



FI content

Fusion Engine V0.3

USER & ADMINISTRATION GUIDE



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

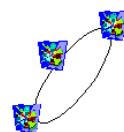


TABLE OF CONTENTS

TABLE OF CONTENTS.....	2
1 - INTRODUCTION	3
1.1 - Enablers. FICONTEXT and FIWARE.....	3
1.2 - Fusion Engine Definition and Contextualization.....	3
1.3 - Fusion Challenges.....	4
2 - ARCHITECTURE OF THE FUSION ENGINE.....	5
2.1 - FE Frontend.....	5
2.2 - FE service	5
3 - USER & ADMINISTRATION GUIDE.....	6
3.1 - FE Frontend.....	6
3.1.1 - Admin.....	6
3.1.1.1 - OCDs.....	7
3.1.1.2 - Configuration.....	9
3.1.2 - Docs.....	15
3.1.3 - Demo	16
3.1.4 - Swagger.....	18
3.1.4.1 - Categories.....	18
3.1.4.2 - POIS.....	20
3.2 - FE Service.....	22
3.3 - Fusion Rules.....	24
3.4 - How to quickly create an OCD	26

1 - INTRODUCTION

1.1 - Enablers. FICONTEXT and FIWARE

In order to strengthen a powerful European industry market for the upcoming years in the context of the Future Internet (FI), an FI-PPP called FIWARE was created (<http://www.fiware.org/>). The main outputs of this project are:

- To provide a reference platform for developing smart applications in an easy and efficient way.
- To provide a set of core (horizontal) components that build the core of this platform, upon which new (vertical) components can be built.

In order to further spread the usage of FIWARE in the European industry, new actions and use case projects were created, one of which was called FICONTEXT (<http://mediafi.org/>) , focussed on topics such as social connected TV, smart city guides and pervasive games.

The components developed in FIWARE are called Generic Enablers (GEs), as they are envisioned for a general (horizontal) market. On the other side, components developed in FIWARE use case projects such as FICONTEXT are called Specific Enablers (SEs), as they are primarily intended for special purposes (e.g. multimedia, TVs, databases, etc.).

All components (enablers) are just middleware that should facilitate big and small companies develop next generation Internet applications in order to be more competitive and/or productive. Some GEs and SEs are open source, and other can be used under specific terms of use defined by each project partner, but all of them have been built with the intention of providing helpful tools for SMEs.

1.2 - Fusion Engine Definition and Contextualization

The Fusion Engine (FE) is an open source specific enabler (SEs) developed in the framework of the FICONTEXT2 project. The specific area of interest of this enabler is Smart Cities, as it provides a tool to fusion data from different data providers to be used for further smart city applications.

The Fusion Engine is about data fusion for smart cities. In this context, the most useful piece of data for general purposes is called a POI (Point of Interest). Though POIs are mostly used for navigational GPS and touristic applications, it can be easily extended to any smart city environment you can imagine. Basically, a POI is something located at a certain position with certain properties that might be of interest while developing your application.

It is important to clarify what fusion means in the context of FE and why it might be helpful. With the appearance of the smart city concept, many cities (town council) are showing strong interest in providing local open data for their citizens. Other companies are also providing more and more open data repositories (e.g. for touristic and environmental purposes). There are also other global data repositories (e.g. OSM and DBpedia) that can act as additional data sources. So many data sources might be apparently helpful, but it typically results in POI replication and multiple connections to different data sources. Why not making this process offline before? One may wish to:

- Select only those data sources that are of interest
- Select only those categories from the sources that are of interest
- Get only one single merged POI from multiple ones
- Get only one single access interface to retrieve the resulting POIs

The result of the fusion generated by the FE is a georeferenced database of POIs. We called this Open City Database (OCD) as the input data (POIs) might typically come from open data. Any way you can also use

proprietary data if you comply with the corresponding license. This is much more a legal issue than a technological issue. In fact, the FE supports in its data model the multiple usage of licenses. It is up to the user to correctly (legally) exploit the data.

1.3 - Fusion Challenges

There are 3 technological challenges targeted by data fusion in general and FE in particular, as depicted in the following Figure 1 (see C in red).

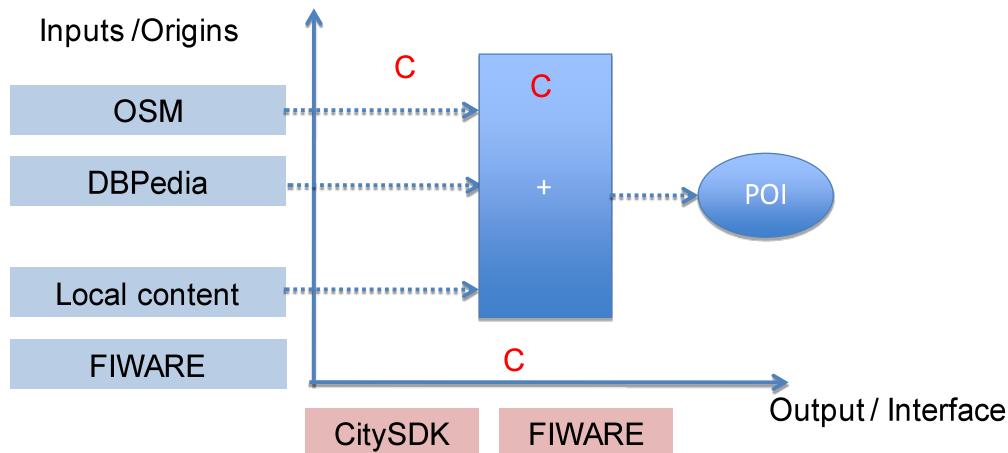


Figure 1. Fusion challenges in FiContent.

The first challenge relates to the input data. There are many data sources to be used:

- Global data sources, such as OSM and DBpedia
- Local data sources, such as the data portals of many cities (e.g. Valencia datos abiertos, opendatabcn, opendatacanarias, etc.)
- Particular data sources, such as FIWARE GEs

Note here that each data source has its own way (format and API) for providing the data. Besides, data categories are not necessarily preserved between data sources and are typically different.

The second challenge is how we are going to perform the fusion, the merging of POIs:

- How do we define that one POI from data source A and another POI from data source B refer to the same POI?
- When two or more POIs match, how do we create one single POI (e.g. which name and location do we preserve?)

The third challenge is about the output of the fusion. Even if it is a georeferenced POI repository, there should be a single API access. One would think in a standard way of defining POIs. Unfortunately, there is no such. Several years ago there was a first POI draft specification by the W3C (<http://www.w3.org/2010/POI/>), which later moved to OCG (<http://www.opengeospatial.org/projects/groups/poiswg>). However, the activity within this last group is frozen for the last two years. Thus, there were basically two alternatives:

- Define a new API format and API interface. This has been the approach within FIWARE
- Use a current POI specification that most resembles the POI draft, called CitySDK (<http://www.citysdk.eu/>).

2 - ARCHITECTURE OF THE FUSION ENGINE

The FE has two main components: (i) the frontend and (i) the service, as depicted in Figure 2.

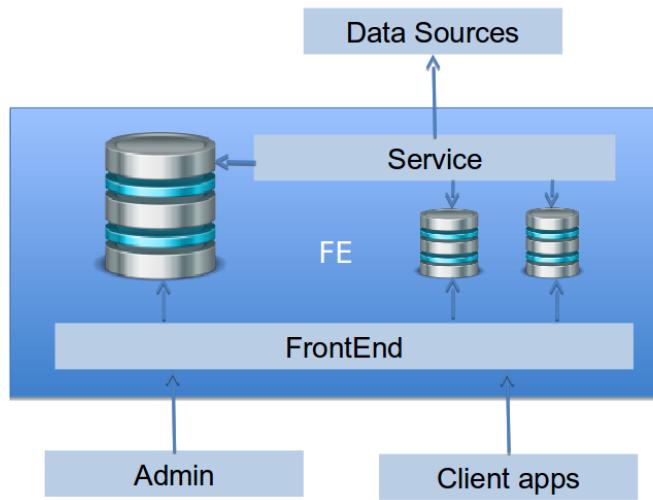


Figure 2. FE general architecture.

2.1 - FE Frontend

The FE frontend acts as interface with the outside world:

- Administrators access the frontend to set up the fusion engine and generate OCD instances
- Users and client applications access the frontend to request for POI data of a particular OCD instance

Note in Figure 2 that the FE has many different databases.

- There is a main database, called OCD_BASE, which indexes and configures all OCD instances. The data stored in this database is entered by the administrator.
- There are multiple databases, corresponding to each of the OCDs that are being built. This allows separating the information about POIs and securing data access. It also facilitates exporting operations of an OCD. Client apps access one of these OCD (small) databases through the frontend.

For client applications, there is a tiny difference with CitySDK interface. As the FE is able to generate as many OCDs as set up by the administrator, you also have to provide in the access API and 'ocdName' parameter, in order to select the city (OCD) you are interested in. The management of different databases allows introducing different access models and therefore business models. For example, an admin (SME) can offer free access to a basic OCD but pay-per-access to an enriched OCD.

2.2 - FE service

The FE service is a standalone process that reads the set up information in the OCD_BASE database and builds the different OCDs taking information from the corresponding (selected) data sources. This is an offline process that can take several minutes depending on the configuration (number of sources, number of categories, city size, etc.)

