer_usernew_1	Thermometer_newuser_1	Temperature_newuser_1		▶ ✎ 🗑

To verify that the device simulated has inserted instances of the defines ontology we can go to the menu **TOOLS -> Query Tools** and select our ontology:



Ontologies: Temperature_newuser_1

Select one Generic or Group Ontology

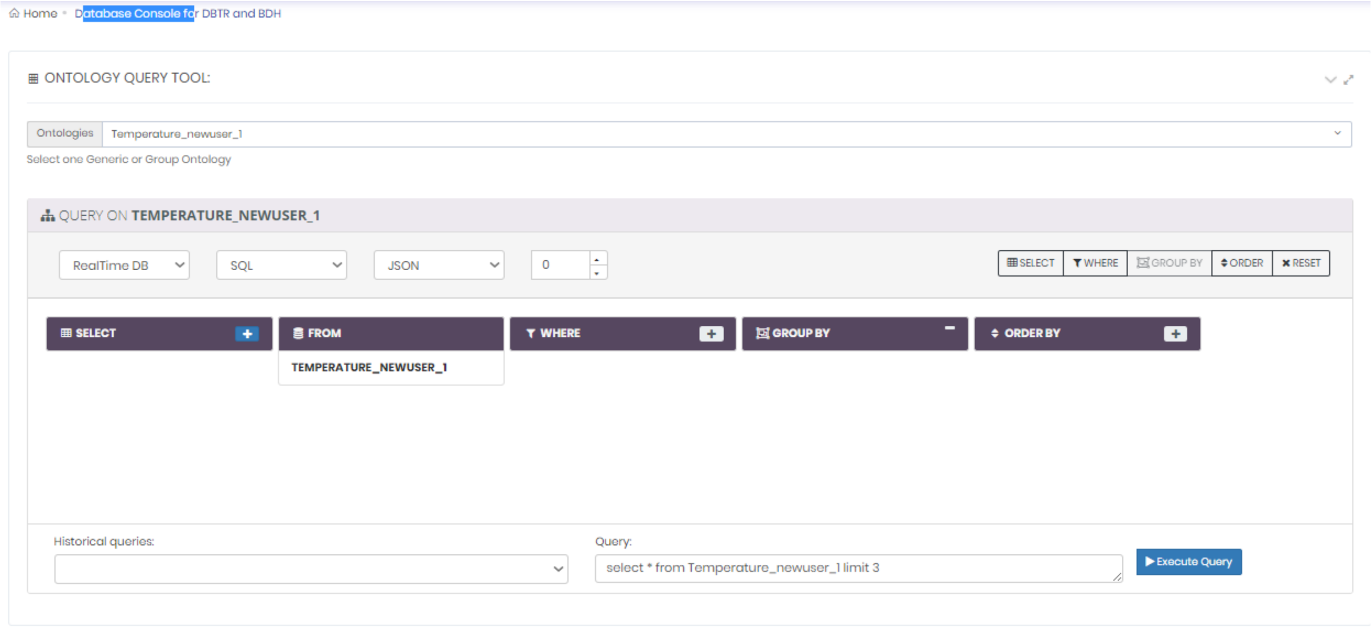
RealTime DB | SQL | JSON | 0

FROM: TEMPERATURE_NEWUSER_1

Historical queries:

Query: select * from Temperature_newuser_1 limit 3

When we pulse the Execute Query button we can see some instances of my ontology:



Ontologies: Temperature_newuser_1

Select one Generic or Group Ontology

RealTime DB | SQL | JSON | 0

FROM: TEMPERATURE_NEWUSER_1

Historical queries:

