
Business API Ecosystem Documentation

Release latest

May 27, 2019

1	Index	3
1.1	Installation and Administration Guide	3
1.2	User Guide	10

This project is part of [FIWARE](#)

[CKAN](#) is an open data publication platform, widely extended, which enable the publication, search, discovery and consumption of open datasets.

In this context, the FIWARE CKAN extensions enhance the default CKAN features in order to integrate this tool within the FIWARE ecosystem, supporting the publication, management and rich visualization of right-time context data, while improving the access control and enabling data monetization.

In particular, the following extensions have been created:

- [OAuth2](#): This extension allows to use an external OAuth2 identity manager (Including but not limited to the FIWARE Idm) to authenticate CKAN users, including both, login in the web portal and accessing the APIs using OAuth2 access tokens.
- [Private Datasets](#): This extension extends the default dataset access permissions by including an access list where it is possible to specify what users can access to the data, even if they are not part of the owner organization
- [NGSI View](#): This extension enables the publication of right-time queries to the [Context Broker](#) as dataset resources.
- [BAE Publisher](#): This extension integrates CKAN with the FIWARE [Business API Ecosystem](#), enabling the automatic creation of products and offerings using the information of a private dataset.
- [WireCloud View](#): This extension allows the creation of rich visualizations for dataset resources by embedding [WireCloud](#) dashboards as resource views.
- [Data Requests](#): This extension enables users to ask for datasets which are not published in the portal, so it can include not only data offers but data demands.

The current documentation includes several guides for the usage, installation and administration of the different FIWARE CKAN extensions, for documentation of CKAN itself please refer to the [CKAN documentation](#).

Installation and Administration Guide The guide for maintainers that explains how to install the different extensions.

User Guide The guide for users and programmers that explains how to use the different extensions.

1.1 Installation and Administration Guide

1.1.1 Introduction

This installation and administration guide covers the FIWARE CKAN extensions for the FIWARE release 7. For details on the installation of CKAN itself please refer to the [CKAN Installation Guide](#)

Any feedback on this document is highly welcomed, including bugs, typos or things you think should be included but aren't. Please send them to the "Contact Person" email that appears in the [Catalogue page for this GEi](#). Or create an issue at [GitHub Issues](#)

The current version of the software has been tested in CKAN versions 2.7, and 2.8

1.1.2 OAuth2

This guide covers the installation of the OAuth2 CKAN extension v0.7.0, compatible with the software provided as part of the FIWARE release 7.

Installation

To install the extension in CKAN, it is required to activate the CKAN virtual environment as follows:

```
$ . /usr/lib/ckan/default/bin/activate
```

Note: The previous command is supposing that your CKAN virtualenv is in `/usr/lib/ckan/default`

The OAuth2 extension is available in pip and can be installed as follows:

```
$ pip install ckanext-oauth2==0.7.0
```

If you want to install the extension for development, you can download it with the following commands:

```
$ git clone https://github.com/conwetlab/ckanext-oauth2
$ cd ckanext-oauth2
$ git checkout v0.7.0
```

Once the extension has been downloaded and the virtualenv has been activated, the extension can be installed with the following command:

```
$ python setup.py develop
```

Configuration

To activate the OAuth2 extension in CKAN, it is required to include it in the `ckan.plugins` setting of the `.ini` file (generally `/etc/ckan/default/production.ini`).

```
ckan.plugins = oauth2 <other-plugins>
```

Additionally, this extension requires a set of settings to be provided in the `.ini` file in order to configure its behaviour and the integration with the selected IdM. In particular, The following is an example of configuration for using FIWARE Keyrock v7.6.0:

```
## OAuth2 configuration
ckan.oauth2.logout_url = /user/logged_out
ckan.oauth2.register_url = http://YOUR_OAUTH_SERVICE/sign_up
ckan.oauth2.reset_url = http://YOUR_OAUTH_SERVICE/users/password/new
ckan.oauth2.edit_url = http://YOUR_OAUTH_SERVICE/idm/settings
ckan.oauth2.authorization_endpoint = http://YOUR_OAUTH_SERVICE/oauth2/authorize
ckan.oauth2.token_endpoint = http://YOUR_OAUTH_SERVICE/oauth2/token
ckan.oauth2.profile_api_url = http://YOUR_OAUTH_SERVICE/user
ckan.oauth2.client_id = CLIENT_ID
ckan.oauth2.client_secret = YOUR_CLIENT_SECRET
ckan.oauth2.jwt.enable = false
ckan.oauth2.scope = all_info
ckan.oauth2.rememberer_name = auth_tkt
ckan.oauth2.profile_api_user_field = username
ckan.oauth2.profile_api_fullname_field = username
ckan.oauth2.profile_api_mail_field = email
ckan.oauth2.authorization_header = Bearer
```

With the following meaning:

- **logout_url:** URL used in CKAN for logging out the user
- **register_url:** URL in the selected IDM for registering users
- **reset_url:** URL in the selected IDM for resetting user password
- **edit_url:** URL in the selected IDM for updating user info
- **authorization_endpoint:** URL in the selected IDM for retrieving authorization code in the OAuth2 process.

