حكومة قطر الرقمية
Qatar Digital Government

Open Data Portal User Guide

Table of Contents

Navigating the catalog

The dataset catalog of MOTC portal is accessible from the front office for all users.

On the left side (1), are displayed:

- the number of available datasets,
- a catalog sort option to change the order of the datasets,
- a variety of catalog filters to find specific datasets,
- different possibilities to download the catalog.

On the right side (2), are displayed the datasets. Each dataset is represented by a rectangle called a "catalog card". On each card, the dataset's main metadata are displayed, along with quick links to visualize the data.

Sorting the catalog

By default, a dataset catalog is sorted by last modification date (i.e. the most recently modified datasets are displayed in first position, at the top of the catalog).

It is possible to change in which order the catalog cards are displayed with the catalog sorting option below the number of datasets, on the left side of the interface.
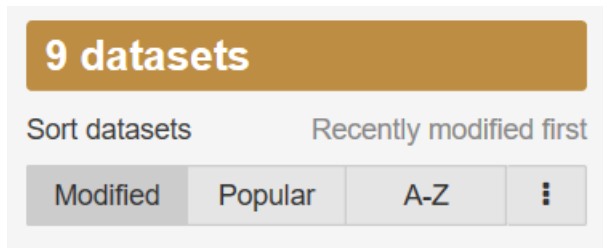
The catalog can be sorted by a total of 10 methods. These methods are based on 5 sorting criterias, each being available in both increasing and decreasing order.

- Dataset title: to sort the datasets in alphabetical order or reverse alphabetical order
- Modification date: to sort the datasets by most recent or oldest modification date
- Number of records: to sort the datasets by biggest or fewest number of records
- Number of downloads: to sort the datasets by biggest or fewest number of downloads
- Popularity: to sort the datasets from the most or least popular

To modify the sorting method, 2 possibilities:

A. Click one of the 3 most popular methods: "Modified", "Popular", "A-Z".
B. Use one of the other methods:

1. Click the button to display all available sorting methods.
2. Click the chosen sorting method in the list.

Once the new sorting method is chosen, the catalog automatically updates.

Filtering the catalog

When navigating a catalog with only a few datasets, it is easy to scroll down and check all the datasets available. But when navigating catalogs with dozens, even hundreds of datasets, scrolling down isn't an option. As an end-user on an open data portal, the reason for visiting that portal may not be just curiosity. The reason for being on the portal may be for a precise purpose; to find specific data or to search for data from a specific topic or time period.

For these reasons and more, a variety of filtering options are available, below the number of datasets and the catalog sorting option, on the left side of the interface.

Qatar Digital Government
حكــومـة قطـر الـرقـمـيـة

Filters

Find a dataset... 🔍

Theme

Business and Economic Development	3
Energy	2
Environment	2
Culture, Heritage	1
Economy	1
Tourism	1
› *More*	

View

⬛ Analyze	9
🌐 Map	5
🎨 Custom view	1
🖼 Image	1

Modified

2017	1
2018	8

Publisher

The World Bank	3
Ministry of Energy & Industry	1
Ministry of Finance	1
Ministry of Tourism	1
Ministry of Transport & Communication	1
World Intellectual Property Organization	1

It is possible to filter:

- by textual research, using the search bar

- by metadata, using the available metadata-based filters

Using the search bar

The search bar should be used to make textual searches in the catalog. When doing a textual search, the platform will look for a match between the search and the metadata from all the available datasets (title, description, keywords etc.). The catalog will be filtered to only display those matching datasets.

To make a textual search:

1. Click on the search bar area.
2. Type the word to search in the datasets.
3. Press enter for the search to be taken into account, and the catalog filtered according to the typed word.

Note: It is possible to make more advanced searches with the search bar, using Query language

Using the selectable filters

Under the search bar there is a list of selectable filters sorted by categories based on the datasets metadata. These filters should be used to filter the datasets by view (visualizations such as Analyze, Map, Calendar etc.), date, publisher, keyword, theme, language, and any other metadata available as filter category. When searching with the selectable filters, the platform will look for a match between the selected values from the filter categories and the values set in the metadata of the available datasets. The catalog will be filtered to only display those matching datasets.

To use the selectable filters:

1. Choose a category of filters.
2. Under the name of the category of filters are listed values (which are the metadata set for the datasets). Click one of the values. The other values disappear and the catalog directly updates to only display datasets matching the newly applied filter.

Downloading the catalog

Under all the filters are different possibilities to download the catalog. This does not allow to download all of the datasets from the catalog (or the filtered ones), but it allows to download an index of the datasets from the catalog along with their main metadata.

Download catalog

This catalog is available as:

- XLS
- CSV
- RDF (Turtle)
- RDF (XML)
- RSS feed
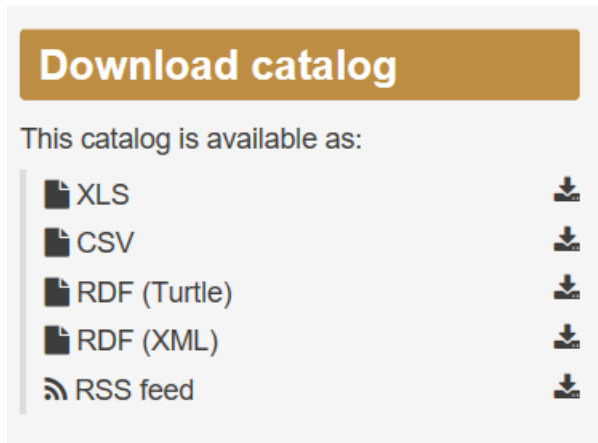
It is possible to download the catalog in the following formats:

- XLS
- CSV
- RDF (Turtle)
- RDF (XML)
- RSS feed

Exploring a dataset from the catalog

Once the right dataset has been found in the catalog, through the searching and filtering options or by scrolling down, it should be time to explore this dataset to see the data and their visualizations.