3 - USER & ADMINISTRATION GUIDE

This guide assumes you have already installed the FE Frontend and the FE service. Otherwise, please have a look at the Installation guide.

This guide also assumes that the user is an administrator and knows what he/she is doing. The FE is not intended for a final end user. Thus using and administering the FE refers currently to the same concept.

3.1 - FE Frontend

The main page (main menu) of the FE Frontend has 4 options (see Figure 3):

- *Admin* : This is by far the most important one as it involves the proper set up of an OCD
- *Docs*: This includes the documentation files (installation guide, user guide and admin manual), in PDF format, so that you can check all the available information online.
- *Demo*: This is just a basic demo tool that allows you to easily visualize POIs of any available OCD. A basic usage has been already described in the installation guide for quickly checking the installation.
- *Swagger*: this is a nice interface for developers that allow them to quickly check the implemented functions and the required parameters.



Figure 3. FE frontend main page

3.1.1 - Admin

The Admin menu is the most important are for the administrator. You have here basically two options:

- *Configure OCDs* : you can here list and create new OCDs.
- *Configuration* : you should here configure all related parameters the OCD relies on (data sources, categories, API types, etc.)



Figure 4. Admin main menu

3.1.1.1 - OCDs

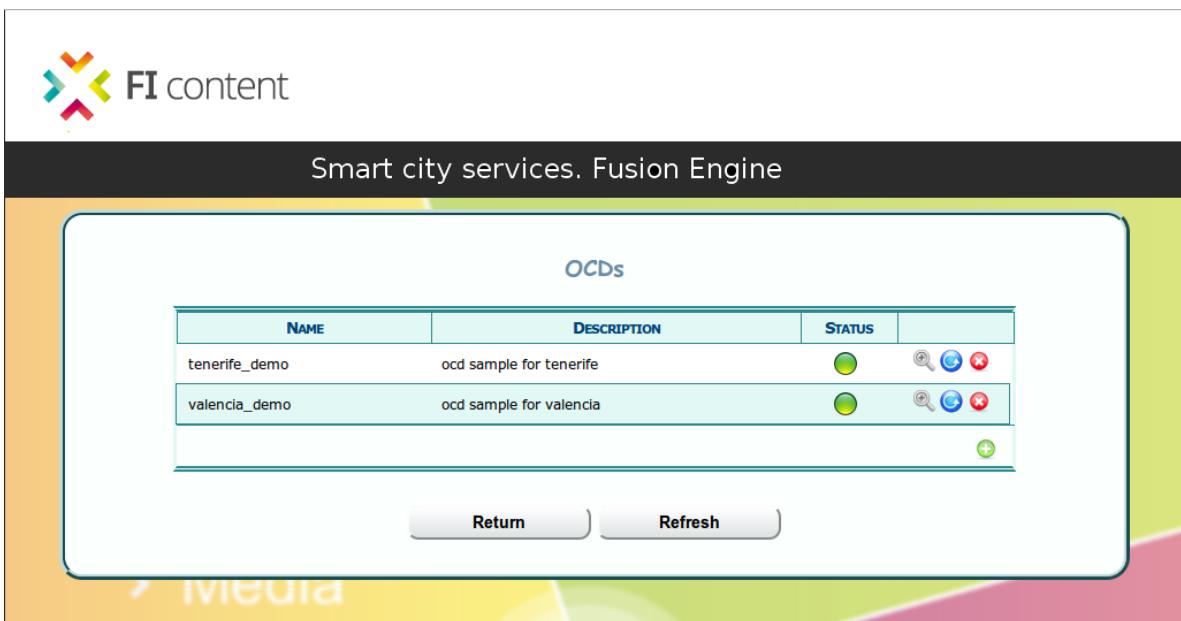
The main page of OCDs is a list of current available OCD in the database (OCD_BASE). After the installation, you should get two demo OCDs: tenerife_demo and valencia_demo.

The list is displayed in form of a table where you can see:

- The name of the OCD
- A description of the OCD
- The status: a green circle indicates that the fusion process terminated successfully and you can access its inquiry API (just go to the 'Demo' or 'Swagger' section in the FE main menu in Figure 3).

There are multiple status an OCD can have:

- *New* : the OCD has been created. Its values are stored in OCD_BASE
- *Initialized* : the configuration values of this OCD available in the OCD_BASE database are replicated in a new OCD_<ocdName> database .
- *Running* : the FE service is performing fusion on this OCD. It is adding new POIs in the OCD_<ocdName> database
- *Finished_ok* : the fusion process terminated successfully
- *Finished_error* : the fusion process terminated with errors
- A set of actions to be performed :
 - Edit/View the OCD : here you can see the settings of the OCD. You can only edit the values before initializing the OCD, otherwise you can only view the OCD configuration parameters.
 - Restart the OCD : the OCD will be restarted. The configuration parameters are preserved but the fusion process is relaunched. This might be helpful after correcting the errors when the fusion takes place.
 - Delete the OCD : the OCD will be deleted



The screenshot shows the 'Smart city services. Fusion Engine' interface. At the top, there's a header bar with the title 'Smart city services. Fusion Engine'. Below it is a main content area with a title 'OCDs'. A table lists two entries:

NAME	DESCRIPTION	STATUS
tenerife_demo	ocd sample for tenerife	
valencia_demo	ocd sample for valencia	

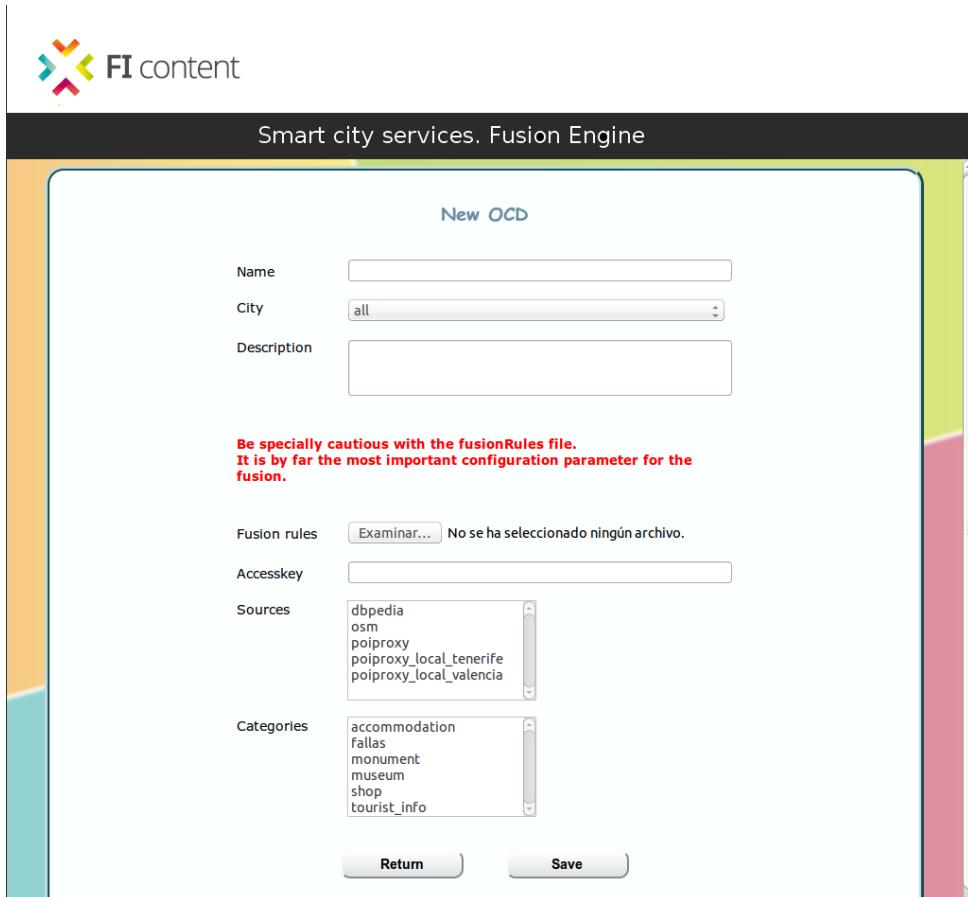
Below the table are two buttons: 'Return' and 'Refresh'. There's also a small green '+' button at the bottom right of the table area.

Figure 5. OCDs main menu

Additionally, you can create a new OCD by clicking on the green + button. For the OCD, you need to enter:

- Name : name of the OCD
- City: the city the OCD belongs to. The city is configured under the *Configuration* submenu (See chapter 3.1.1.2 -).
- Description: optional. Enter here a description of the OCD.
- FusionRules: Add here a XML with the FusionRules. See section 3.3 - for more information. Be cautious here as the FusionRules file is very important.
- Accesskey: this is just a key for accessing this particular OCD later. It is currently (you can enter whatever you want) not used but you might use it for later control access.

- Sources: Select all sources you want to use for your OCD. Sources are configured under the Configuration submenu (See chapter 3.1.1.2 -).
- Categories: Select all categories you want to use for your OCD. Categories are configured under the Configuration submenu (See chapter 3.1.1.2 -).