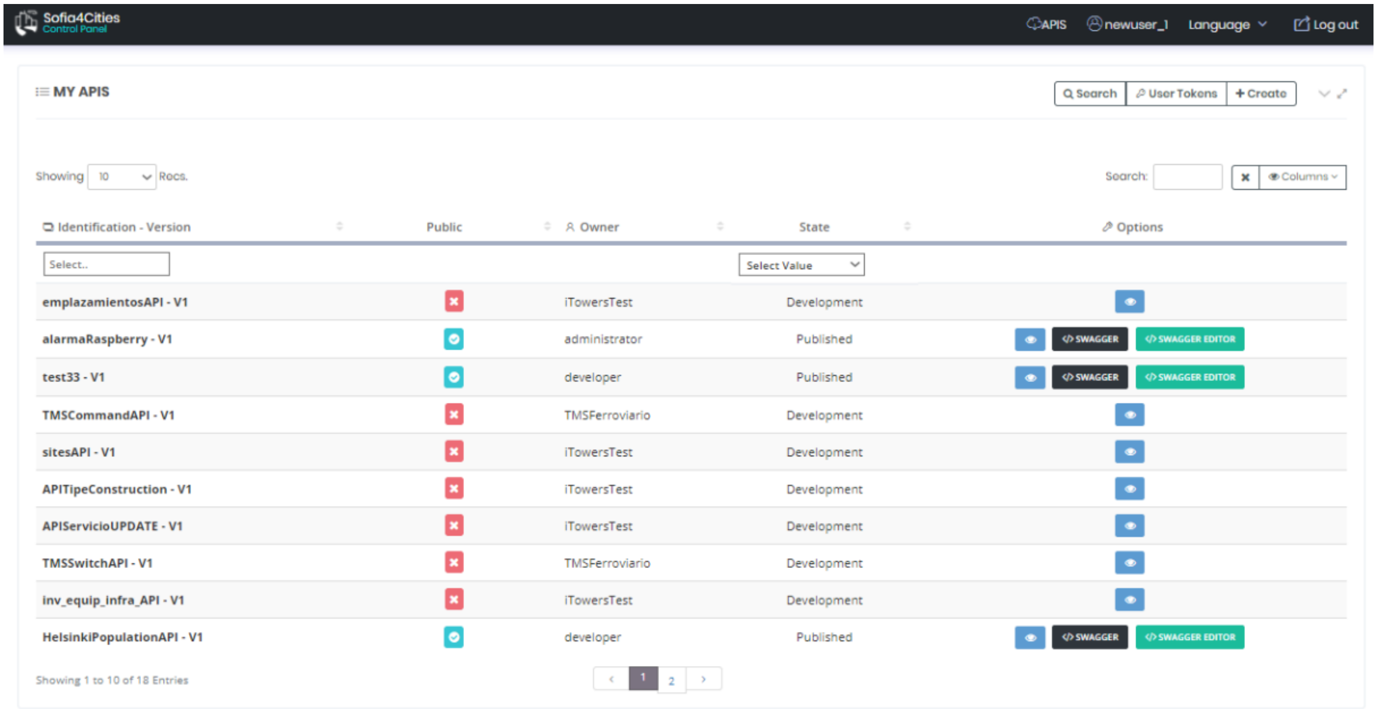
Query: select * from Temperature_newuser_1 limit 3

Creating a Dashboard

TODO: We do it after the review of the functionality of this module: Antona Díaz, Pedro Luis

Creating an API REST to publish the ontology

It is possible to publish operations over your Ontologies using creating REST APIs. In order to create an API go to the menú back to the Control Panel web application and use the **DEVELOPMENT -> My APIS** menu option.



Showing 10 Recs.

Search:

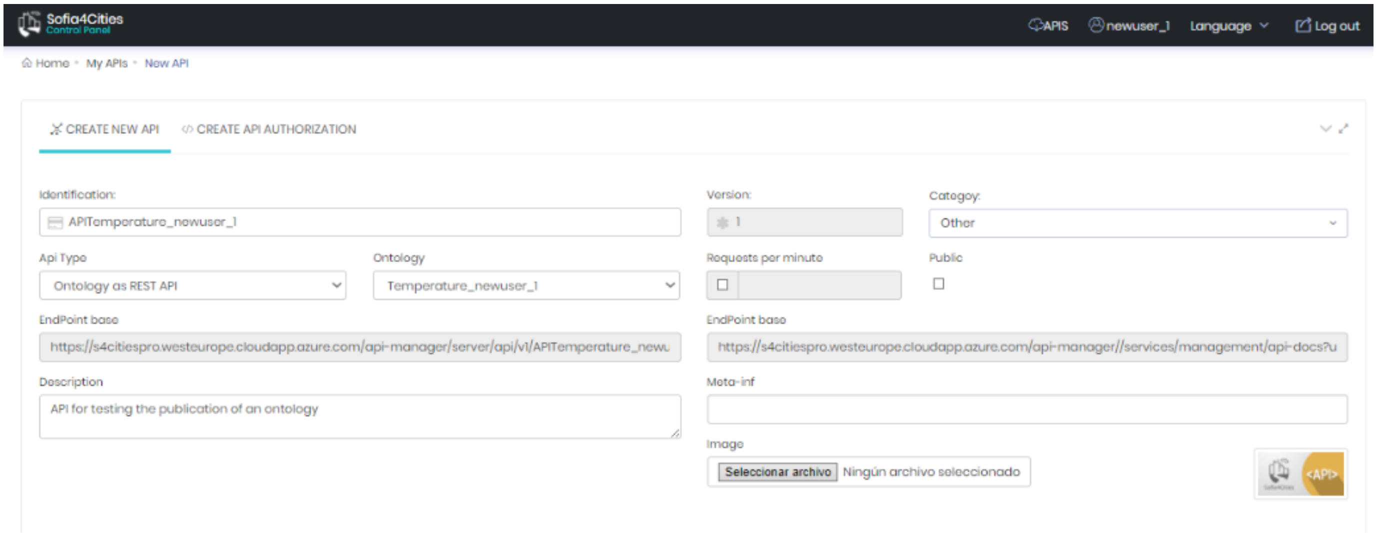
Identification - Version	Public	Owner	State	Options
Select..			Select Value	
emplazamientosAPI - V1	☒	iTowersTest	Development	👁
alarmaRaspberry - V1	✓	administrator	Published	👁 SWAGGER SWAGGER EDITOR
test33 - V1	✓	developer	Published	👁 SWAGGER SWAGGER EDITOR
TMSCommandAPI - V1	☒	TMSFerroviario	Development	👁
sitesAPI - V1	☒	iTowersTest	Development	👁
APITipeConstruction - V1	☒	iTowersTest	Development	👁
APIServicioUPDATE - V1	☒	iTowersTest	Development	👁
TMSSwitchAPI - V1	☒	TMSFerroviario	Development	👁
inv_equip_infra_API - V1	☒	iTowersTest	Development	👁
HelsinkiPopulationAPI - V1	✓	developer	Published	👁 SWAGGER SWAGGER EDITOR

Showing 1 to 10 of 18 Entries

‹ 1 2 ›