From the catalog interface, only the catalog card of the dataset is displayed. The catalog card should show the following information:

- a pictogram, to represent the theme of the dataset
- the title of the dataset
- the description of the dataset
- the name of the publisher
- the license of the dataset
- the keywords of the dataset
- links to the available visualizations of the dataset (e.g. Table, Map, etc.)
- links to tabs to share or reuse the dataset (e.g. Export, API)

To see more of the dataset, either click the catalog card or any of the links to directly access a precise visualization or tab. From the interface dedicated to the dataset, it is possible to:

- visualize the data,
- search the data,
- share and reuse the data,
- react and subscribe to the dataset.

Visualizing data

Qatar Open Datar Portal features an in-depth visualization experience for you to explore a dataset in a rich and interactive manner.

The dataset information page displays the dataset metadata like its title, description, source, keywords, license, etc.

Depending on your dataset's columns, you will be able to explore it as a classic table, to project it on a map, or to display it thanks to various charts and widgets. You will also find all the images present in the dataset.

Finally, you will be able to export it in several file formats (depending on the nature of your data) and to consume it through an API.

Displaying data on a map

This can be achieved by switching to the **Map** tab in the dataset explore page. For that tab to be available, your dataset must contain a geopoint or a geoshape field.

Displaying data on a chart

This can be achieved by switching to the **Analyze** tab in the dataset explore page. For that tab to be available, your dataset must contain facets or date/time fields.

Products Imported by Qatar from All Countries
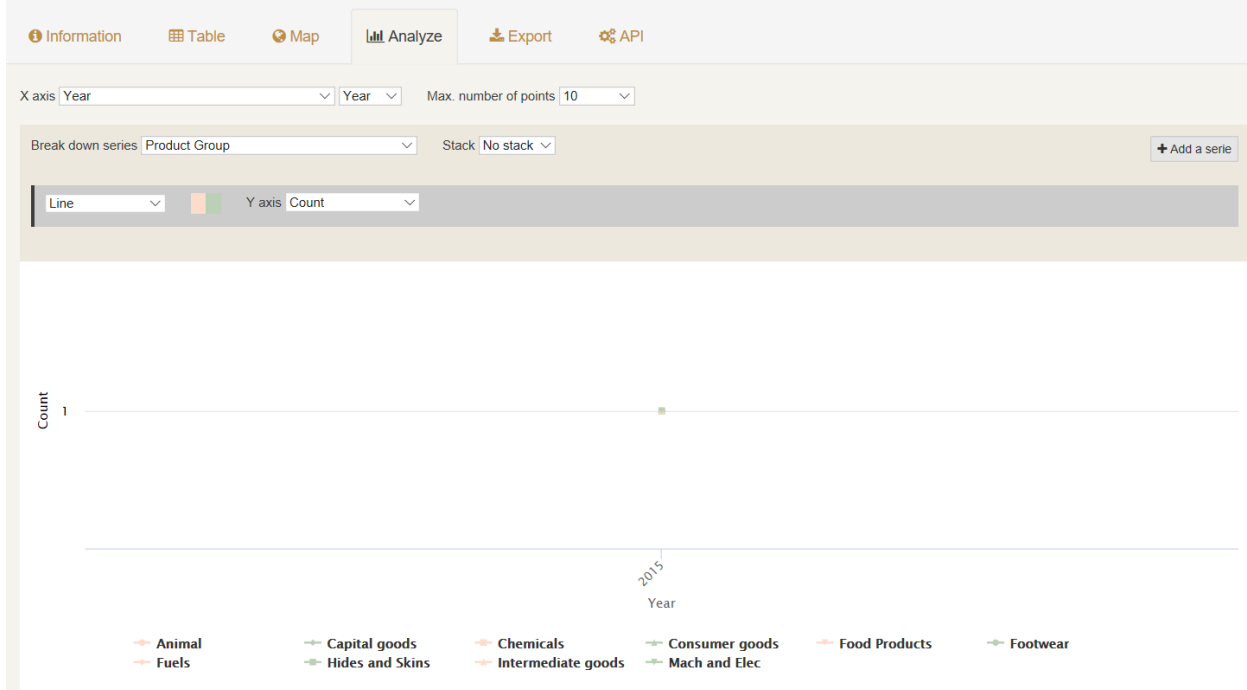
Reusing a visualization

One of the main goals of Qatar Open Data Portal is to have the published data reused by the public, thus producing new analysis or new services. Qatar Open Data Portal motivates users to create their own analysis, studies, and inventions by using the available data, and allows the user to publish their own work on the portal to promote it.