The screenshot shows the 'Smart city services. Fusion Engine' interface with a 'New OCD' configuration form. The form includes fields for Name, City (set to 'all'), and Description. A note in red text at the bottom of the form reads: 'Be specially cautious with the fusionRules file. It is by far the most important configuration parameter for the fusion.' Below the note, there are fields for Fusion rules (with a placeholder 'Examinar... No se ha seleccionado ningún archivo.') and Accesskey. Under 'Sources', a list box contains 'dbpedia', 'osm', 'poiproxy', 'poiproxy_local_tenerife', and 'poiproxy_local_valencia'. Under 'Categories', a list box contains 'accommodation', 'fallas', 'monument', 'museum', 'shop', and 'tourist_info'. At the bottom of the form are 'Return' and 'Save' buttons.

Figure 6. New OCD

After entering properly the data and clicking the 'Save' button, you will get a message whether the process went well (successful message) or wrong (error message). If everything went well, after pressing the 'Return' button, you will see the new OCD in the OCD's list. Note that the status should be 'new'. For a quick how-to on how to create an OCD please have a look at section 3.4 -.

3.1.1.2 - Configuration

In the Configuration menu you can set up various things OCDs relies on:

- Licenses : refers to license terms
- City : cities to be represented by OCDs
- API: refers to the APIs of data sources to catch data from. Currently only CitySDK is supported
- Persistence : allows configuring database parameters
- Data sources : represents the list of sources to apply fusion with
- Categories : represents the different categories available

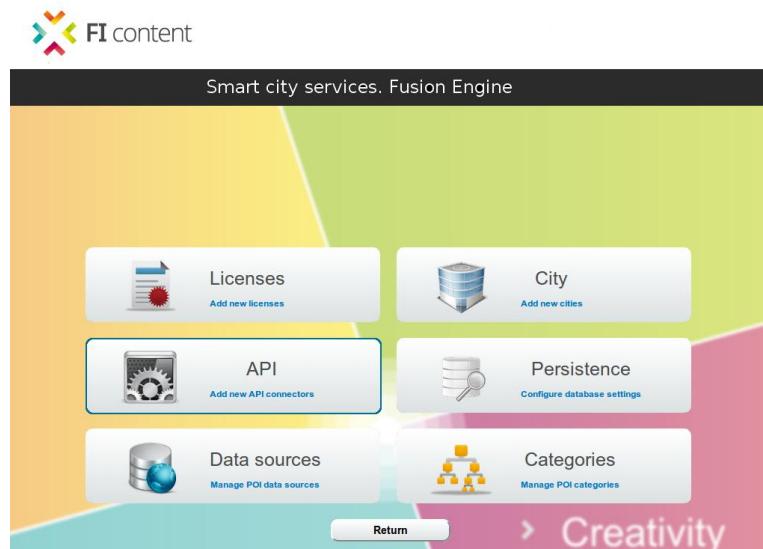


Figure 7. Configuration main menu

3.1.1.2.1 - Licenses

The Licenses section allows entering multiple Licenses. Note that if we are dealing with multiple data sources, each one has its own licenses and terms of use, and you should preserve it during and after the fusion. The Fusion database only keeps track of the different licenses. It is up to the administrator and his/her company to legally use the fused data considering all the involved licenses.

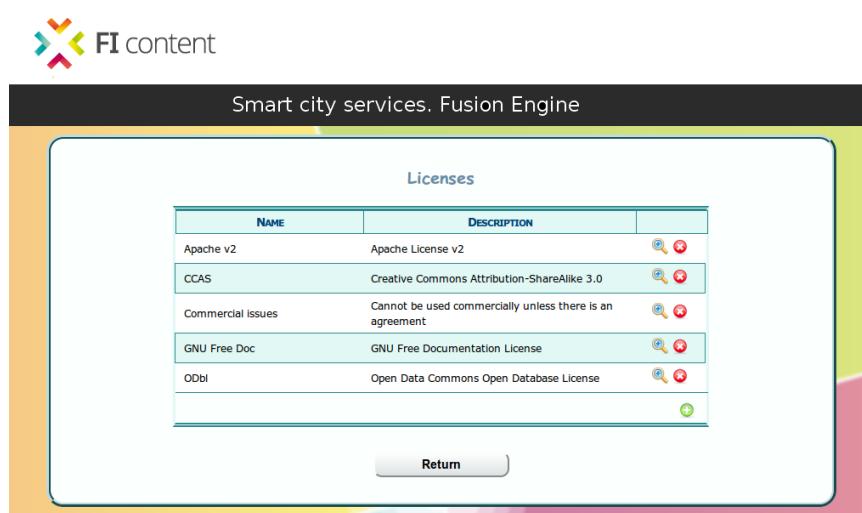


Figure 8. Licenses main menu

Currently there are several Licenses introduced, as by default the FE uses OSM, DBPedia and POIProxy. For simplicity, an aggregated 'Commercial issues' special license is already there for any license that involves some agreement between parties.

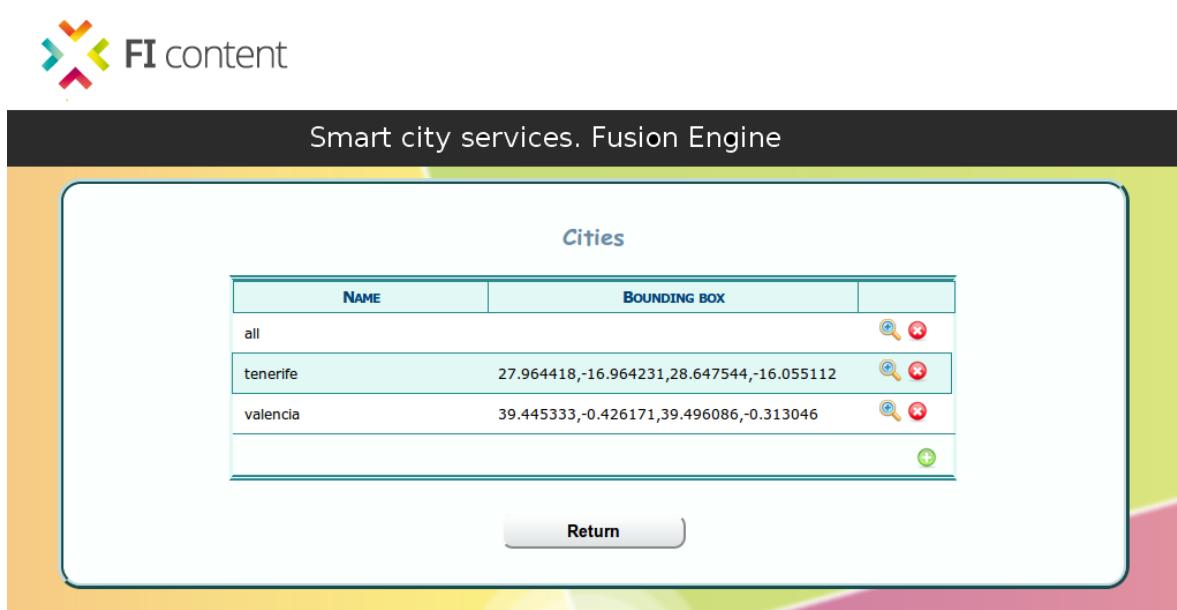
You can edit and delete any of the listed licenses. You can also create your own license by clicking on the green + button. You should only introduce a name, a description and (optionally) additional information.

3.1.1.2.2 - City

The *City* section allows describing cities, which are represented by a name and a bounding box. By default (after a clean installation) there are three cities:

- All: this represents the whole world. This is helpful for global data sources that can provide information from all over the world.
- Tenerife : represents the Tenerife island
- Valencia : represents the city of Valencia

Note that even if it is called a city, as long as its main property is a bounding box, you can even insert here villages, countries, etc.



The screenshot shows the 'Smart city services. Fusion Engine' interface. In the center, there is a table titled 'Cities' with three entries:

NAME	BOUNDING BOX	
all		
tenerife	27.964418,-16.964231,28.647544,-16.055112	
valencia	39.445333,-0.426171,39.496086,-0.313046	

At the bottom right of the table is a green '+' button. Below the table is a 'Return' button.

Figure 9. City main menu

You can edit and delete any of the listed cities. You can also create new cities by clicking on the green + button. You just need to provide a name and a valid bounding box. The bounding box is represented by [lat1,lon1,lat2,lon2], where

- Lat1 is the latitude of the bottom left corner in WGS84 system
- Lon1 is the longitude of the bottom left corner in WGS84 system
- Lat2 is the latitude of the top right corner in WGS84 system
- Lon2 is the longitude of the top right corner in WGS84 system

3.1.1.2.3 - API

API types refer to the different interfaces exposed by the data sources to be used by the FE. It is important to know how to access this data and the data format. Currently only CitySDK is supported for the fusion (though FIWARE API is also listed), but you can add new types and program your own connector. This issue will be commented in the *Developer's guide*.

You can edit and delete each of the listed API types. You can also create your own API type by clicking on the green + button. Requires parameters are:

- Name : name of the API type
- Description : some description
- API rules file (future use) : a file that better describes the API that can be interpreted by the FE.



The screenshot shows a web interface titled "Smart city services. Fusion Engine". Below it, a section titled "API Types" displays a table with two rows:

NAME	DESCRIPTION	
CitySDK	Following CitySDK project and OGC POI SWG	 
FI-PPP POI API	Following FI-PPP project	 
		

At the bottom of the table is a "Return" button.