- **token_endpoint**: URL in the selected IDM for retrieving the access token in the OAuth2 process
- **profile_api_url**: URL in the selected IDM for retrieving user info
- **client_id**: Client ID given by the IDM to the CKAN instance
- **jwt.enable**: Whether to use JSON Web Tokens to load user info
- **client_secret**: Client Secret given by the selected IDM for the CKAN instance
- **profile_api_user_field**: Name of the field which contains the user id within the user info object as provided by the IDM
- **profile_api_fullname_field**: Name of the field which contains the user display name within the user info object as provided by the IDM
- **profile_api_mail_field**: Name of the field which contains the user email within the user info object as provided by the IDM
- **authorization_header**: Type of header that will be used for accessing CKAN APIs using an access token for authentication

Version 0.7.0 of the extension supports the usage of JSON Web Tokens of both FIWARE Keyrock and JBoss Keycloak. To enable JWT using Keyrock the following changes in the configuration are required:

```
ckan.oauth2.jwt.enable = true
ckan.oauth2.scope = jwt
```

On the other hand, the following is a valid configuration for Keycloak IDM:

```
ckan.oauth2.logout_url = /user/logged_out
ckan.oauth2.edit_url = http://YOUR_OAUTH_SERVICE/auth/realms/REALM/account
ckan.oauth2.authorization_endpoint = http://YOUR_OAUTH_SERVICE/auth/realms/REALM/
↳protocol/openid-connect/auth
ckan.oauth2.token_endpoint = http://YOUR_OAUTH_SERVICE/auth/realms/REALM/protocol/
↳openid-connect/token
ckan.oauth2.profile_api_url = http://YOUR_OAUTH_SERVICE/auth/realms/REALM/protocol/
↳openid-connect/userinfo
ckan.oauth2.client_id = CLIENT_ID
ckan.oauth2.client_secret = YOUR_CLIENT_SECRET
ckan.oauth2.jwt.enable = true
ckan.oauth2.scope = profile email openid
ckan.oauth2.profile_api_user_field = user_name
ckan.oauth2.profile_api_mail_field = email
ckan.oauth2.authorization_header = Bearer
ckan.oauth2.profile_api_fullname_field = user_name
```

It is important to note that this extension has been designed by default to be used in a CKAN deployed in HTTPS, so trying to use it in an unsecured CKAN instance will result in an error. To use this extension in a not secured instance it is required to set the `OAUTHLIB_INSECURE_TRANSPORT` environment variable to true:

```
$ export OAUTHLIB_INSECURE_TRANSPORT=True
```

Or if CKAN is deployed in Apache, modifying `/etc/apache2/envvars`

Finally, the callback URL that must be used in the IDM for registering CKAN is:

```
https://YOUR_CKAN_INSTANCE/oauth2/callback
```

1.1.3 Private Datasets

This guide covers the installation of the Private Datasets CKAN extension v0.4, compatible with the software provided as part of the FIWARE release 7.

Installation

To install the extension in CKAN, it is required to activate the CKAN virtual environment as follows:

```
$ . /usr/lib/ckan/default/bin/activate
```

Note: The previous command is supposing that your CKAN virtualenv is in /usr/lib/ckan/default

The Private Datasets extension is available in pip and can be installed with the following command:

```
$ pip install ckanext-privatedatasets==0.4
```

If you want to install the extension for development, you can download it with the following commands:

```
$ git clone https://github.com/conwetlab/ckanext-privatedatasets
$ cd ckanext-privatedatasets
$ git checkout v0.4
```

Once the extension has been downloaded and the virtualenv has been activated, the extension can be installed with the following command:

```
$ python setup.py develop
```

Configuration

To activate the Private Datasets extension it is required to include *privatedatasets* in the *ckan.plugins* setting of your *.ini* file:

```
ckan.plugins = privatedatasets <other-plugins>
```

In addition, some settings can be configured in order to customize the extension behaviour. In particular:

```
ckan.privatedatasets.parser = ckanext.privatedatasets.parsers.
↳fiware:FiWareNotificationParser
ckan.privatedatasets.show_acquire_url_on_create = True
ckan.privatedatasets.show_acquire_url_on_edit = True
```

With the following meaning:

- **parser:** Class used to parse user access grants and revokes to datasets.
- **show_acquire_url_on_create:** Whether to show the acquire URL (URL where access grants can be obtained) on dataset creation form
- **show_acquire_url_on_edit:** Whether to show the acquire URL on dataset update form

1.1.4 Right Time Context

This guide covers the installation of the Right Time Context CKAN extension v0.9, compatible with the software provided as part of the FIWARE release 7.

Installation

To install the extension in CKAN, it is required to activate the CKAN virtual environment as follows:

```
$ . /usr/lib/ckan/default/bin/activate
```

Note: The previous command is supposing that your CKAN virtualenv is in `/usr/lib/ckan/default`

The Right Time Context extension is available in pip and can be installed with the following command:

```
$ pip install ckanext-right_time_context==0.9
```

If you want to install the extension for development, you can download it with the following commands:

```
$ git clone https://github.com/conwetlab/ckanext-right_time_context
$ cd ckanext-right_time_context
$ git checkout v0.9
```

Once the extension has been downloaded and the virtualenv has been activated, the plugin can be installed with the following command:

```
$ python setup.py develop
```

Configuration

To activate the Right Time Context extension it is required to include `right_time_context` in the `ckan.plugins` and `ckan.views.default_views` settings of your `.ini` file:

```
ckan.plugins = right_time_context <other-plugins>

ckan.views.default_views = right_time_context <other-views>
```

1.1.5 BAE Publisher

This guide covers the installation of the BAE Publisher CKAN extension v0.5, compatible with the software provided as part of the FIWARE release 7.