Any visualization built with Qatar Open Data Portal can easily be reused as is.
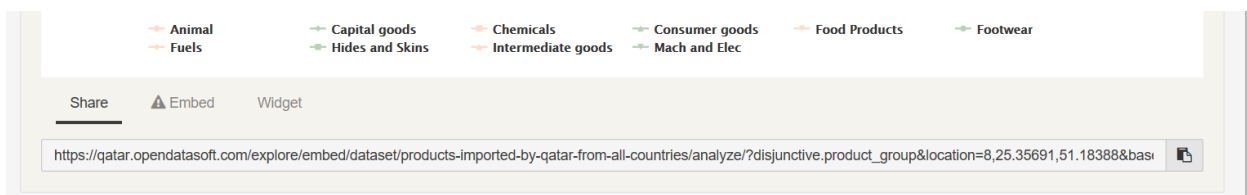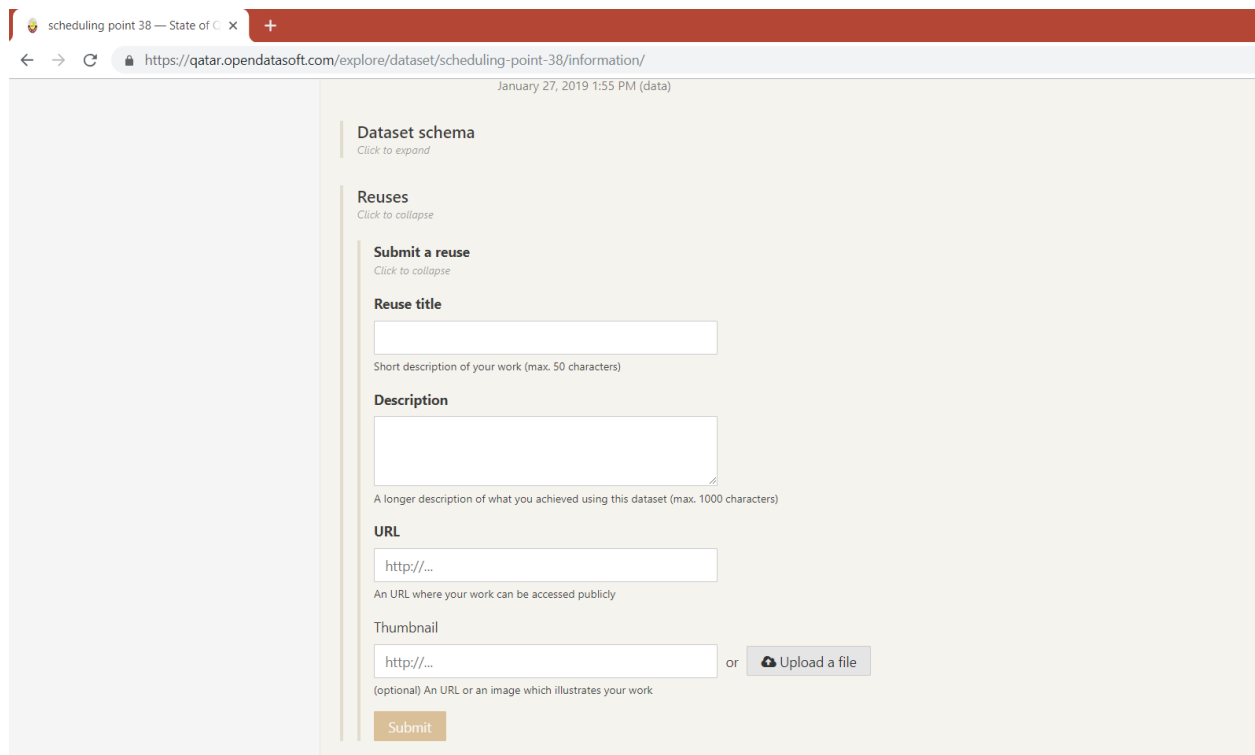
At the bottom of the *Map* tab, there is a widget that allows you to share a visualization you made:

- as a **Share** link, to share a direct access to the visualization,
- as an **Embed** code, to embed the visualization in an iframe on a remote web site,
- as a **Widget** code, to integrate the visualization as a widget in a content page using the OpenDataSoft ods-widgets framework.

Listing reuses

You can list reuses on the dataset page, to help promote both the reuse itself and your data. Reuses section appears at the bottom of the information tab on each dataset page. Additionally, this section will provide a form through which users can submit their reuses and will list all approved reuses.

A user can submit a reuse by filling in the below form:

A reuse request will be sent to the portal admin for approval. Once it is approved it is be displayed under the dataset → information tab → reuses section

Filtering the records

On the left hand side of the dataset page, there is a filtering toolbar very similar to what is described in the paragraph Filtering the catalog.

Searching in the data

Search for records	Query	Example (based on ods-api-monitoring dataset)
containing a value	value	`explore` returns every record containing the string explore
containing several values	value1 AND value2	`action:explore` explore in the column named action
containing several values	value1 AND value2	`explore AND search` returns every record containing both explore and search
containing at least one of the values	value1 OR value2	`action OR explore` returns every record containing either
not containing a value	NOT value	`NOT anonymous` returns every record which does not contain the string anonymous
containing an exact value in a field	#exact(id_field,"value")	`#exact(user_id,"anonymous")` returns every record containing the exact string anonymous in the column

Search for records	Query	Example (based on ods-api-monitoring dataset)
		named user_id
if a field is empty	#null(id_field)	`#null(referer)` returns every record which has no value in the column named referer
where a date's field is anterior to a value	id_date_field<=YYYY/MM/DD	`timestamp<=2016/09` returns every record with a timestamp prior and equal to September 2016
where a date's field is the current date minus a period	id_date_field>#now(days=-value)	`timestamp>#now(days=-7)` returns every record with a timestamp equal to current day minus 7 days
containing a geo field located in a specific area	#distance("latitude,longitude",distance)	`#distance(48.866667,2.333333,1000)` returns every record located at 1 km from the center of Paris

Query Language and Geo Filtering

Query language

The Qatar Open Data Portal query language makes it possible to express complex boolean conditions as a filtering context.

Full-text search

The query language accepts full text queries.

If a given word or compounds is surrounded with double quotes, only exact matches are returned (modulo an accent and case insensitive match).

- `film` returns results that contain film, films, filmography...
- `"film"` only returns the ones containing exactly film.

Boolean expressions

The query language supports the following boolean operators `AND`, `OR` and `NOT`.

Parenthesis can be used to group together expressions and alter the default priority model:

- `NOT`
- `AND`
- `OR`

Samples

- `film OR trees`
- `(film OR trees) AND paris`

Field queries

One of the major feature of the query language is to allow per field filtering. You can use field names as a prefix to your queries to filter the results based on a specific field's value.