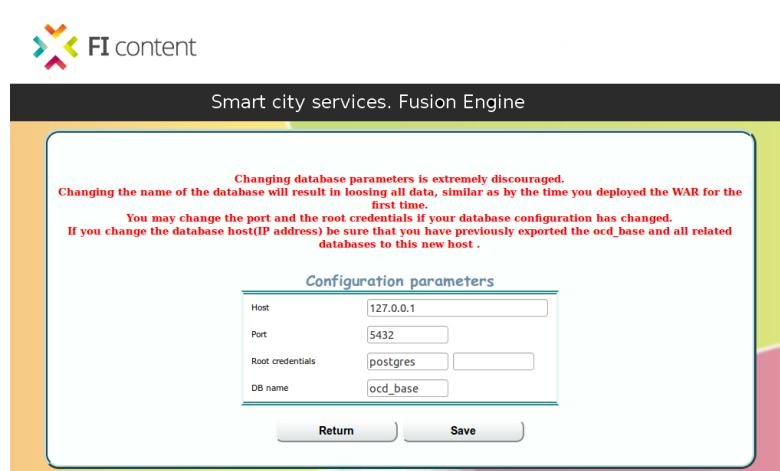
Figure 10. API types list

3.1.1.2.4 - Persistence

Here you can specify the connection parameters to the database. Note that changing database parameters is extremely discouraged after installation if you have already created OCDs. Changing the name of the database will result in losing all data, similar as by the time you deployed the WAR for the first time.

You may change the port and the root credentials if your database configuration has changed.

If you change the database host (IP address) be sure that you have previously exported the OCD_BASE and all related databases to this new host.



The screenshot shows a web interface titled "Smart city services. Fusion Engine". A message box contains the following text:

Changing database parameters is extremely discouraged.
 Changing the name of the database will result in losing all data, similar as by the time you deployed the WAR for the first time.
 You may change the port and the root credentials if your database configuration has changed.
 If you change the database host(IP address) be sure that you have previously exported the ocd_base and all related databases to this new host.

Below the message box is a "Configuration parameters" form with the following fields:

Host	127.0.0.1
Port	5432
Root credentials	postgres
DB name	ocd_base

At the bottom of the form are "Return" and "Save" buttons.

Figure 11. Persistence menu

3.1.1.2.5 - Data sources

Data sources refer to all sources we want to consider in the FE. It includes both global data un local data. After a clean installation, you can find the following sources:

- dBpedia : refers to dBpedia, the structured Wikipedia. Global data source.
- OSM : OpenStreetMaps. Global data source.
- Poiproxy : POIProxy is another enabler developed in FiContent2 thta can provide information from multiple social networks. Thus it is considered as global data source
- Poiproxy_local_tenerife : POIProxy also allows to retrieve information from local data sources. In this case, we consider it an independent local data source for the city of Tenerife
- Poiproxy_local_valencia : similar to the previous one but for the city of Valencia

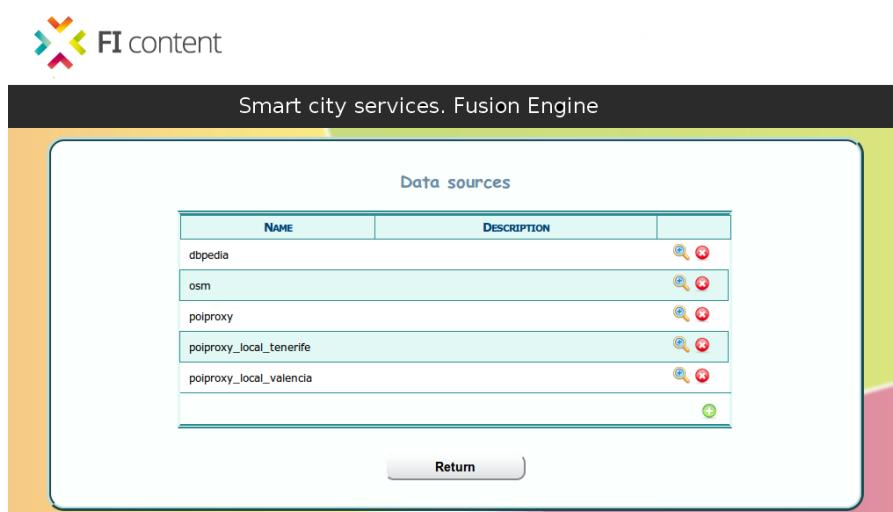
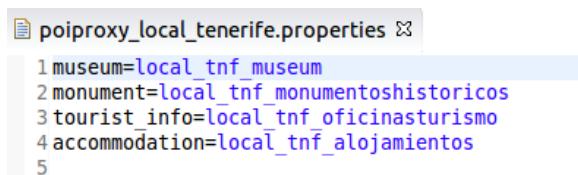


Figure 12. Data sources menu

You can edit and delete any of the listed data sources, though typically you may want to have here as many sources as possible to enrich the result of the fusion. If you click on the 'Edit' icon, you can see the required parameters for each data source (see Figure 14):

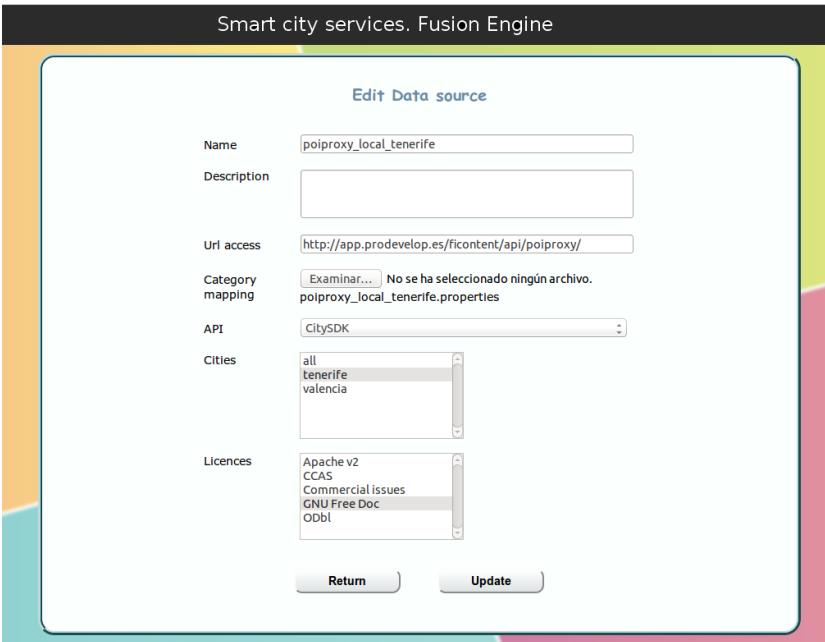
- Name : name of the data source
- Description : some description about it (optional)
- URL access : URL endpoint to query for data
- Category mapping : this is a properties file that maps the global categories of the FE with the particular categories of a data source. For example, in Figure 13, 'museum' refers to a category in the FE domain whereas 'local_tnf_museum' refers to a category in the Poiproxy_local_tenerife data source. Note that the mapping can be 1:N, not necessarily 1:1. The category mapping files are located in the config/categoryMapping directory where the WAR has been deployed.



```
poiproxy_local_tenerife.properties
1 museum=local_tnf_museum
2 monument=local_tnf_monumentoshistoricos
3 tourist_info=local_tnf_oficinasturismo
4 accommodation=local_tnf_alojamientos
5
```

Figure 13. Category mapping example

- API : API type. Currently only CitySDK supported
- Cities : supported cities
- Licenses : licenses assigned to this data sources



The screenshot shows the 'Edit Data source' page of the Smart city services. Fusion Engine. The form includes fields for Name (poiproxy_local_tenerife), Description, Url access (http://app.prodevelop.es/ficontent/api/poiproxy/), Category mapping (Examinar... No se ha seleccionado ningún archivo. poiproxy_local_tenerife.properties), API (CitySDK), Cities (dropdown menu with options: all, tenerife, valencia, where 'tenerife' is selected), and Licences (dropdown menu with options: Apache v2, CCAS, Commercial issues, GNU Free Doc, ODbL, where 'Commercial issues' is selected). At the bottom are 'Return' and 'Update' buttons.

Figure 14. Data sources (edit source)

3.1.1.2.6 - Categories

Categories are a way of dealing with and organizing POIs. The main problem is that each data source might have its own category catalogue, and we should deal with only one common catalogue for the FE. Thus we have listed some common categories after the installation: accommodation, fallas, monument, museum, shop, tourist_info. You should add as many categories as possible.



The screenshot shows the 'Categories' main menu of the Smart city services. Fusion Engine. It displays a table with columns 'NAME' and 'DESCRIPTION'. The categories listed are accommodation, fallas, monument, museum, shop, and tourist_info. Each row has a small edit icon (pencil) and a delete icon (red circle with a slash) to the right. A green plus sign icon is located at the bottom right of the table area. At the bottom is a 'Return' button.