Requirements

- Business API Ecosystem v5.4.0 or higher
- OAuth2 CKAN extension installed
- Private Datasets extension installed

Installation

To install the extension in CKAN, it is required to activate the CKAN virtual environment as follows:

```
$ . /usr/lib/ckan/default/bin/activate
```

Note: The previous command is supposing that your CKAN virtualenv is in /usr/lib/ckan/default

The BAE Publisher extension is available in pip and can be installed with the following command:

```
$ pip install ckanext-baepublisher==0.5
```

If you want to install the extension for development, you can download it with the following commands:

```
$ git clone https://github.com/FIWARE-TMForum/ckanext-baepublisher
$ cd ckanext-baepublisher
$ git checkout v0.5
```

Once the extension has been downloaded and the virtualenv has been activated, the extension can be installed with the following command:

```
$ python setup.py develop
```

Configuration

To activate the BAE publisher extension it is required to include *baepublisher* in the *ckan.plugins* setting of your *.ini* file. In addition, it is necessary to configure the URL of the BAE instance to be used with *ckan.storepublisher.store_url*:

```
ckan.plugins = baepublisher <other-plugins>

ckan.baepublisher.store_url = https://YOUR_BAE_INSTANCE/
```

1.1.6 WireCloud View

This guide covers the installation of the WireCloud View CKAN extension v1.1.0, compatible with the software provided as part of the FIWARE release 7.

Requirements

- WireCloud v1.0 or higher
- OAuth2 CKAN extension installed

Installation

To install the extension in CKAN, it is required to activate the CKAN virtual environment as follows:

```
$ . /usr/lib/ckan/default/bin/activate
```

Note: The previous command is supposing that your CKAN virtualenv is in `/usr/lib/ckan/default`

The WireCloud View extension is available in pip and can be installed with the following command:

```
$ pip install ckanext-wirecloud_view==1.1.0
```

If you want to install the extension for development, you can download it with the following commands:

```
$ git clone https://github.com/conwetlab/ckanext-wirecloud_view
$ cd ckanext-wirecloud_view
$ git checkout 1.1.0
```

Once the extension has been downloaded and the virtualenv has been activated, the extension can be installed with the following command:

```
$ python setup.py develop
```

Configuration

To activate the WireCloud View extension it is required to include `wirecloud_view` in the `ckan.plugins` setting of your `.ini`:

```
ckan.plugins = wirecloud_view <other-plugins>
```

In addition, it is necessary to configure some settings related with the WireCloud instance used. In particular:

```
ckan.wirecloud_view.url = https://YOUR_WIRECLOUD_INSTANCE
ckan.wirecloud_view.editor_dashboard = wirecloud/ckan-editor
```

With the following meaning:

- **url:** URL of the WireCloud server
- **editor_dashboard:** Dashboard in the WireCloud instance used as wizard for basic dashboard creation

1.1.7 Data Requests

This guide covers the installation of the Data Requests CKAN extension v1.1.0, compatible with the software provided as part of the FIWARE release 7.

Installation

To install the extension in CKAN, it is required to activate the CKAN virtual environment as follows:

```
$ . /usr/lib/ckan/default/bin/activate
```

Note: The previous command is supposing that your CKAN virtualenv is in `/usr/lib/ckan/default`

The Data Requests extension is available in pip and can be installed with the following command:

```
$ pip install ckanext-datarequests==1.1.0
```

If you want to install the extension for development, you can download it with the following commands:

```
$ git clone https://github.com/conwetlab/ckanext-datarequests
$ cd ckanext-datarequests
$ git checkout v1.1.0
```

Once the extension has been downloaded and the virtualenv has been activated, the extension can be installed with the following command:

```
$ python setup.py develop
```

Configuration

To activate the Data Requests extension it is required to include *datarequests* in the *ckan.plugins* setting of your *.ini*:

```
ckan.plugins = datarequests <other-plugins>
```

Additionally, it is possible to customize the behaviour of the extension by including some settings. In particular:

```
ckan.datarequests.comments = true
ckan.datarequests.show_datarequests_badge = true
```

With the following meaning:

- **comments**: Whether the comments system is enabled or not
- **show_datarequests_badge**: Whether to show the number of data requests in a badge in the menu