For dataset search in the catalog, the list of available fields corresponds exactly to available metadata. By default:

Field Name	Description
publisher	The dataset publisher
title	The dataset title
description	The dataset description

Field Name	Description
license	The dataset license
records_count	The number of records in the dataset
modified	The last modification date of the dataset
language	The language of the dataset (iso code)
theme	The theme of the dataset
references	The references for the dataset

For record search in a dataset, the list of available fields depend on the schema of the dataset. To fetch the list of available fields for a given dataset, you may use the search dataset or lookup dataset APIs.

Multiple operator fields can be used between the field name and the query:

- `:`, `-`, `==`: Return results whose field exactly matches the given value (granted the fields are of text or numeric type)
- `>`, `<`, `>=`, `<=`: Return results whose field values are larger, smaller, larger or equal, smaller or equal to the given value (granted the field is of date or numeric type).
- `[start_date TO end_date]`: Queries Records whose date is between `start_date` and `end_date`.

Date formats can be specified in different formats: simple (`YYYY[[/mm]/dd]`) or ISO 8601 (`YYYY-mm-DDTHH:MM:SS`)

Examples:

- `film_date >= 2002`
- `film_date >= 2013/02/11`
- `film_date: [1950 TO 2000]`
- `film_box_office > 10000 AND film_date < 1965`

Query language functions

Advanced functions can be used in the query language. Function names need to be prefixed with a sharp (`#`) sign.

Function name	Description
now	Returns the current date. This function may be called as a query value for a field. When called without an argument, it will evaluate to the current datetime: `birthdate >= #now()` returns all Records containing a birth date greater or equal to the current datetime. This function can also accept parameters, see below for the `#now` function available parameters.
null	This function may be called specifying a field name as a parameter. It returns the hits for which no value is defined for the specified field. For example `#null(birthdate)`
exact	This function makes it possible to search for records with a field exactly matching a given value. For example, `#exact(firstname, "Marie")` will return records with a field `firstname` containing exactly "Marie" and nothing else.
attr	This function makes it possible to search for records with a field matching a value in the querying user's SAML attributes. For example, `#attr(firstname, user_first_name)`, performed by a user who has a SAML attribute `user_first_name` with a value of "Marie", will return records with a field `firstname` containing "Marie".

Available parameters for the ``#now`` function:

- years, months, weeks, days, hours, minutes, seconds, microseconds: These parameters add time to the current date.

  For example: `#now(years=-1, hours=-1)` returns the current date minus a year and an hour

- year, month, day, hour, minute, second, microsecond: can also be used to specify an absolute date.

  For example: `#now(year=2001)` returns the current time, day and month for year 2001

- weekday: Specifies a day of the week. This parameter accepts either an integer between 0 and 6 (where 0 is Monday and 6 is Sunday) or the first two letters of the day (in English) followed by the cardinal of the first week on which to start the query.

  `#now(weeks=-2, weekday=1)` returns the Tuesday before last.

  `#now(weekday=MO(2))` returns Monday after next.

Exporting data

Qatar Open Data Portal lets you download datasets in various formats. CSV, JSON and Excel formats are always available, but depending on the nature of the data, more options can be available like GeoJSON or Shapefile if the dataset features a geoshape or geopoint field.

To export a dataset, go to the **Export** tab and choose the file format you want to download. If you selected filtering options, another option to download only the selected records will appear.

Using the API

An API (Application Programming Interface) is an interface for programs to communicate and exchange data.

Qatar Open Data Portal provides access to 6 APIs, that can be put into 3 categories:

- ODS Search API v1 and ODS Search API v2: OpenDataSoft's own APIs, used internally by the platform and the ODS widgets
- Triple Pattern Fragments API: ODS API for triple pattern querying over datasets from OpenDataSoft portals
- OData, WFS and CSW, 3 standard protocols supported and provided by OpenDataSoft

All these APIs provide access to any data pushed to the platform, no matter their source or format, as long as the security rules defined by the data owner allow that access.

Qatar Open Data Search

This Search API is used internally by the Qatar Open Data platform and widgets. To develop an application, we recommend using this API.

There are currently 2 versions of the ODS Search API: the v1 and the v2 which is still under development. Listed below, the main differences between the 2 versions.

- The ODS Search API v2 introduces a better implementation of REST principles. Each API entry-point provides links to easily navigate between linked resources (HATEOAS).
- Key changes in the v2:
  - a more complete query language (ODSQL) is available, with new functions and arithmetic expressions
  - special filters (such as geofilter and facet filters) are now directly integrated to ODSQL, in the `where` parameter (i.e `geofilter.distance='42,1,100'` is replaced by `where=distance(geo_field,geom'Point(42 1)',100)`)
  - the aggregation API (former analyze API) now shares the same structure and ODSQL language as the Search API, and is available on both the catalog (to aggregate datasets) and the dataset (to aggregate records) APIs
  - exporters have their own entry-point under `/api/catalog/exports` and `/api/catalog/datasets/<dataset_id>/exports`
  - navigation search (former facet search) and standard search are completely split

Triple Pattern Fragments

The Triple Pattern Fragments (TPF) API enables dataset querying in Resource Description Framework (RDF) format. This work is still in progress and is part of a PhD that aims to improve OpenDataSoft datasets quality using semantic web technologies.

OData, WFS and CSW
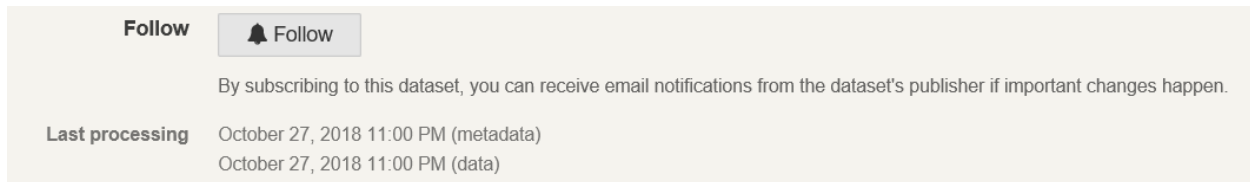
OData, WFS and CSW refer to other standards implemented by the platform. We recommend to users who are not familiar with these 3 APIs to stick to the ODS Search API since it provides the most comprehensive access to the data.