In the list you can see APIS published as Public for other users.

Use the **Create** button. You have to provide several information to create the API:



Identification: APITemperature_newuser_1

Version: 1

Category: Other

Api Type: Ontology as REST API

Ontology: Temperature_newuser_1

Requests per minute: □

Public: □

EndPoint base: https://s4citiespro.westeurope.cloudapp.azure.com/api-manager/server/api/v1/APITemperature_newu

EndPoint base: https://s4citiespro.westeurope.cloudapp.azure.com/api-manager//services/management/api-docs?u

Description: API for testing the publication of an ontology

Meta-inf:

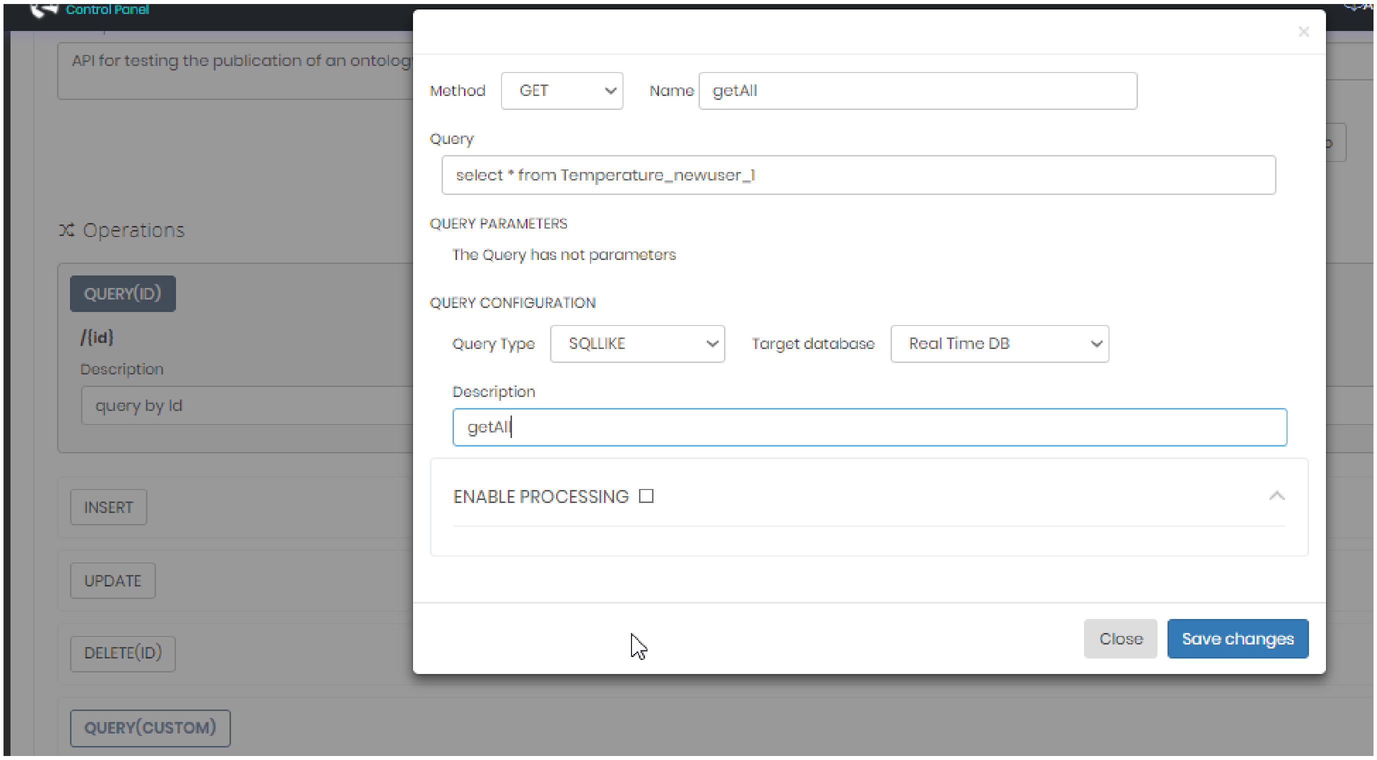
Image: Seleccionar archivo Ningún archivo seleccionado