1.2 User Guide

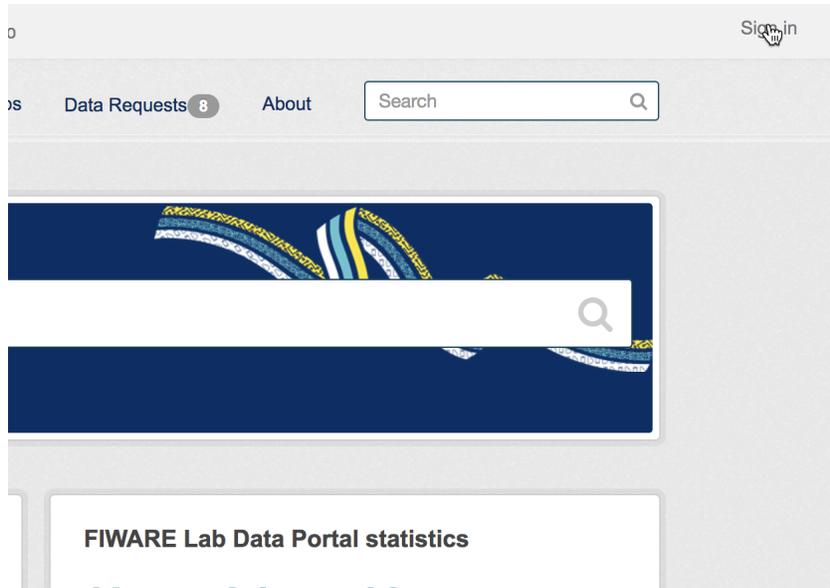
1.2.1 Introduction

This user guide covers the FIWARE CKAN extensions for the FIWARE release 7. For details on the usage of CKAN itself please refer to the [CKAN User Guide](#)

Any feedback on this document is highly welcomed, including bugs, typos or things you think should be included but aren't. Please send them to the "Contact Person" email that appears in the [Catalogue page for this GEi](#). Or create an issue at [GitHub Issues](#)

1.2.2 OAuth2

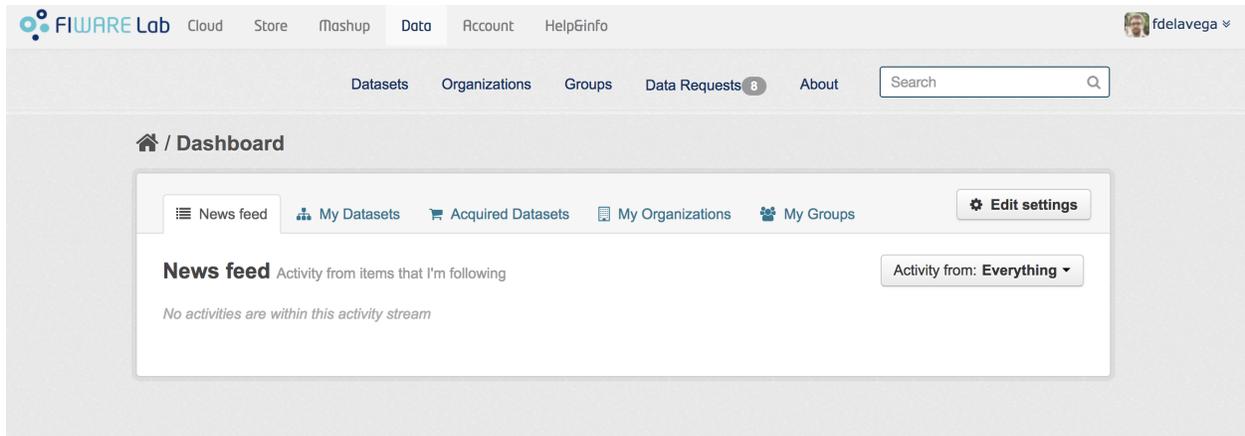
When the OAuth2 extension is installed and active in a CKAN instance, all user and sign in features are delegated to the external IDM. In this regard, the sign in process starts normally by clicking on *sign in*



Then, the OAuth2 plugin redirects CKAN to the configured OAuth2 IDM in order to provide user credentials

A screenshot of a 'Log In' form. The title 'Log In' is at the top left. Below it are two input fields: 'Email' containing 'f.delavega.garcia@gmail.com' and 'Password' with a masked password '••••••' and an eye icon. There is a 'remember me' checkbox and a 'Sign In' button. At the bottom, there are links for 'Sign up', 'Forgot password', and 'Didn't receive confirmation instructions?'.

Once done, the browser is redirected back to CKAN with the user signed in

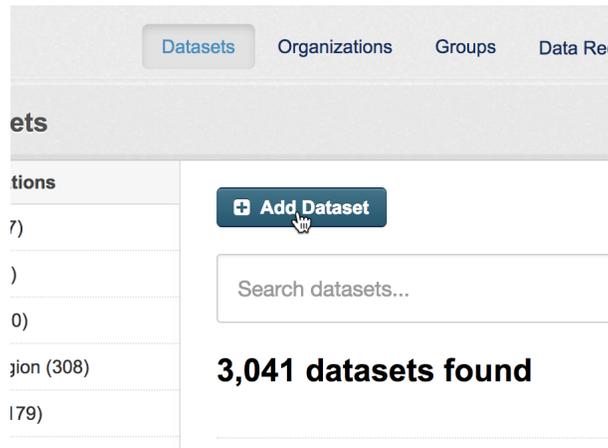


1.2.3 Private Datasets

The Private Datasets extension enhance the default dataset access control in CKAN by allowing dataset owners to specify what users of the platform can access to the data, even if they are not part of the same organization.

Additionally, it establishes the basic mechanism for acquiring access to the data by allowing to discover private datasets and configuring the site where this access can be acquired.

All these options can be configured in the dataset creation form, so the first step is clicking on *Add Dataset*



In this form, the extension integrates a set of fields to be filled by the dataset owner

Visibility: Private datasets can only be accessed by certain users, while public datasets can be accessed by anyone.

Searchable: Searchable datasets can be searched by anyone, while not-searchable datasets can only be accessed by entering directly its URL.