- OData: standard for REST APIs that provides a common language to be used across APIs to perform requests. The trade-off for this norm's generic approach is its higher complexity compared to traditional custom REST APIs. This norm was implemented for the simple operations (searching for datasets and records) ; for the more complex ones (analysis, aggregations, geographic computations etc.) our Qatar Open Data Search APIs (v1 and v2) must be used.
- WFS and CSW: standards focusing on geographic data. They are especially relevant for example to interface the platform with other GIS software.

Following dataset updates

By following a dataset, users can indicate that they are interested in further changes in your dataset: they may have a mobile application that rely on it, use it for academic reasons... As a publisher, you are able to send them notifications about important changes or evolutions, and ensure you keep a relation of trust with your users.

Every user that is logged to the domain is able to follow any dataset they can explore. Users will see a grayed-out button that will encourage them to sign up to be able to subscribe.

Follow	🔔 Follow
	By subscribing to this dataset, you can receive email notifications from the dataset's publisher if important changes happen.
Last processing	October 27, 2018 11:00 PM (metadata)
	October 27, 2018 11:00 PM (data)

Map builder

Overview of Map Builder

Map Builder is a tool to create maps. Map Builder's maps are more advanced than those simply created within the Map tab of a published dataset because you can overlay as many datasets as you want in the same map, creating different data layers. This specificity thus makes these maps great for enriching your data by crossing them together. You can also group your layers and edit their descriptions and titles in order to obtain and share a map finely designed to meet specific needs and purposes.
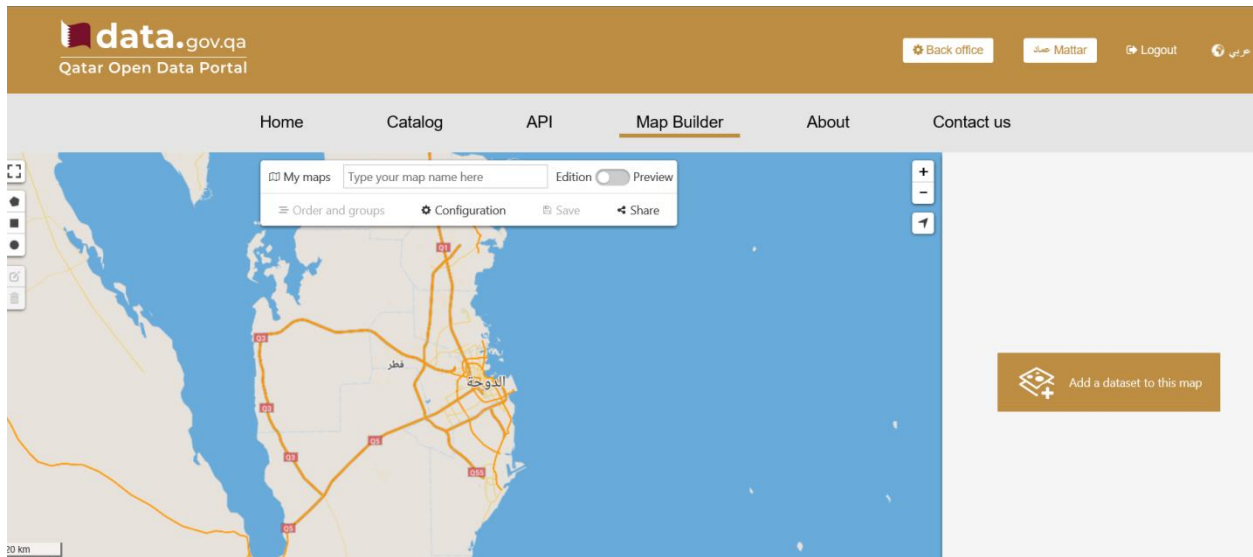
Map Builder has 2 modes:

- **Edition**, to create and configure your map to define what it will look like to the final map viewers
- **Preview**, to see what your map looks like to the final map viewers
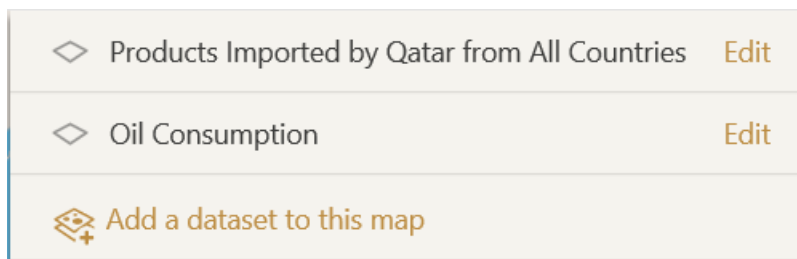
Map Builder, in Edition mode, is also split into 3 parts:

- the layer editing area
- the Map Builder menu
- the interactive, multi-layered map

### Overview of the layer editing area of Map Builder

The layer editing area is located on the right-hand side of your page, right next to the map. Through this part of Map Builder, you can [create a multiple layers map](#).

If you haven't started creating your map, you will only see a link, **Add a dataset to this map**, to add your first dataset and thus start the creation of your first map. Afterward, this section will display:

- all your added datasets
- **Edit**, a link to access the visualization configuration options of the related dataset
- **Add a dataset to this map**, a link to add a new dataset to your map

Overview of the Map Builder menu
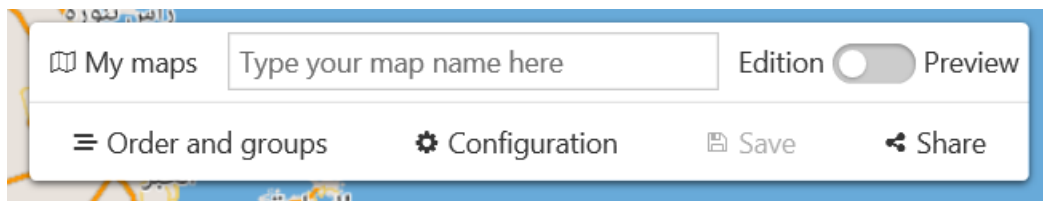
The Map Builder menu is located at the very top of the map. Through this part of Map Builder, you can rename and save a map, manage your maps, reorder and group layers in a map, configure your map and share your map.