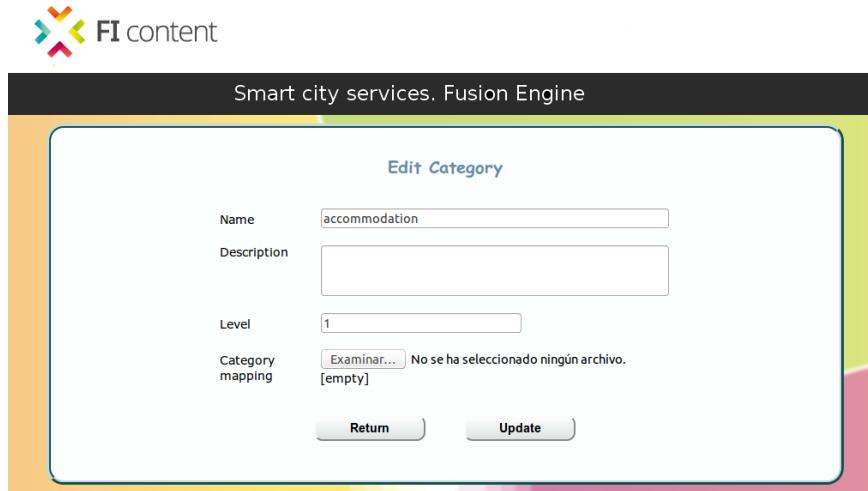
NAME	DESCRIPTION	
accommodation		
fallas		
monument		
museum		
shop		
tourist_info		

Figure 15. Categories main menu



You can edit and delete each of the listed categories, and create new ones. The required parameters are:

- *Name* : name of the category in the FE domain
- *Description* : some description (optional)
- *Level* : level of the category. This is just for future use and allows to build subcategories
- *Icon* : icon identifying the category (optional). This is just for future use and allows to easily identify categories by icons, when you work with multiple categories.



The screenshot shows a web-based application interface titled "Smart city services. Fusion Engine". At the top, there is a header bar with the "FI content" logo. Below the header, the main content area has a title "Edit Category". The form contains the following fields:

- Name:** accommodation
- Description:** (empty text area)
- Level:** 1
- Category mapping:** A file selection input field showing "Examinar..." and the message "No se ha seleccionado ningún archivo. [empty]".

At the bottom of the form are two buttons: "Return" and "Update".

Figure 16. Edit category menu



3.1.2 - Docs

The *Docs* menu is just a menu that allows you to see online or download all the documentation regarding the FE. You can find here a link to three different files:

- *Installation guide*: describes how to install the FE enabler (FE frontend and FE service)
- *User & Admin guide*: describes how to use the FE
- *Developer guide*: describes more details for developers if they want to change or adapt the code for their needs.



Figure 17. FE documentation menu



3.1.3 - Demo

This *Demo* menu provides an easy and fast way to show POIs from a particular built OCD. In the main page you have to select the OCD you want to show. In this select you will find all OCDs that have been created previously and are available in the OCD_BASE database. Note that you will only see those OCDs whose fusion process has already been finished. Running OCDs will not be shown.

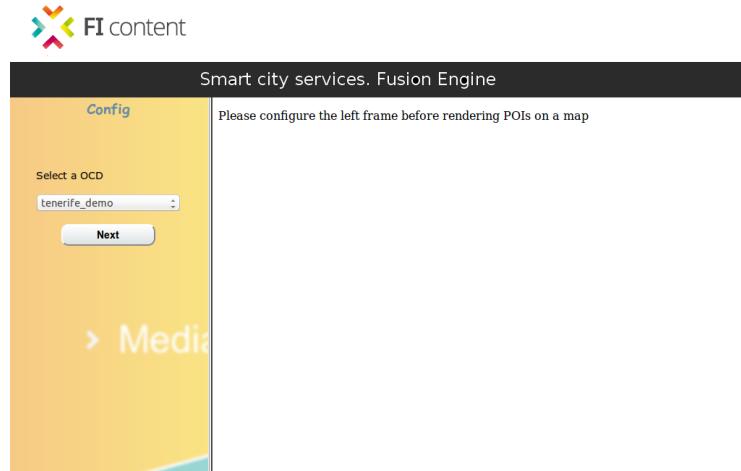


Figure 18. Demo initial page

Just select one of the OCDs and Click on the 'Next' button. You will see a list of defined categories for this OCD (see Figure 19) . This list has been defined by the time the OCD was built.



Figure 19. Demo (selecting category)

Just click on one or more categories and press the 'Show' button. You will see on the right frame all POIs found (see Figure 20)

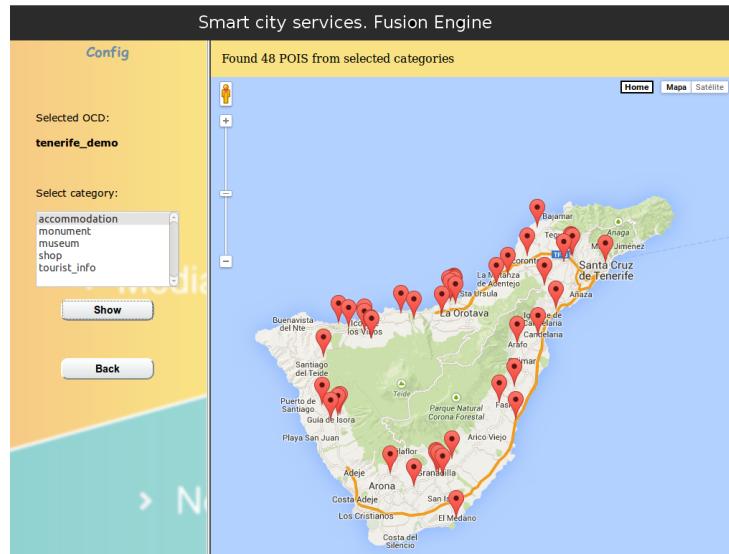


Figure 20. Demo (showing POIs)

On the upper side you can find a text showing the number of found POIs. You can get more information on a particular POI by clicking on it. A window will be open with more detailed information. This is a part of the information obtained by the fusion engine. For your custom application, you can select which items to show. The whole data items for each POI are available through the REST API (see the Swagger menu for more information). Or you can also access the source code and extract it from the database. For this, you will probably need to have a look at the *Developer's guide*.

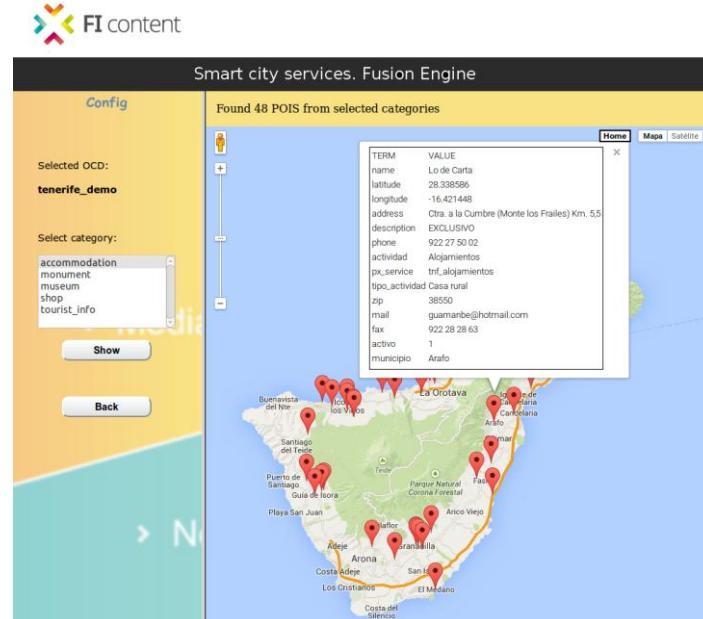


Figure 21. Demo (showing POI details)



3.1.4 - Swagger

Swagger is a simple yet powerful representation of your RESTful API. With the largest ecosystem of API tooling on the planet, thousands of developers are supporting Swagger in almost every modern programming language and deployment environment. With a Swagger-enabled API, you get interactive documentation, client SDK generation and discoverability (<http://swagger.io/>).

This menu in the FE frontend is a way of describing the access API once a OCD has been created and the fusion process has finished. This is important to note: the API is for accessing the OCD after the fusion, not for the fusion itself.

The main menu displays the API test page. Once an OCD is available (fusion process has finished), you can request basically two things, according to the CitySDK interface:

- Categories : show all categories for an existing OCD
- POIs : show all POIs from an existing OCD (with certain features)

The screenshot shows the Swagger UI for the Fusion Engine API Test. At the top, there's a header with the FI content logo and the text "Smart city services. Fusion Engine". Below the header, the title "Fusion Engine API Test" is displayed, followed by a subtitle "Basic API to access to POI data generated after the fusion". There are two main sections: "categories" and "pois". Each section has a "Show/Hide" button, a "List Operations" button, and an "Expand Operations" button. At the bottom left, it says "[BASE URL: /fic2_fe_v3_frontend , API VERSION: 0.2.0]". On the right, there's a green "VALID" button and a copy icon. A status bar at the bottom right indicates "VALID" and shows a copy icon.

Figure 22. Swagger main menu

3.1.4.1 - Categories

If you click on 'categories', you will get the basic REST operation (GET) in a blue line to obtain the categories

The screenshot shows the "categories" section of the Swagger menu. It includes the FI content logo, the "Smart city services. Fusion Engine" header, and the "Fusion Engine API Test" title. Below the title is the subtitle "Basic API to access to POI data generated after the fusion". The "categories" section is currently selected and highlighted in blue. It shows a GET operation with the endpoint "/citysdk/categories/search". There are "Show/Hide", "List Operations", and "Expand Operations" buttons for this section. Below the "categories" section, there's another section for "pois" with similar controls. At the bottom left, it says "[BASE URL: /fic2_fe_v3_frontend , API VERSION: 0.2.0]". On the right, there's a green "VALID" button and a copy icon. A status bar at the bottom right indicates "VALID" and shows a copy icon.