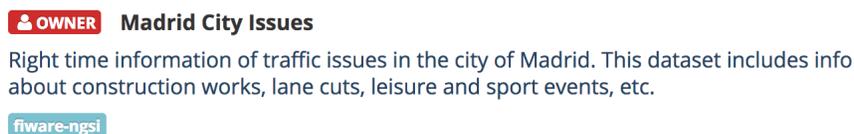
Allowed Users:

Acquire URL:

In particular:

- **Visibility:** Public or Private
- **Searchable:** True or false, whether a dataset set as private can be discovered by users, that is, it appears in searches
- **Allowed Users:** List of platform users authorized to access to the data when the dataset has been set as private
- **Acquire URL:** URL where it is possible to acquire access grants to the data

Depending of the chosen configuration and access rights of the users, datasets will appear in a different way. A private dataset created by the user logged in will appear as:



While a private dataset set as searchable will be seen as follows by an user without permissions.

PRIVATE Madrid City Issues

Right time information of traffic issues in the city of Madrid. This dataset includes info about construction works, lane cuts, leisure and sport events, etc.

If the dataset also has an *Acquire URL*, a button to access to the site will appear

PRIVATE Madrid City Issues [Acquire](#)

Right time information of traffic issues in the city of Madrid. This dataset includes info about construction works, lane cuts, leisure and sport events, etc.

Finally, private datasets will be seen as follows for those users who has access permissions

ACQUIRED Madrid City Issues

Right time information of traffic issues in the city of Madrid. This dataset includes info about construction works, lane cuts, leisure and sport events, etc.

fiware-ngsi

These *acquired* datasets are also located in the user dashboard under the *Acquired Datasets* tab

News feed My Datasets **Acquired Datasets** My Organizations My Groups Edit settings

Acquire Dataset

ACQUIRED Santander Parking
Real time NGSi data with the number of available parking spots in the centre of Santander
fiware-ngsi

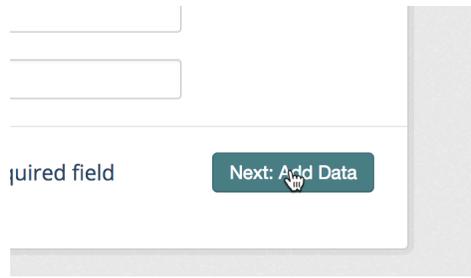
ACQUIRED Premium Issues
some premium data
fiware-ngsi

ACQUIRED Madrid City Issues
Right time information of traffic issues in the city of Madrid. This dataset includes info about construction works, lane cuts, leisure and sport events, etc.
fiware-ngsi

1.2.4 Right Time Context

The Right Time Context extension enables the publication of right-time context information as dataset resources by allowing to configure a NGSIV2 query.

These queries are provided as dataset resources, so the extension modifies the dataset resource form



Once creating the dataset resource, it is needed to use *fiware-ngsi* as data type in order to publish the query

A screenshot of a web interface for adding data to a dataset. At the top, a green progress bar shows two steps: '1 Create dataset' and '2 Add data', with '2 Add data' being the active step. Below the progress bar, there are two buttons: 'Upload' and 'Link'. The 'Name' field contains the text 'eg. January 2011 Gold Prices'. The 'Description' field contains the text 'Some useful notes about the data' and has a note at the bottom: 'You can use Markdown formatting here'. The 'Format' dropdown menu is open, showing 'eg. CSV, XML or JSON' at the top, a search input with 'fiware-ngsi' and a magnifying glass icon, and a blue highlighted option 'fiware-ngsi'. At the bottom right, there are three buttons: 'Previous', 'Save & add another', and 'Finish'.

It can be seen that once the *fiware-ngsi* data type is selected, the extension updates the form limiting the data providing mechanism to *URL* and injects a set of Context Broker specific fields

URL

Name

Description
You can use [Markdown](#) formatting here

Format This will be guessed automatically. Leave blank if you wish

Tenant

Service Path

OAuth-Token

These fields are:

- **URL:** Must include the query URL using NGSIv2
- **Tenant:** Corresponds to the *Fiware-Service* header used by the Context Broker
- **Service Path:** Corresponds to the *Fiware-ServicePath* header used by the Context Broker
- **Auth Required:** Whether to inject the OAuth2 access token of the logged user in the request, that is, the Context Broker is secured

Once the resource has been configured, the resource page includes the executed query and the result in JSON

NGSI query

🔧 Manage
🔗 Go to resource

URL: <https://context.opplafy.eu/v2/entities?type=CityIssue>

Tenant: madrid

Service Path: /transport/issues

From the dataset abstract

Right time information of traffic issues in the city of Madrid. This dataset includes info about construction works, lane cuts, leisure and sport events, etc.