It displays the following information, action buttons and submenus:

- **My Maps**, to manage your maps, including your not-yet-saved maps (drafts)
- the title of your map
- a switch button, whether you want to display Map Builder in an edition or in a preview mode
- **Order and groups**, to manage in what order your datasets are displayed and to group your datasets together
- **Configuration**, to define which features will be activated when your map is in preview or view mode
- **Save**, to save your map
- **Share**, to access links to share your map as a link, inside an iframe or as a widget pasted in a page

Overview of the interactive map of Map Builder
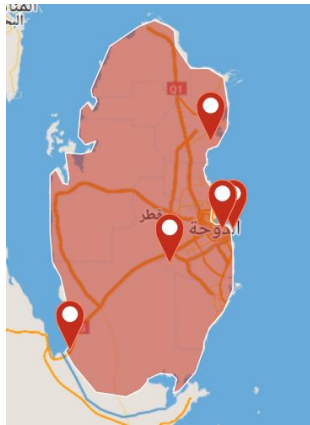
The interactive map displays your data. You can navigate on your Map Builder map to play with this data.

Create a multiple layers map

Add a dataset to your map

1. In the layer editing area, click on **Add a dataset to this map**.
2. Choose the dataset you want to display on your map. You can select a dataset with geographical information either from your own catalog of datasets published on your domain, or from the **All available data** catalog.
3. Click on **Select** if you want to add all data, or **Filter data** if you just want to add part of the data.
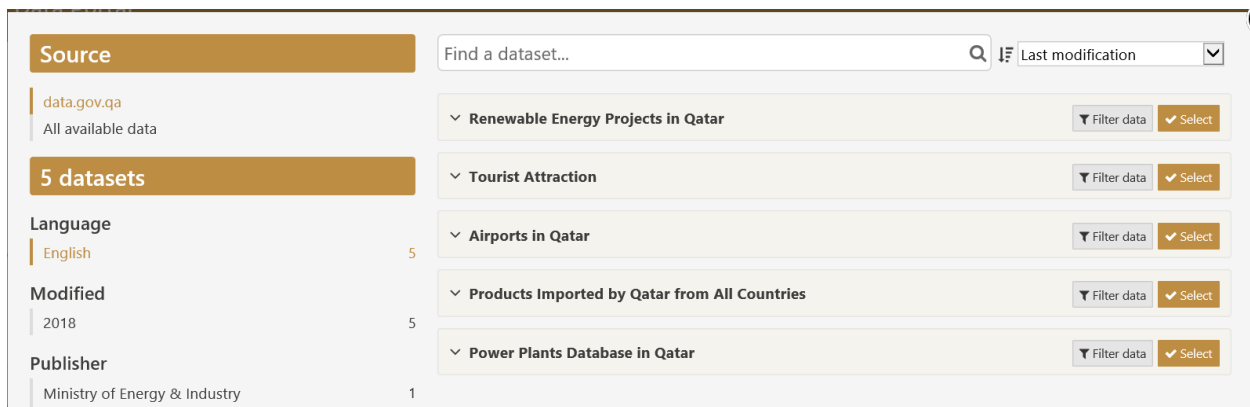
You can add as many datasets as you want on your map, and all of the data they contain can be displayed on that same map, at the same time. This is when the whole concept of layers comes in play. In Map Builder, because you overlay several datasets, each dataset is called a layer. This is why Map Builder allows you to configure your layers to make sure that you can represent many different data on the same map and still make the reading of this map easy and clear.

Edit a layer

The edition of a layer (meaning, a dataset displayed on a map created with Map Builder) has 2 aspects:

- **Display**: the style, in other words, how the data is displayed on the map
- **Information**: the information and caption that go with the dataset to describe it better
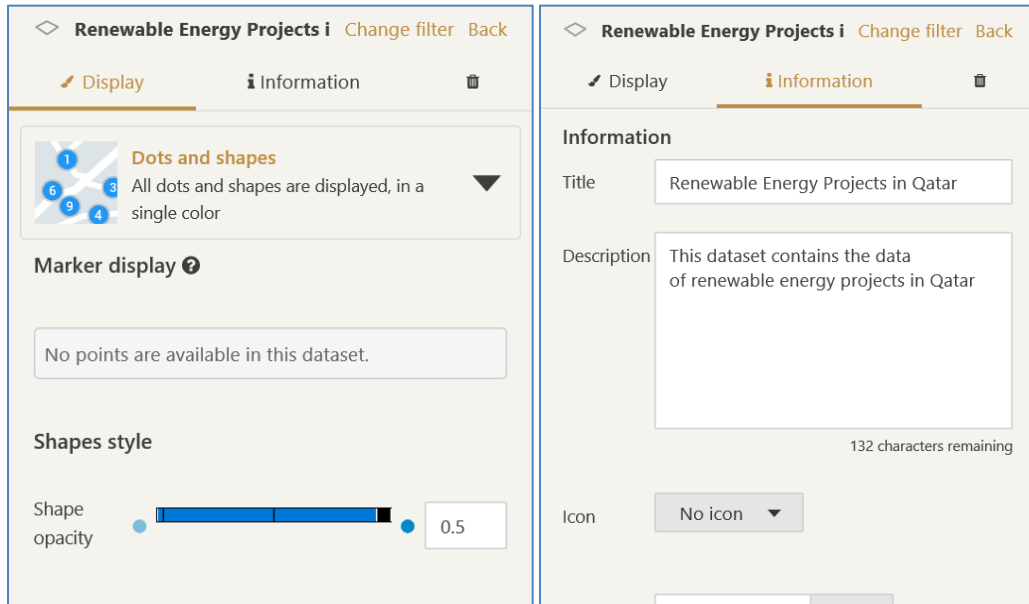
Edit style

1. Click on the **Display** tab. The menu with all style configurations drops down.
2. Choose a visualization mode and configure your layer as you like.