- **Identification**: select a unique name for your API. For example APITemperature_newuser_1
- **API type:** currently only expose an Ontology as API REST is available.
- **Ontology**: choose the Ontology used in the API. In our case Temperature_newuser_1,

- **Description**: provide a description for the API.
- **Category**: choose the category that better match your API.
- **Public**: if marked, this API will be visible for all the users and all the user will be able to use it.
- **Meta-inf**: provide the meta information that you want for your API.
- **Image**: if you want to use an image for the representation of your API.

After this steps you see Operationsm then you have to choose the operations that you want to expose in your API. There are several pre created operations. Additionally, you can provide any other operation based on queries.

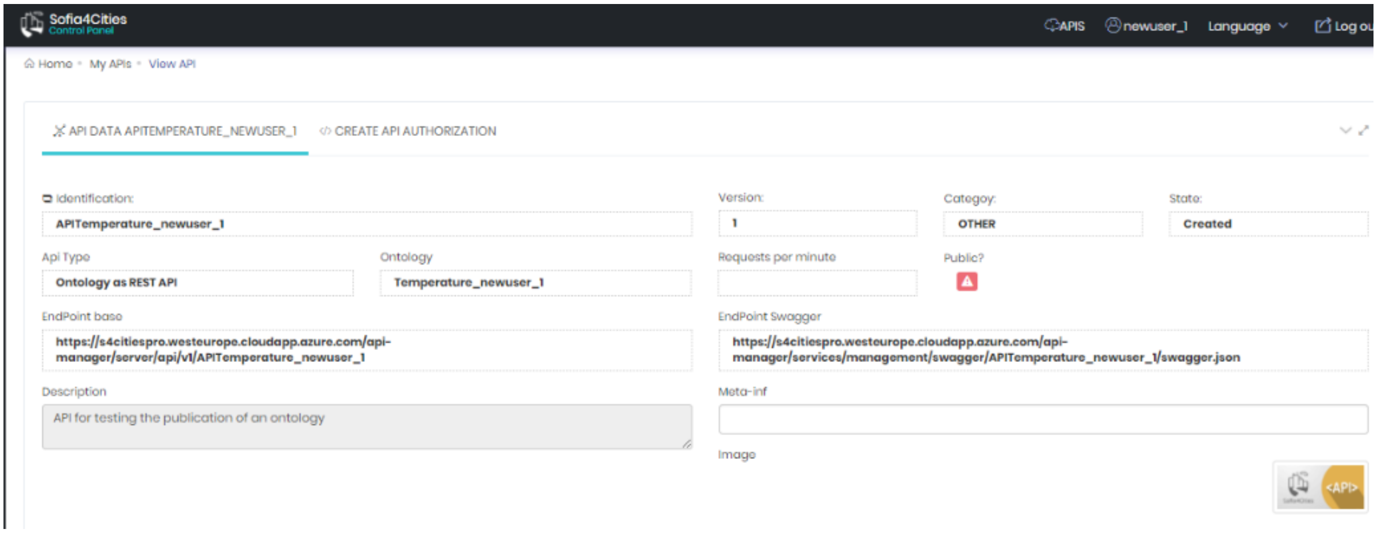
- **QUERY(ID)**: if selected, it allows to use get data by id operation from the Ontology.
- **INSERT**: if selected, it allows to perform insert operations to the Ontology.
- **UPDATE**: if selected, it allows to perform update operations to the Ontology.
- **DELETE(ID):** if selected, it allows to perform delete operations by the id of the data stored in the Ontology.
- **QUERY CUSTOM**: this option allows to define all the required operations that could be required based on database queries. For example we select it and complete these information:



Method: GET
Name: getAll
Query: select * from Temperature_newuser_1
QUERY PARAMETERS
The Query has not parameters
QUERY CONFIGURATION
Query Type: SQLLIKE
Target database: Real Time DB
Description: getAll
ENABLE PROCESSING □

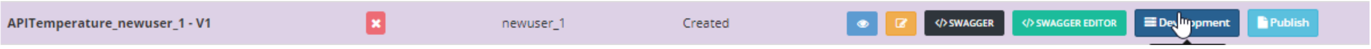
Then we selecte Save Changes.

Finally, use the New button to save the **API** data:



Identification: APITemperature_newuser_1
Version: 1
Categoy: OTHER
State: Created
Api Type: Ontology as REST API
Ontology: Temperature_newuser_1
Requests per minute:
Public?
EndPoint base: https://s4citiespro.westeurope.cloudapp.azure.com/api-manager/server/api/v1/APITemperature_newuser_1
EndPoint Swagger: https://s4citiespro.westeurope.cloudapp.azure.com/api-manager/services/management/swagger/APITemperature_newuser_1/swagger.json
Description: API for testing the publication of an ontology
Meta-inf:
Image

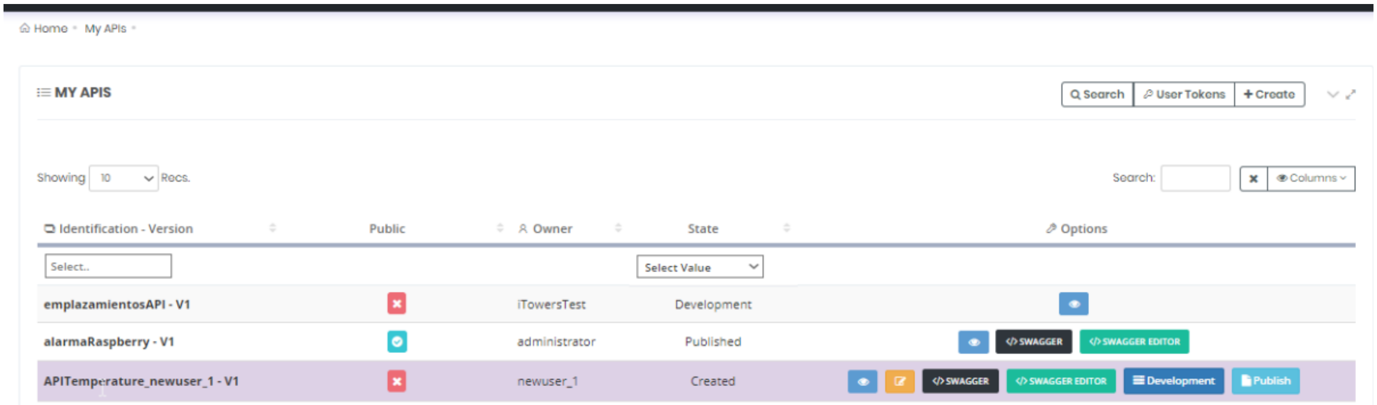
When we have our API finished then we can change its state, in order to do thar in the list of APIS we click the **Development** button:



APITemperature_newuser_1 - V1	✖	newuser_1	Created

Invoking the REST API

We return to the option My APIs and we can see the new API:



Home • My APIs •

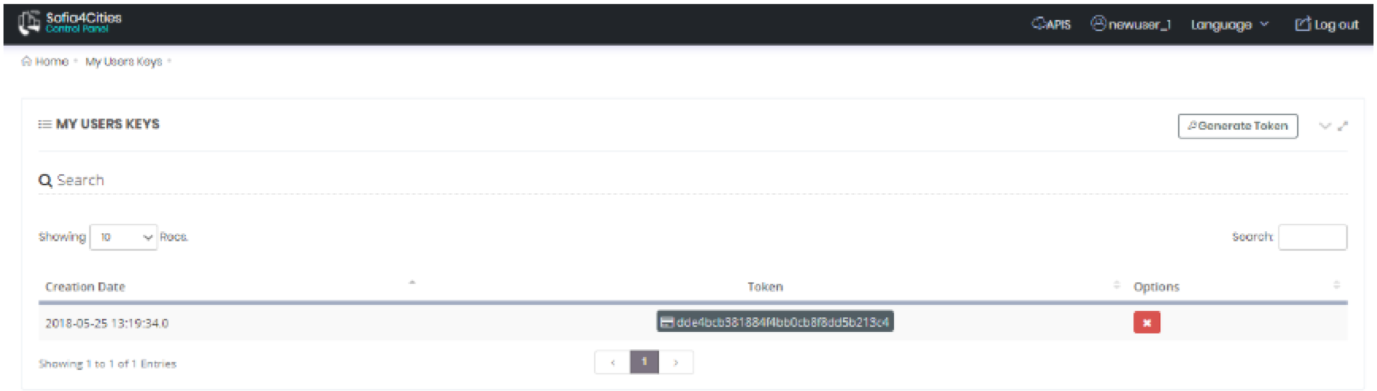
MY APIS

Showing 10 Recs.

Search:

Identification - Version	Public	Owner	State	Options
Select..			Select Value	
emplazamientosAPI - V1	✖	iTowersTest	Development	
alarmaRaspberry - V1	✓	administrator	Published	SWAGGER · SWAGGER EDITOR
APITemperature_newuser_1 - V1	✖	newuser_1	Created	SWAGGER · SWAGGER EDITOR · Development · Publish

In order to invoke the API we need a Token, you can access it in the section **User Tokens** on the upper side. We´ll see this screen:



Home • My Users Keys •

MY USERS KEYS

Search

Showing 10 Recs.

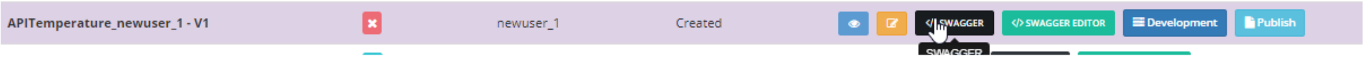
Search:

Creation Date	Token	Options
2018-05-25 13:19:34.0	dde4bcb381884f4bb0cb8f8dd5b213c4	✖

Showing 1 to 1 of 1 Entries

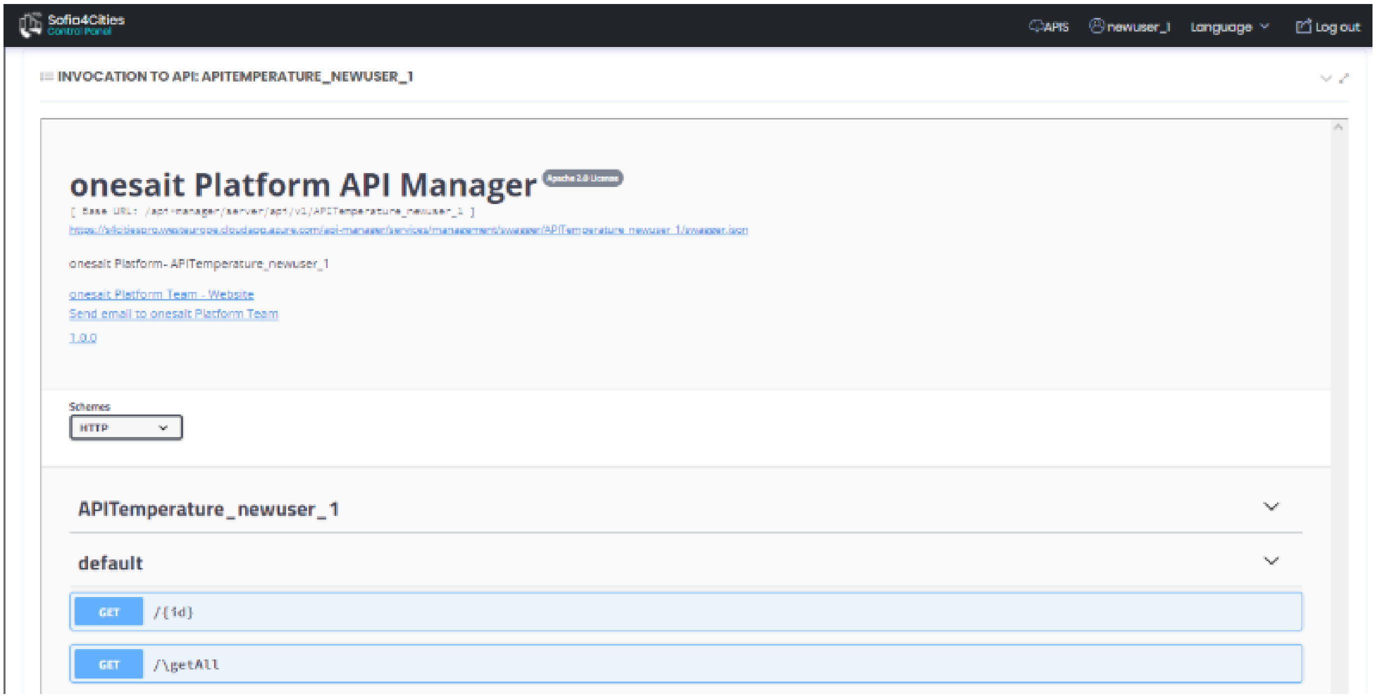
We copy the value of the token (in our case dde4bcb381884f4bb0cb8f8dd5b213)