Source: Madrid City Issues

📄 NGSI
🗺️ Map

NGSI resource </> Embed

Download

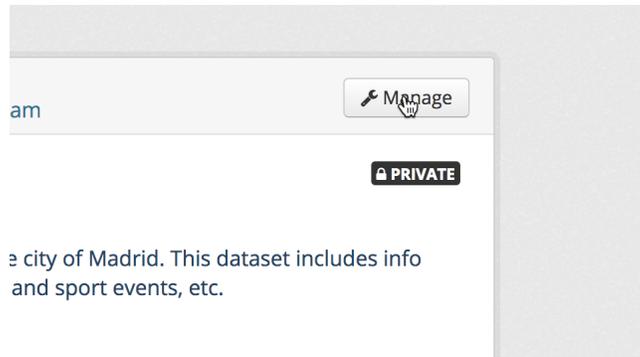
```
[
  {
    "id": "11507",
    "type": "CityIssue",
    "category": {
      "type": "List",
      "value": [
        "construction"
      ],
      "metadata": {}
    },
    "dateEnd": {
      "type": "DateTime",
      "value": "2017-12-07T22:59:00.00Z",
      "metadata": {}
    },
    "dateStart": {
      "type": "DateTime",
```

1.2.5 BAE Publisher

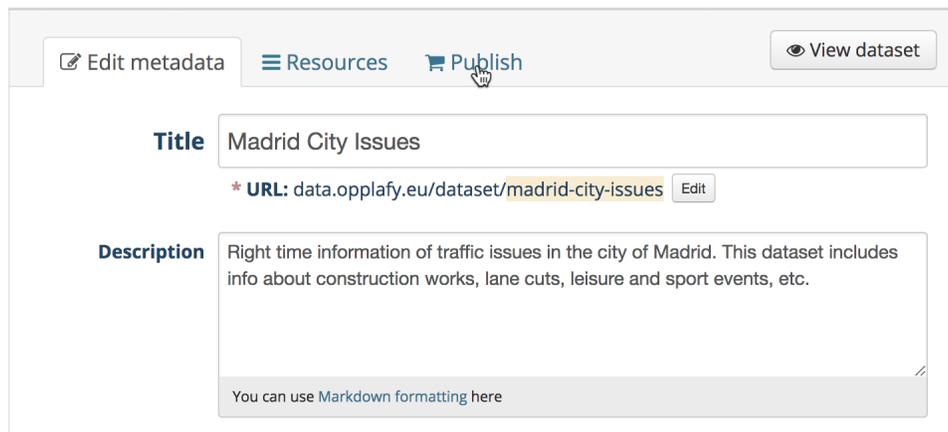
The BAE Publisher extension integrates CKAN with an instance of the Business API Ecosystem GE by providing a form that can be filled in order to create products and offerings of private datasets directly from the CKAN interface.

This extension automatically fills the *Acquire URL* field of the dataset, so it delegates in the Private Dataset extension the access control management in the CKAN portal.

To publish a private dataset as an offering in the BAE, the first step is accessing to its *Management* section.



The publication form is located under the *Publish* tab



The extension accesses the BAE API in order to retrieve some basic information (Available categories and user catalogs) and presents a single form that can be used in order to create a product and a basic offering in the BAE

Edit metadata
Resources
Publish
View dataset

*** Name**

Description

Categories

Catalogs

*** Version**

License

License Description

Acquisition Role

Image Attach an image to the offering. If you do not upload an image, a default one will be used.

Price Price in Euros. Leave it in blank to create a free offering.

The fields to fill are:

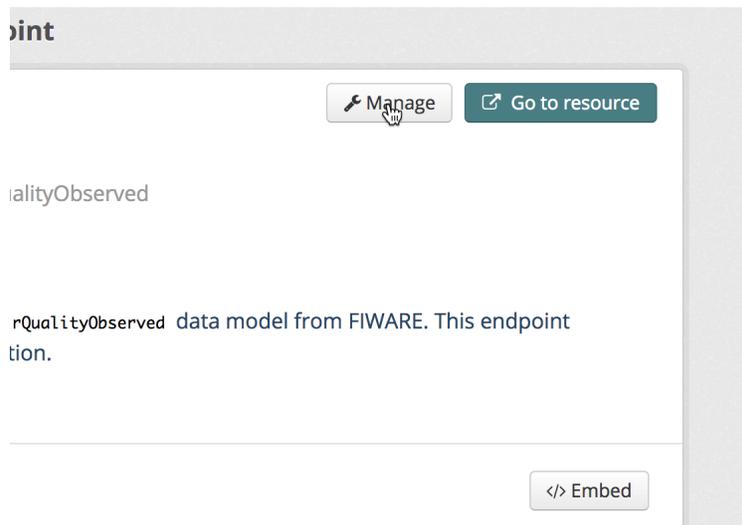
- **Name:** Name of the offering to be created in the BAE
- **Description:** Description of the offering to be created in the BAE
- **Categories:** List of categories for the offering to be created. The available list of categories is retrieved from the BAE
- **Catalog:** Catalog where publishing the offering to be created. The list of available user catalogs are retrieved from the BAE
- **Version:** Version of the offering to be created in the BAE
- **License:** Header of the terms and conditions to be accepted by customers
- **License Description:** Text of the terms and conditions to be accepted by customers
- **Acquisition Role:** If the dataset includes a secured backend service (e.g a Context Broker), role to be granted to the customers so they can access directly to the data service
- **Image:** Image to be displayed with the offering, by default the CKAN logo
- **Price:** Price of the offering or none if the offering is free. For the creation of more complex price models the BAE interface should be used, see the [BAE user guide](#) for details on this topic