The style configurations entirely depend on the visualization mode you choose to apply to your layer. There are 5 different modes:

Note: The style configurations displayed for each visualization mode also depend on your dataset. Depending on the dataset you added to your Map Builder map, you may not see all the configurations explained in this documentation.

Dots and shapes

The Dots and shapes visualization mode simply displays your data as markers (dots or icons) or as shapes. It is the most common visualization mode.

Style configuration	Description
Marker display *What your data will look like on the map*	**Dot**: simple plain circle. Dots are recommended for high density datasets. **Icon**: themed icon (available in the library). **Map marker (with an optional icon)**: standard drop-shaped map marker, with an optional themed icon (available in the library) inside. If you choose the **Icon** or the **Map marker (with an optional icon)** mode: . In the **Shapes style** section below the **Market display** section, click on the **Icon** menu.

26

Style configuration	Description
	. Choose the icon which will represent your data the best. . Click on the **Select icon** button.
Shapes style *Customize your markers and shapes*	**Icon**: choose an icon for both the **Icon** and **Map marker (with an optional icon)** modes (see procedure above). **Size**: make your dots/icons larger or smaller. **Point opacity**: if your markers are dots or icons - add transparency to these markers. **Shape opacity**: if your data are represented as shapes instead of markers (dots or icons) - increase or decrease their transparency. **Line thickness**: if your data are represented as lines - increase or decrease their thickness. For all the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each slider. **Color**: change the color of your markers, shapes and lines. To change the color: . Click on the color menu. . Choose the color you want from the Nice colors or with the Color picker. . Click on the **Select color** button.
Shapes border *Customize the border of your markers and shapes*	Note: Borders are only available for dots and shapes, icons and map markers do not have borders. **Pattern**: choose the pattern of the border, whether you want it to be a simple line, or a dashed or dotted line. For the **Pattern** option, simply click on the provided menu and choose the one you want. **Thickness**: make the border of your shapes thicker or thinner.

Style configuration	Description
	Opacity: increase or decrease the transparency of the border. For the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each line. **Color**: change the color of the border. To change the color: . Click on the color menu. . Choose the color you want from the *Nice colors* or with the *Color picker*. . Click on the **Select color** button.

Cluster

The Cluster visualization mode groups your data into clusters instead of displaying them separately and individually. Clusters are grouped according to a chosen type of calculation: linear or logarithmic.

Style configuration	Description
Cluster style *What your cluster will look like on the*	**Min size**: choose the size of the smallest clusters. **Max size**: choose the size of the biggest clusters. **Opacity**: increase or decrease the transparency of your clusters. For all the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each

Style configuration	Description
map	slider. **Color**: change the color of the border. To change the color: . Click on the color menu. . Choose the color you want from the *Nice colors* or with the *Color picker*. . Click on the **Select color** button.
Cluster border *Customize the border of your clusters*	**Thickness**: make the border of your clusters more or less thick. **Opacity**: add transparency to the border. For the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each slider. **Color**: change the color of the border. To change the color: . Click on the color menu. . Choose the color you want from the *Nice colors* or with the *Color picker*. . Click on the **Select color** button.
Aggregation operations *Choose the base operation for your clusters*	Choose, out of the available operations, which one you want your clusters to be based on. The result value will be displayed inside the cluster. • **Count**: how many markers are in the area. • **Average**: the average of a selected field from the dataset. • **Sum**: the total of a selected field from the dataset. • **Minimum**: the minimum of a selected field from the dataset. • **Maximum**: the maximum of a selected field from the dataset.

Style configuration	Description
	• **Standard deviation**: the standard deviation of a selected field from the dataset, to indicate whether the values are close to the average or not.
Computing *Choose how the clusters size is calculated*	**Linear**: for your values ranges to be the same size, evenly spread from the minimum to the maximum (e.g. 1, 2, 3, 4). **Logarithmic**: for your values to vary in size and spread from a very large spectrum (e.g. 1, 10, 100).

Choropleth

The Choropleth visualization mode displays data as markers (dots or icons) or as shapes. It also enhances the visualization with colors. The markers and/or shapes are colored according to a color scheme, defined by a values range based on a chosen variable. The Choropleth mode allows the creation of a thematic layer.

Style configuration	Description
Marker display *What your data will look like on the map*	**Dot**: simple plain circle. Dots are recommended for high density datasets. **Icon**: themed icon (available in the library). **Map marker (with an optional icon)**: standard drop-shaped map marker, with an optional themed icon (available in the library) inside. If you choose the **Icon** or the **Map marker (with an optional icon)** mode: . In the **Shapes style** section below the **Market display** section, click on the **Icon** menu. . Choose the icon which will represent your data the best. . Click on the **Select icon** button.
Shapes	**Icon**: choose an icon for both the **Icon** and **Map marker (with an optional**