To test the API we can use the integrated Swagger client, in order to do it in the list of APIs we select our API and click the SWAGGER button:



APITemperature_newuser_1 - V1	✖	newuser_1	Created

This shows you an user interface like this:



INVOCATION TO API: APITEMPERATURE_NEWUSER_1

onesait Platform API Manager Apache 2.0 License

[Base URL: /api-manager/server/api/v1/APITemperature_newuser_1]

https://s4citiespro.westeurope.cloudapp.azure.com/api-manager/services/management/swagger/APITemperature_newuser_1/swagger.json

onesait Platform- APITemperature_newuser_1

onesait Platform Team - Website

Send email to onesait Platform Team

1.0.0

Schemes: HTTP

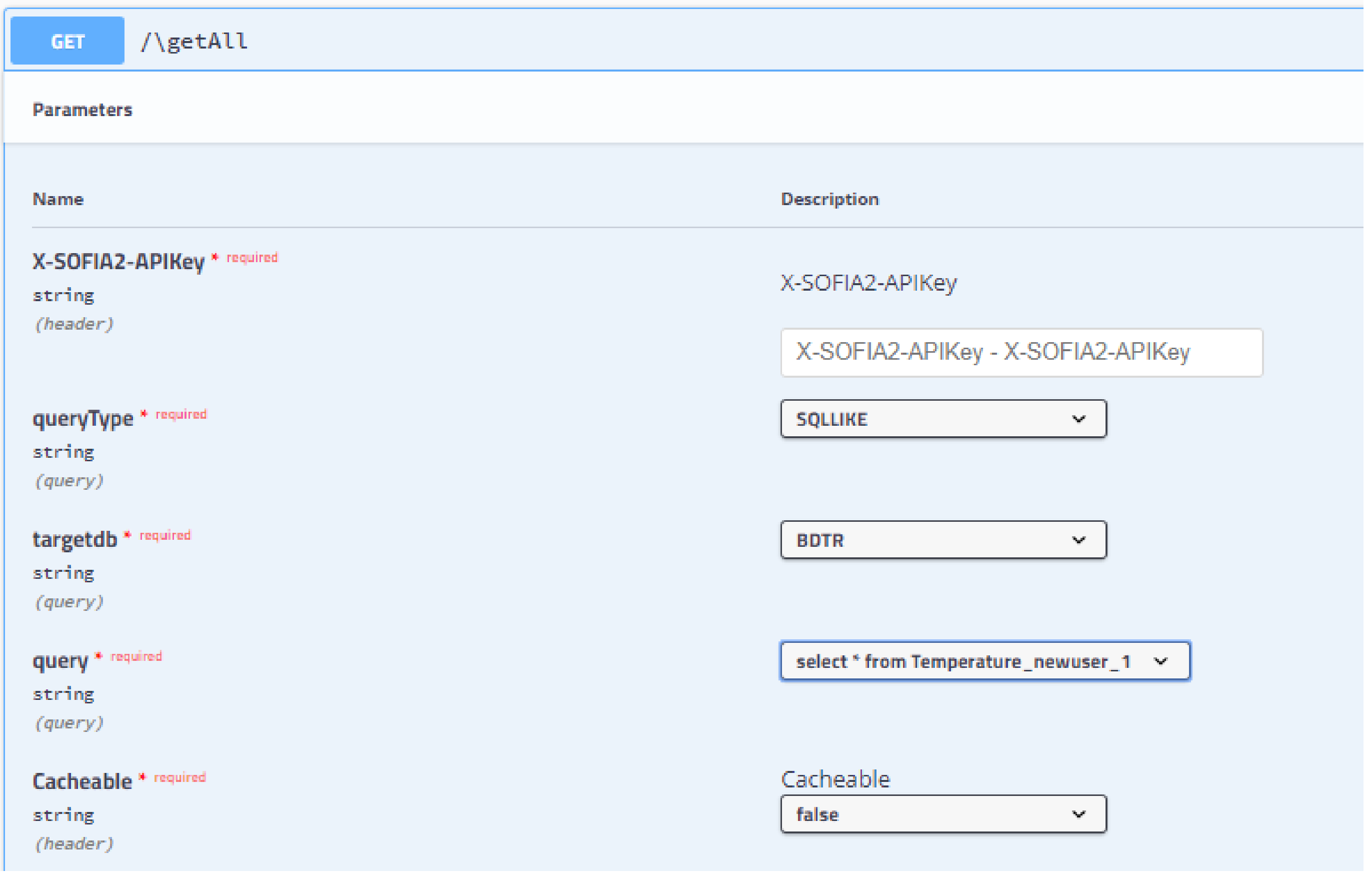
APITemperature_newuser_1

default

GET /{id}

GET /\getAll

You can click the section **/getAll** and then the Try It out



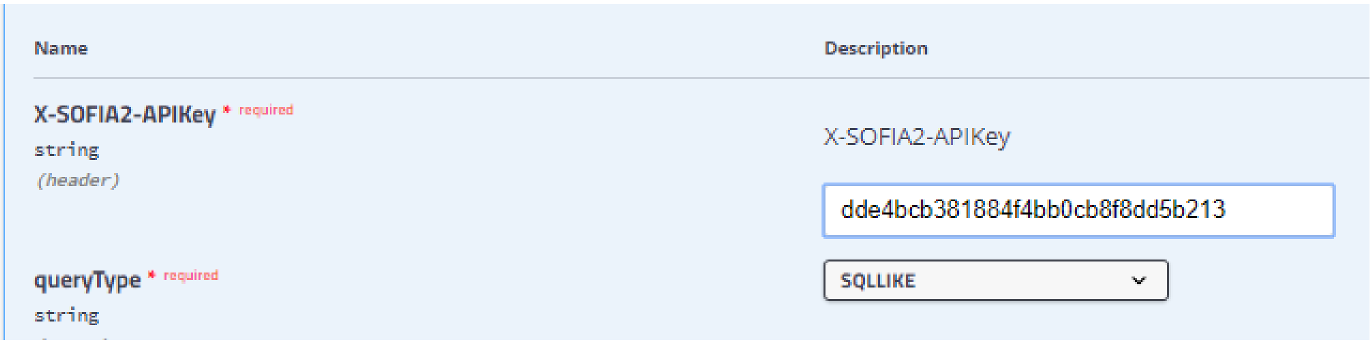
GET /\getAll

Parameters

Name	Description
X-SOFIA2-APIKey * required string *(header)*	X-SOFIA2-APIKey X-SOFIA2-APIKey - X-SOFIA2-APIKey
queryType * required string *(query)*	SQLLIKE
targetdb * required string *(query)*	BDTR
query * required string *(query)*	select * from Temperature_newuser_1
Cacheable * required string *(header)*	Cacheable false

This allows the form to complete data,

in our case we only have to fill the input **X-SOFIA2-APIKey** with our UserToken (dde4bcb381884f4bb0cb8f8dd5b213)



Name	Description
X-SOFIA2-APIKey * required string *(header)*	X-SOFIA2-APIKey dde4bcb381884f4bb0cb8f8dd5b213
queryType * required string	SQLLIKE

Now we click the **Execute button:**

The same test can be done with an external REST client, for instance Postman.
This is done by including in the header of the HTTP requests one parameter with key X-SOFIA2-APIKey and the token as value.