Figure 23. Swagger menu. Categories



If you further click on the blue line, you can see the model schema in JSON format. If you want to test it, you only have to provide two input parameters:

- ocdName : name of a current OCD (e.g. valencia_demo)
- list : poi

The screenshot shows the Swagger UI interface for the Fusion Engine API Test. The title bar says "Smart city services. Fusion Engine". The main section is titled "Fusion Engine API Test" with the subtitle "Basic API to access to POI data generated after the fusion".

The "categories" section is selected. At the top, there is a "GET /citysdk/categories/search" button, a "Response Class (Status 200)" link, and "Model Schema" tabs. The "Model Schema" tab is active, displaying the following JSON schema:

```
{  
  "categories": [  
    {  
      "label": [  
        {  
          "term": "string",  
          "value": "string",  
          "source": "string"  
        }  
      ]  
    }  
  ]  
}
```

Below the schema, the "Response Content Type" is set to "application/json". The "Parameters" table lists two parameters:

Parameter	Value	Description	Parameter Type	Data Type
ocdName	valencia_demo	Name of the ocd to look for	query	string
list	poi	Currently only "poi" supported	query	string

The "Response Messages" section shows a single entry:

HTTP Status Code	Reason	Response Model
default	Wrong syntax	

A "Try it out!" button is located at the bottom left of the interface.

Figure 24. Swagger. Categories request

You can click on the '*Try it out!*' button and get the response (see Figure 25). You can see here all available categories for the OCD *valencia_demo*.



Smart city services. Fusion Engine

Request URL

```
http://localhost:8080/fic2_fe_v3_frontend/citysdk/categories/search?ocdName=valencia_demo&list=poi
```

Response Body

```
{
  "categories": [
    {
      "label": [
        {
          "term": "primary",
          "value": "accommodation"
        }
      ]
    },
    {
      "label": [
        {
          "term": "primary",
          "value": "fallas"
        }
      ]
    },
    {
      "label": [
        ...
      ]
    }
  ]
}
```

Response Code

```
200
```

Response Headers

```
{
  "server": "Apache-Coyote/1.1",
  "content-type": "application/json;charset=UTF-8",
  "transfer-encoding": "chunked",
  "date": "Wed, 15 Apr 2015 10:04:58 GMT"
}
```

Figure 25. Swagger. Categories response

3.1.4.2 - POIS

Starting from the Swagger main menu, if you click on pois, you get the basic REST operation (GET).



Smart city services. Fusion Engine

Fusion Engine API Test

Basic API to access to POI data generated after the fusion

categories	Show/Hide List Operations Expand Operations
pois	Show/Hide List Operations Expand Operations
GET /citysdk/pois/search	Search pois by category and (optionally) including coordinates

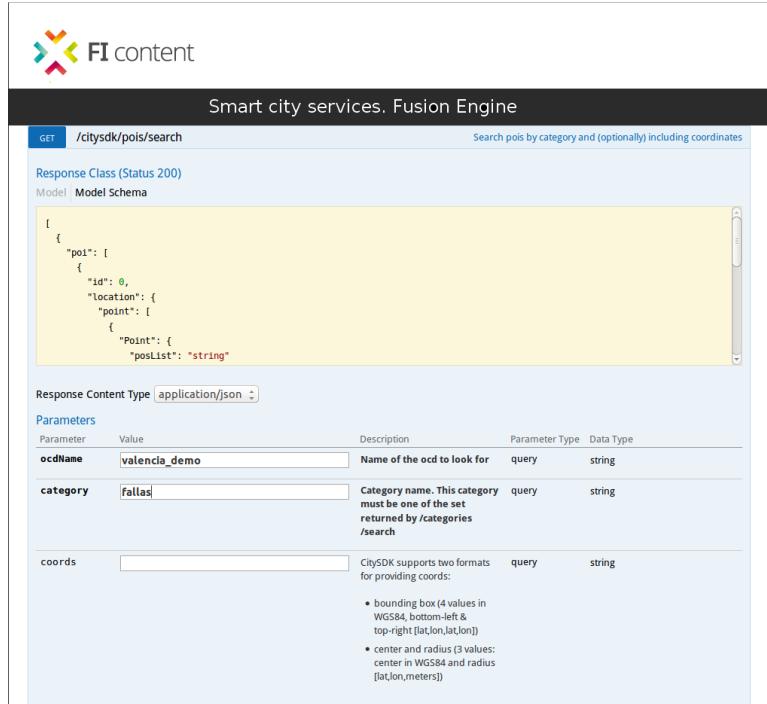
[BASE URL: /fic2_fe_v3_frontend , API VERSION: 0.2.0]

Figure 26. Swagger pois

If you click again on the blue line (you can also click on ‘Expand Operations’ on the right side), you can see the different parameters required:

- **ocdName (mandatory)** : OCD to query (e.g. valencia_demo)
- **category (mandatory)** : one of the categories available in this OCD. This corresponds to one of the categories that you have obtained in the previous chapter.

- *coords (optional)* : this allows you to introduce either a bounding box or a circle within the OCD in order to restrict the search to a particular area.



The screenshot shows the Swagger UI interface for a 'Smart city services. Fusion Engine'. A GET request is defined for the endpoint '/citysdk/pois/search'. The 'Response Class (Status 200)' section displays a JSON schema example:

```
{
  "poi": [
    {
      "id": 0,
      "location": {
        "point": [
          {
            "Point": {
              "posList": "string"
            }
          }
        ]
      },
      "label": [
        {
          "term": "name",
          "source": "poiproxy_local_valencia",
          "value": "poiproxy_local_valencia"
        }
      ]
    }
  ]
}
```

The 'Parameters' section lists the required query parameters:

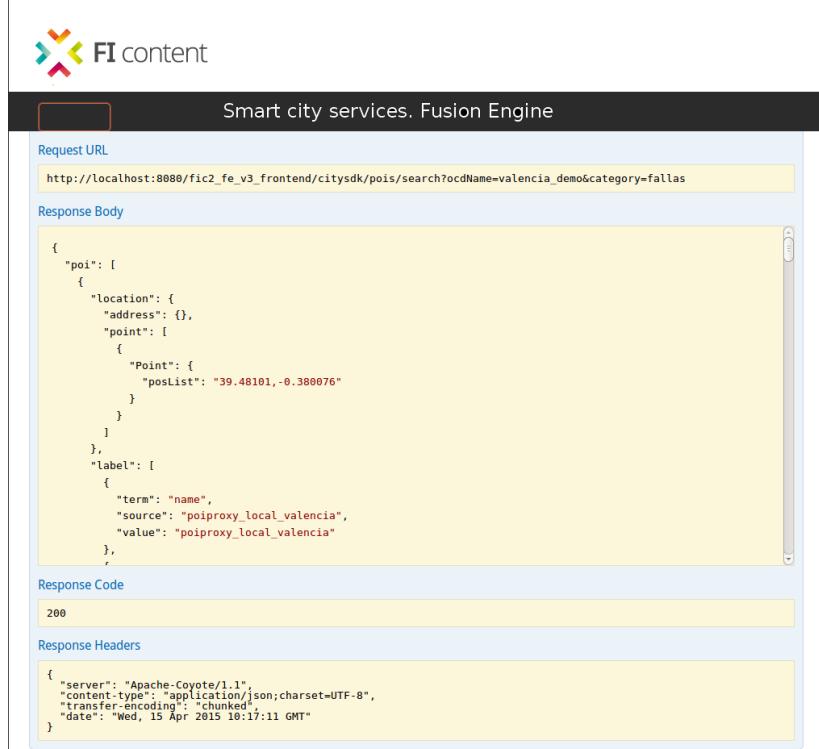
Parameter	Value	Description	Parameter Type	Data Type
ocdName	valencia_demo	Name of the ocd to look for	query	string
category	fallas	Category name. This category must be one of the set returned by /categories /search	query	string
coords		CitySDK supports two formats for providing coords:	query	string

Below the parameters, there is a note about 'coords' supported formats:

- bounding box (4 values in WGS84, bottom-left & top-right [lat,lon,lat,lon])
- center and radius (3 values: center in WGS84 and radius [lat,lon,meters])

Figure 27. Swagger pois request

If you click on the 'Try it out' button, you will get the response. Depending on the request the number of items in the JSON response can be quite large (e.g. for the fallas category you should get around 350 items)



The screenshot shows the Swagger UI interface for a 'Smart city services. Fusion Engine'. A GET request is defined for the endpoint '/citysdk/pois/search' with the query parameters 'ocdName' (valencia_demo) and 'category' (fallas). The 'Response Body' section displays the JSON response:

```
{
  "poi": [
    {
      "location": {
        "address": {},
        "point": [
          {
            "Point": {
              "posList": "39.48101,-0.388076"
            }
          }
        ]
      },
      "label": [
        {
          "term": "name",
          "source": "poiproxy_local_valencia",
          "value": "poiproxy_local_valencia"
        }
      ]
    }
  ]
}
```

The 'Response Code' section shows the status code 200. The 'Response Headers' section shows the following header information:

```
{
  "server": "Apache-Coyote/1.1",
  "content-type": "application/json;charset=UTF-8",
  "transfer-encoding": "chunked",
  "date": "Wed, 15 Apr 2015 10:17:11 GMT"
}
```