1.2.6 WireCloud View

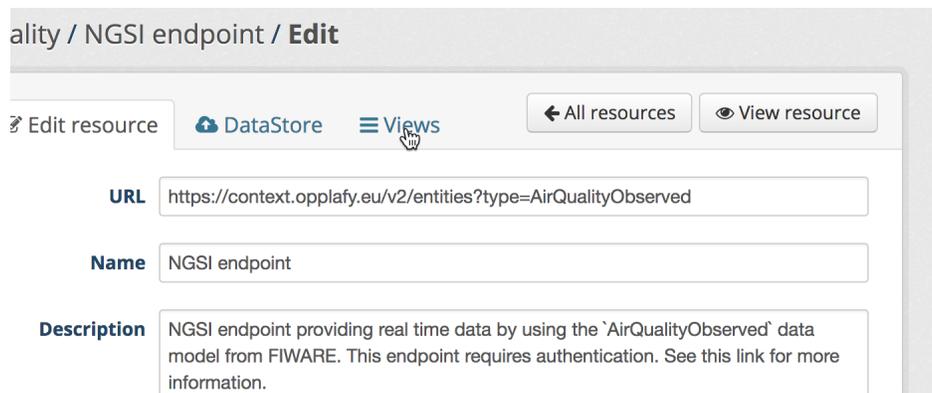
The WireCloud View extension allows to embed a WireCloud dashboard in CKAN as a dataset resource view. This way its is possible to create rich and customizable data visualizations (including right-time NGSI) without the need of

installing new specific CKAN view extensions or restarting the CKAN instance.

The first step for embedding a WireCloud Dashboard is accessing the Management section of the particular data resource.



And accessing to the *Views* tab



The WireCloud view is one of the allowed ones.



For creating the view, it is needed to provide a name and a description. Then, the user can select one of the available visualization dashboards.

✎ Add view
← All views
👁 View view

*** Title**

Description

You can use Markdown formatting here

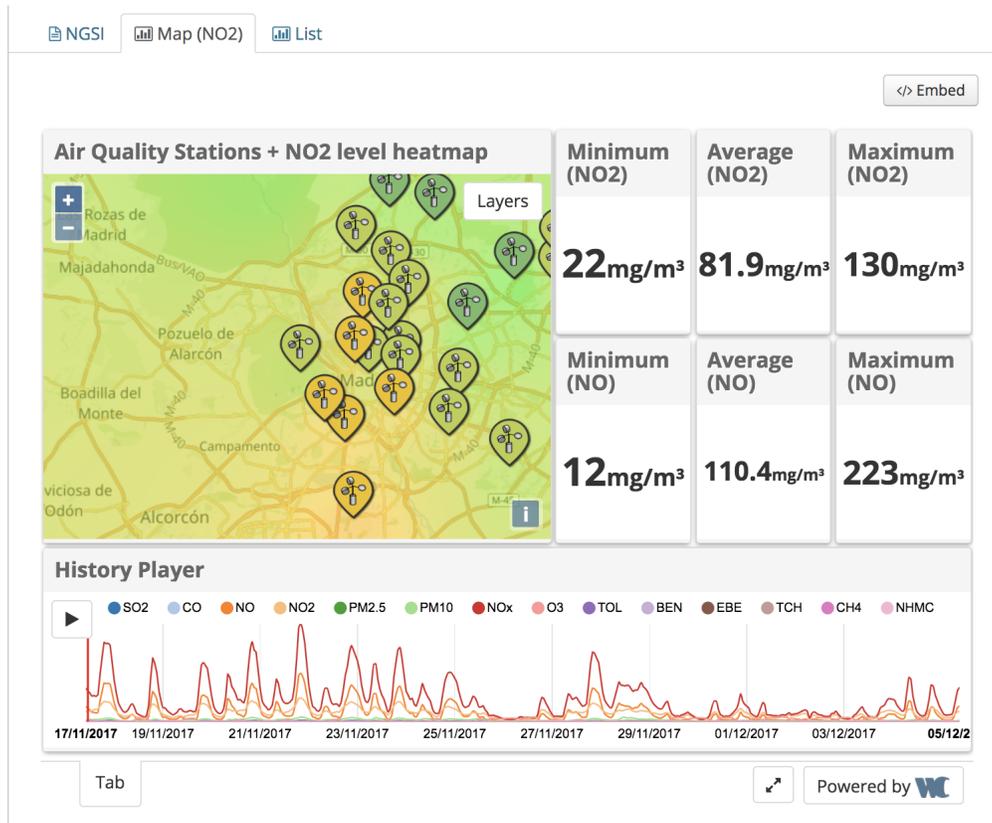
Filters [Add Filter](#)

Dashboard

Create a new dashboard

Preview Add

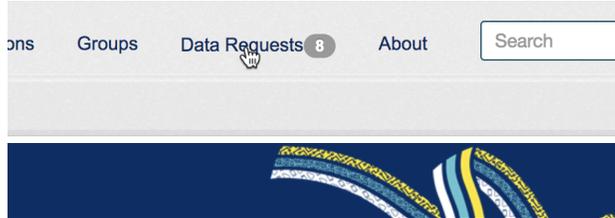
Once created, the dashboard is included as a resource visualization.