Figure 28. Swagger pois response

3.2 - FE Service

The configuration has been already described in the installation guide and is performed via the *config.xml* file. You will have to edit the values manually. You only need to configure some values of this configuration file (see Figure 29) :

- *The connection to the OCD_BASE database (user, pwd, host, port and name)*. This should be the same as for the frontend service. You have to use here also the root/admin user, as each OCD has its own database and the service needs to create one per each OCD.
- *LogFile*. This is the directory where to log data. Be sure that you have write permissions on this directory. This directory is important because :
 - The main thread will log data in the file specified by the 'logFile' parameter
 - The other threads (one per OCD) will log data in the same directory with the name <id_ocd>.log

```

config.xml ✘
1  <?xml version="1.0"?>
2  <configuration>
3    <DB>
4
5      <platform>postgresql</platform>
6      <driverName>org.postgresql.Driver</driverName>
7      <user>postgres</user>
8      <pwd></pwd>
9
10     <host>127.0.0.1</host>
11     <port>5432</port>
12     <name>ocd_base</name>
13
14     <resetOCDPOIScript>fe_ocd_reset_poi_data_postgres.sql</resetOCDPOIScript>
15
16     <scriptSeparator>;</scriptSeparator>
17
18     <geometryFromLonLatSrid>ST_SetSRID(ST_MakePoint({0}, {1}), {2})</geometryFromLonLatSrid>
19     <lonFromPoint>ST_X({0})</lonFromPoint>
20     <latFromPoint>ST_Y({0})</latFromPoint>
21
22   </DB>
23
24   <logFile>/home/fusion/fic2_fe_v3_service.log</logFile>
25
26
27 </configuration>
28

```

Figure 29. Config.xml example

Check the config.xml before running the service. Once executed, the FE will look up in the OCD_BASE database for OCDs to be fused. As long as there is no new OCD, you may only see in the main log file (fic2_fe_v3_service.log) this loop.

The signalling for starting to fusion a new OCD is performed via the FrontEnd service, as described in the previous chapter. Please see previous chapter if you don't know how to create a new OCD and start the fusion process.



Figure 30. OCD menu

If you click on one of them, you should trigger a new Fusion Engine process, and a new log file should appear in the configured FE service log directory. Besides, the status of the clicked OCD in the Frontend Service turns to 'Running' until the Fusion process ends (this may take several minutes).

For more information about the FE service, you may also look the Frontend Service, as both are complimentary services. You may therefore have a look at the *User guide* or even the *Admin guide*. Both files are available in *Github* under the '*doc*' folder.

3.3 - Fusion Rules

Fusion Rules is a key aspect when building OCDs, as it tells the FE service how to perform the fusion. You can have a look at initial fusion rules available under the config/fusionRules folder where you deployed the FE frontend WAR.

You can also view one FusionRules XML file by editing one of the available demo OCDs (either tenerife_demo or valencia_demo), clicking on the 'View' button.

Let's see an example for the tenerife_demo and the 'museum' category

```
<?xml version="1.0"?>
<fusion_rules city="tenerife_demo" bbox="27.964418,-16.964231,28.647544,-16.055112" fe_version="1.0">
    <fusion_rule category="museum" limit="50">
        <general_settings>
            <location>
                <source>poiproxy_local_tenerife</source>
                <source>dbpedia</source>
            </location>
            <name>
                <source>poiproxy_local_tenerife</source>
                <source>dbpedia</source>
            </name>
            <description>
                <source>dbpedia</source>
                <source>poiproxy_local_tenerife</source>
            </description>
            <image>
                <active>true</active>
                <limit>10</limit>
                <source>flickr</source>
            </image>
        </general_settings>
        <specific_settings>
            <max_distance>50</max_distance>
            <similarity_percentage>70</similarity_percentage>
        </specific_settings>
    </fusion_rule>
</fusion_rules>
```

The main element is *fusion_rules* and includes de *city* and the *bounding box*. Note that you have already entered these parameters in the *Configuration* section when you declared the city.

Then each category is described in a *fusion_rule* element, with a *limit* parameter. Each *fusion_rule* element is decomposed by two parts:

- *General settings* : this part indicates the priorities of the sources for some POI parameters. Following the example, when we have two POIs from different data sources that refer to the same, the location (GPS coordinates) of the merged POI will take as location the one provided by poiproxy_local_tenerife. The same applies for the name. For the description, we prioritize first dBpedia, but in case the description is not available we can take it from poiproxy_local_tenerife
- *Specific settings* : this refers to the POI matching mechanism, which is performed considering two criteria:
 - *Max_distance*: POIs a certain distance away from a reference POIs are not considered to be potential matching POIS. Such distance is set up in the fusionRules file. In Figure 31, POIs 4 and 5 will not be considered for matching candidates.

- Similarity_percentage POIs with a really dissimilar name are supposed to be different POIs. Such similarity (in percentage) value is set up in the fusionrules file. In Figure 31, POI 2 will provide an exact naming match whereas POIs 1 and 3 are quite dissimilar. There are several algorithms for detecting and evaluating similarity between two names (strings). Currently FE is using Monge & Elkan. There is a Java implementation provided at <http://secondstring.sourceforge.net/>

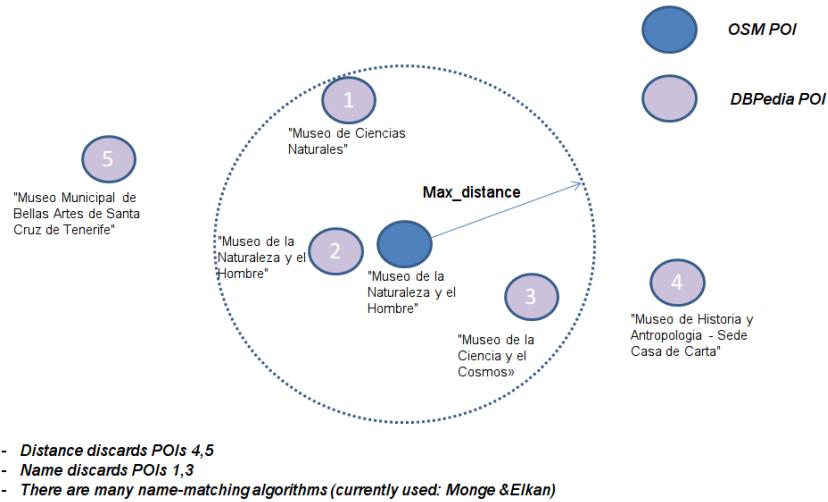


Figure 31. POI matching mechanism

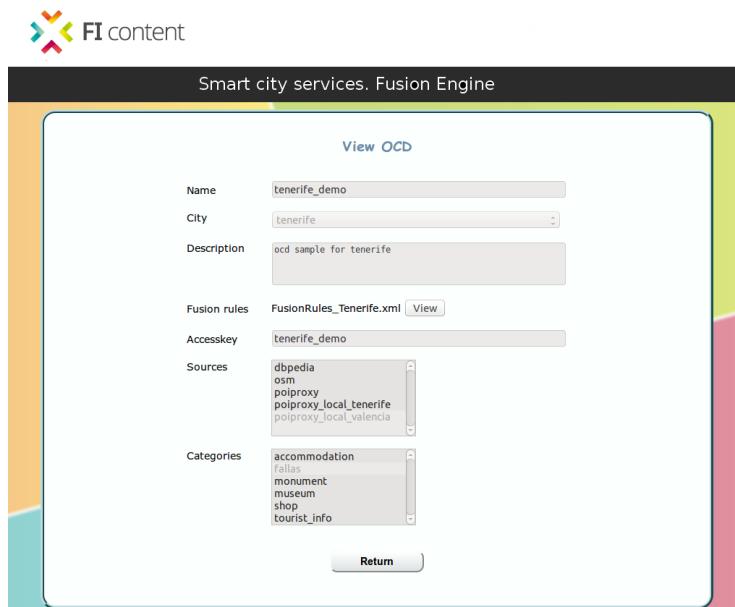
3.4 - How to quickly create an OCD

Follow these steps if you want to see in a fast view the process of building a new OCD. We will base on some work previously done for the demo cities (Tenerife and Valencia).

- 1) Go to the main Admin menu and press OCDs



- 2) Edit one OCD (e.g. Tenerife). Click on the 'View' button and Save the XML file as FusionRules_Tenerife2.xml into your computer.



- 3) Edit the file and leave just one category (e.g. museum). Change the name of the city to tenerife_demo2. You can also set the <active> element to false within the <image> element. You should have something like this :

```
<?xml version="1.0" encoding="UTF-8"?>
<fusion_rules city="tenerife_demo2" bbox="27.964418,-16.964231,28.647544,-16.055112"
fe_version="1.0">
    <fusion_rule category="museum" limit="50">
        <general_settings>
            <location>
                <source>poiproxy_local_tenerife</source>
                <source>dbpedia</source>
            </location>
        <name>
```

```

<source>poiproxy_local_tenerife</source>

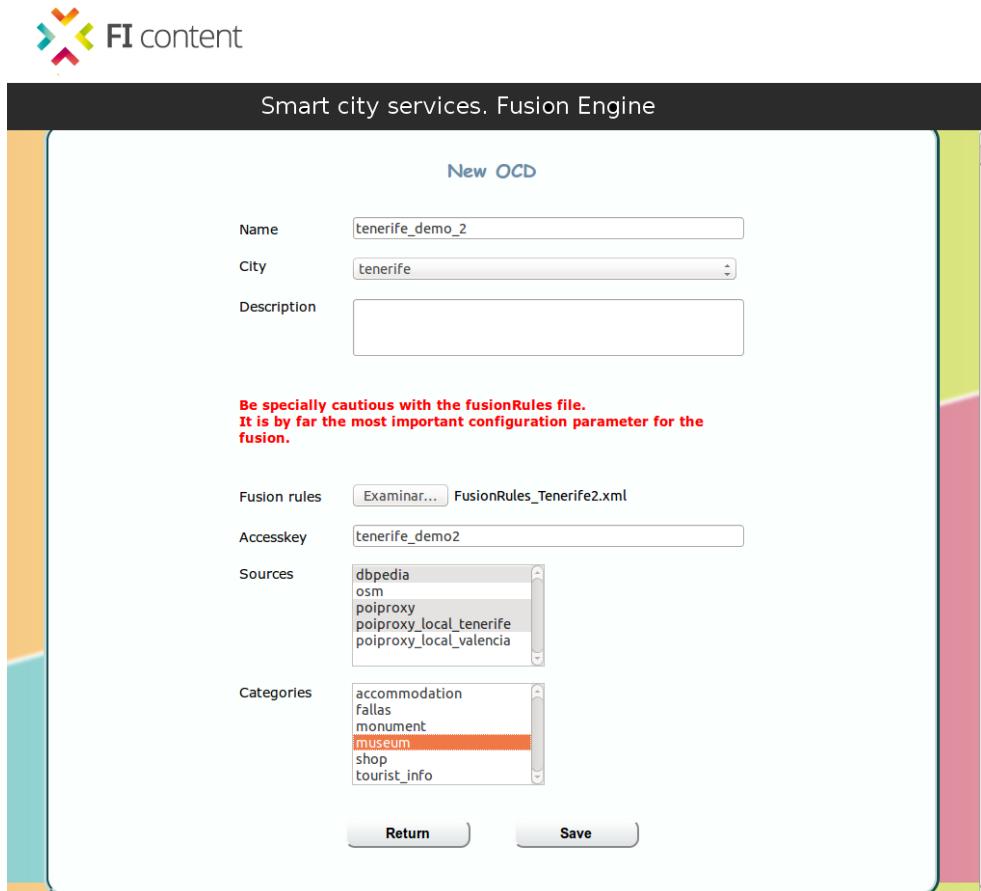
<source>dbpedia</source>
</name>
<description>
    <source>dbpedia</source>
    <source>poiproxy_local_tenerife</source>

</description>
<image>
    <active>false</active>
    <limit>10</limit>
    <source>flickr</source>

</image>
</general_settings>
<specific_settings>
    <max_distance>50</max_distance>
    <similarity_percentage>70</similarity_percentage>
</specific_settings>
</fusion_rule>
</fusion_rules>

```

- 4) Go to the OCD main menu and press the green + button. Enter the following parameters :



The screenshot shows the 'New OCD' configuration interface for the Fusion Engine. The form includes fields for Name (tenerife_demo_2), City (tenerife), Description (empty), and a note about the fusionRules file. It also includes fields for Fusion rules (FusionRules_Tenerife2.xml), Accesskey (tenerife_demo2), Sources (dbpedia, osm, poiproxy, poiproxy_local_tenerife, poiproxy_local_valencia), and Categories (accommodation, Fallas, monument, museum, shop, tourist_info). The 'museum' category is highlighted with a red background.

New OCD	
Name	tenerife_demo_2
City	tenerife
Description	
Be specially cautious with the fusionRules file. It is by far the most important configuration parameter for the fusion.	
Fusion rules	<input type="button" value="Examinar..."/> FusionRules_Tenerife2.xml
Accesskey	tenerife_demo2
Sources	<input type="checkbox"/> dbpedia <input type="checkbox"/> osm <input type="checkbox"/> poiproxy <input checked="" type="checkbox"/> poiproxy_local_tenerife <input type="checkbox"/> poiproxy_local_valencia
Categories	<input type="checkbox"/> accommodation <input type="checkbox"/> Fallas <input type="checkbox"/> monument <input checked="" type="checkbox"/> museum <input type="checkbox"/> shop <input type="checkbox"/> tourist_info
<input type="button" value="Return"/> <input type="button" value="Save"/>	



Click on the 'Save' button. The OCD should be created (successful message) and in the OCD list you should have the new OCD.

The screenshot shows the 'OCDs' section of the Smart city services. Fusion Engine interface. It displays a table with three rows of data:

NAME	DESCRIPTION	STATUS
tenerife_demo	ocd sample for tenerife	Green (OK)
tenerife_demo_2		Grey (New)
valencia_demo	ocd sample for valencia	Green (OK)

Below the table are two buttons: 'Return' and 'Refresh'.

- 5) Note the grey status icon (status is new). Click on it to initialize the OCD. Confirm the confirmation message and if everything goes well you should get a confirmation message

The screenshot shows a confirmation message: "OCD tenerife_demo_2 successfully initialized". Below the message is a single 'Return' button.

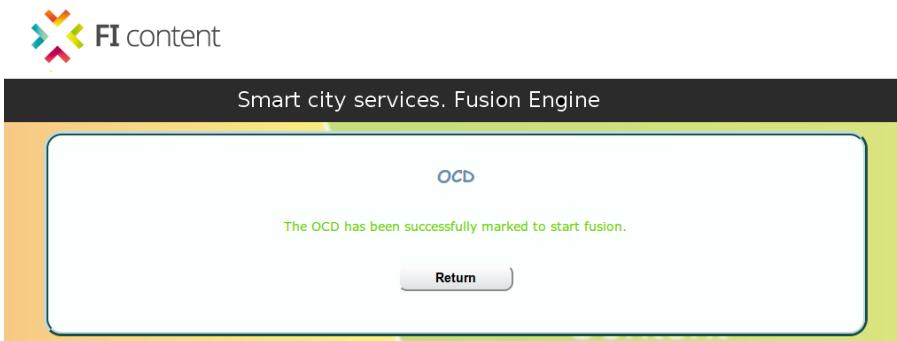
- 6) Return to the OCD main menu and you will see your OCD with a new status (initialized)

The screenshot shows the 'OCDs' section again. The 'tenerife_demo_2' entry now has a yellow status icon, indicating it has been initialized. The table data is as follows:

NAME	DESCRIPTION	STATUS
tenerife_demo	ocd sample for tenerife	Green (OK)
tenerife_demo_2		Yellow (Initialized)
valencia_demo	ocd sample for valencia	Green (OK)

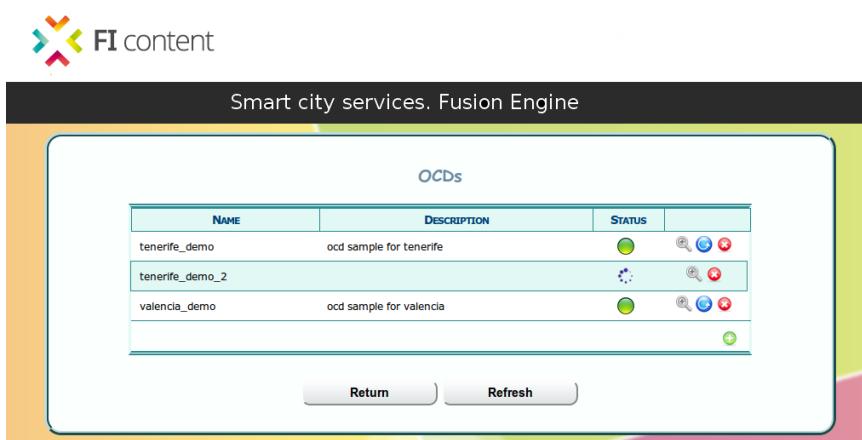
Below the table are 'Return' and 'Refresh' buttons.

- 7) Click on the yellow status icon to start the fusion. After accepting a confirmation message you will get the following message



The screenshot shows a web interface titled "Smart city services. Fusion Engine". A central box is labeled "OCD". Below it, a message says "The OCD has been successfully marked to start fusion." A "Return" button is at the bottom.

- 8) Return to the OCD main list. You will see that the status has changed. Please wait a little bit (for this small example it should be less than a minute)



The screenshot shows a table titled "OCDS" listing three entries:

NAME	DESCRIPTION	STATUS
tenerife_demo	ocd sample for tenerife	
tenerife_demo_2		
valencia_demo	ocd sample for valencia	

Buttons for "Return" and "Refresh" are at the bottom.

- 9) Click the 'Refresh' button to update the status. If the FE service went well, you should see a green status light for this OCD.



The screenshot shows the same table as before, but now all entries have a green status light:

NAME	DESCRIPTION	STATUS
tenerife_demo	ocd sample for tenerife	
tenerife_demo_2		
valencia_demo	ocd sample for valencia	

Buttons for "Return" and "Refresh" are at the bottom.

- 10) You can now go to the '*Demo*' section in the main menu and see the POIs. Obviously there is only one category to how POIs

Smart city services. Fusion Engine

Config

Selected OCD:
tenerife_demo_2

Select category:

museum

Show

Back

Found 11 POIs from selected categories

Home Mapa Satélite