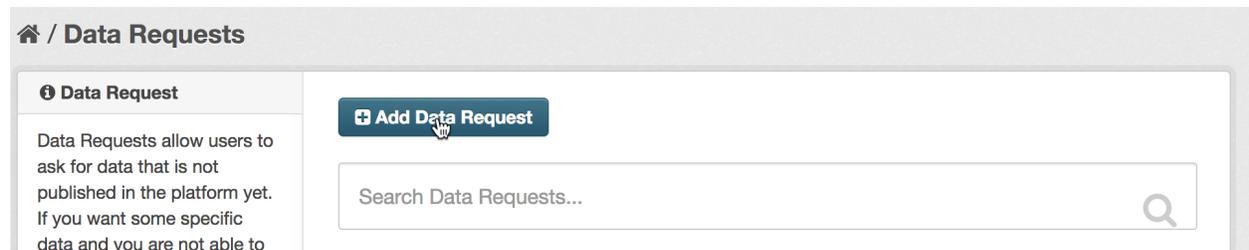
1.2.7 Data Requests

The Data Requests extension allows users of the platform to ask for data which is not yet published in the CKAN instance, while enables admins to manage the lifecycle of that request.

To create a new Data Request, the first step is accessing to the *Data Requests* section included by the extension.



And click on *Add Data Request*.



In the displayed form, it is needed to provide a title for the request and a complete description of the requested data.

Create Data Request

* **Title:**

Description:

You can use Markdown formatting here

Organization:

Once created, the new data request appears in the *Data Requests* section, being possible to be searched.

[+ Add Data Request](#)

Search Data Requests... 

1 data request found Order by: [Newest](#)

[OPEN](#) **Madrid Parking Data**

I would like to have right-time Madrid parking data in NGSI format

 1 22 minutes ago

In addition, it is also possible to view the data request details.

[Data Request](#) [Comments 0](#) [Manage](#) [Close](#)

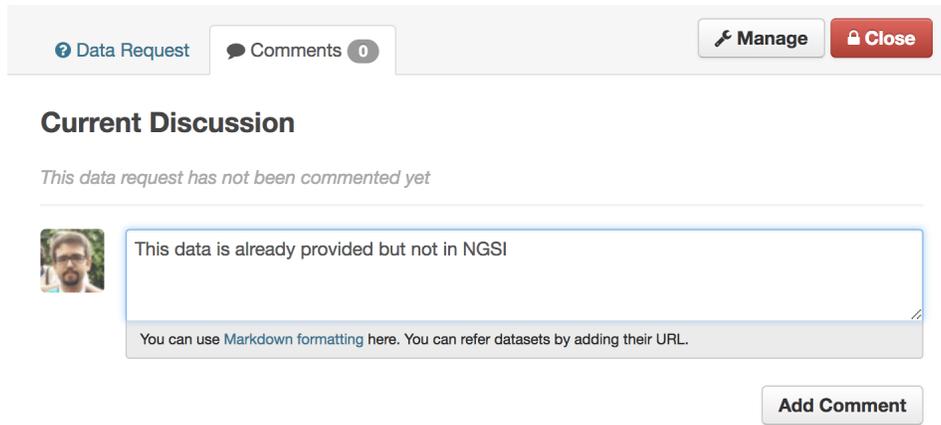
Madrid Parking Data [OPEN](#)

I would like to have right-time Madrid parking data in NGSI format

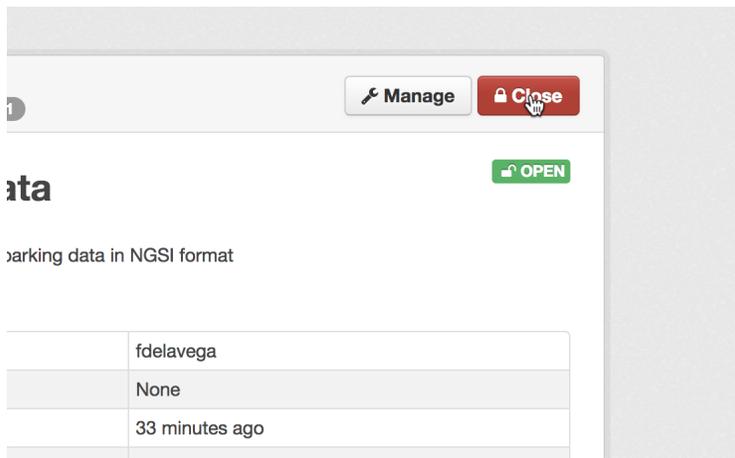
Additional Info

Creator	fdelavega
Organization	None
Created	Just now
Closed	Not closed yet

This extension allows the creation of comments, so it enables the communication between data requesters and data providers in order to make easier the resolution of the request. This option is available in the *Comments* tab of the data request details page.



Once the data has been provided or it has been decided that the request cannot be satisfied, it is possible to close the request by clicking on the *Close* button of the data request details.



Then, the extension will ask for the dataset which satisfies the request. If the data request is being closed because it cannot be satisfied, it is possible to leave this value to *No Dataset*

Close Data Request



Now, the data request appears are closed in the list.

[+ Add Data Request](#)

Search Data Requests... 

1 data request found Order by:

CLOSED Madrid Parking Data

I would like to have right-time Madrid parking data in NGSI format

 1 37 minutes ago

Finally, it is possible to manage requests updating the title or description or deleting it from CKAN.

Edit Data Request

*** Title:**

Description:

You can use [Markdown formatting here](#)

Organization:

[Delete](#) [Update Data Request](#)