Style configuration	Description
style *Customize your dots and shapes*	**icon)** modes (see procedure above). **Size**: make your dots/icons larger or smaller. **Point opacity**: if your markers are dots or icons - add transparency to these markers. **Shape opacity**: if your data are represented as shapes instead of markers (dots or icons) - increase or decrease their transparency. **Line thickness**: if your data are represented as lines - increase or decrease their thickness. For all the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each slider. **Palette**: define on which field and operation your choropleth layer will be based on, and the color of your markers, shapes and/or lines. . Choose whether you want to base the color of your shape on the **Value of a field** or on the **Aggregation on a field**. . Depending on the choice you just made for your shape's color, choose the field which value you want to use, or decide which aggregation operation you want to apply on your chosen field. . Define the **Values range**. . Define the number of **Tiers**, meaning in how many categories your values range will be divided. . Choose a **computing mode**: **linear** (for your values ranges to be the same size, evenly spread from the minimum to the maximum), **logarithmic** (for your values to vary in size and spread from a very large spectrum) or **custom**. . Choose the **colors range** by clicking the colors menu and selecting your colors. . Choose a **gradient type** (RGB, Lab, HSL or Lch), meaning a way to calculate the color gradient between the 2 colors you just chose
Shapes border *Customize the border of*	Note: Borders are only available for dots and shapes, icons and map markers do not have borders. • **Pattern**: choose the pattern of the border, whether you want it to be

Style configuration	Description
your shapes	a simple line, or a dashed or dotted line. For the **Pattern** option, simply click on the provided menu and choose the one you want. **Thickness**: make the border of your shapes thicker or thinner. **Opacity**: increase or decrease the transparency of the border. For the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each line. **Color**: change the color of the border. To change the color: - Click on the color menu. - Choose the color you want from the *Nice colors* or with the *Color picker*. - Click on the **Select color** button.

Color by category

The Color by category visualization mode displays data as markers (dots or icons) or as shapes. It also enhances the visualization with colors. The markers and/or shapes are colored according to a color scheme based on defined categories - which can be numerical or not, just defined by a certain word. The Color by category mode allows the creation of a categorized layer.

Style configuration	Description
Marker display	**Dot**: simple plain circle. Dots are recommended for high density datasets.

Style configuration	Description
What your data will look like on the map	**Icon**: themed icon (available in the library). **Map marker (with an optional icon)**: standard drop-shaped map marker, with an optional themed icon (available in the library) inside. If you choose the **Icon** or the **Map marker (with an optional icon)** mode: . In the **Shapes style** section below the **Market display** section, click on the **Icon** menu. . Choose the icon which will represent your data the best. . Click on the **Select icon** button.
Shapes style *Customize your dots and shapes*	**Icon**: choose an icon for both the **Icon** and **Map marker (with an optional icon)** modes (see procedure above). **Size**: make your dots/icons larger or smaller. **Point opacity**: if your markers are dots or icons - add transparency to these markers. **Shape opacity**: if your data are represented as shapes instead of markers (dots or icons) - increase or decrease their transparency. **Line thickness**: if your data are represented as lines - increase or decrease their thickness. For all the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each slider. **Palette**: choose whether you want: to create a **Custom palette based on a field's value** . Click the Brush icon to access the custom palette interface. . Choose the field you want to categories to be based on. . Choose whether or not you want to enable the creation of an additional category. . Choose a color for each category. . Click the **Apply** button. a palette **Generated with colors contained in a field** (in case your dataset already contains a field which provides color information: an

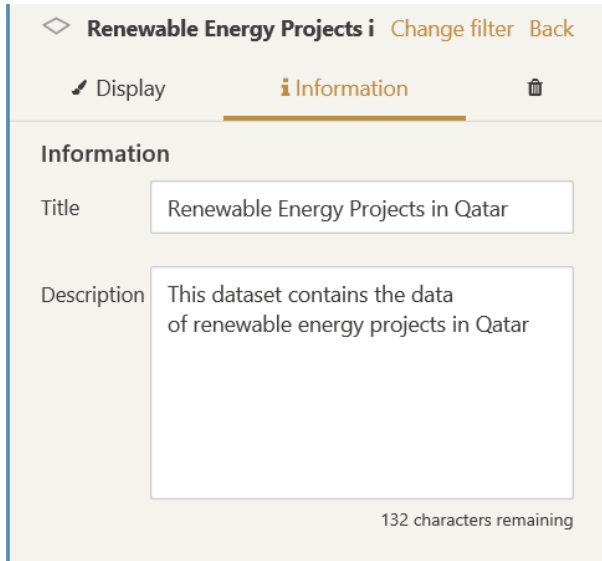| Style configuration	Description
	hexadecimal color code for instance). In that case, simply choose the right field and the colors will automatically appear on the corresponding markers and/or dots of your map.
Shapes border *Customize the border of your markers and shapes*	Note: Borders are only available for dots and shapes, icons and map markers do not have borders. **Pattern**: choose the pattern of the border, whether you want it to be a simple line, or a dashed or dotted line. For the **Pattern** option, simply click on the provided menu and choose the one you want. **Thickness**: make the border of your shapes thicker or thinner. **Opacity**: increase or decrease the transparency of the border. For the options above, simply drag the dot along the slider or directly change the numbers displayed on the right of each line. **Color**: change the color of the border. To change the color: . Click on the color menu. . Choose the color you want from the *Nice colors* or with the *Color picker*. . Click on the **Select color** button.

Heatmap

The Heatmap visualization mode is not about markers or shapes. Instead, this mode groups data and displays them as a span of colors to show the differences of intensity of one variable but at different spots in the map.

Style configuration	Description
Aggregation operations *Choose on what operation your heatmap is based*	Choose, out of the available operations depending on your dataset, which one you want your heatmap to be based on. **Count**: how many markers are in the area. **Average**: the average of a selected field from the dataset. **Sum**: the total of a selected field from the dataset. **Minimum**: the minimum of a selected field from the dataset. **Maximum**: the maximum of a selected field from the dataset. **Standard deviation**: the standard deviation of a selected field from the dataset, to indicate whether the values are close to the average or not.
Computing *Choose how is calculated the progression of your heatmap*	**Linear**: for your values ranges to be the same size, evenly spread from the minimum to the maximum (e.g. 1, 2, 3, 4). **Logarithmic**: for your values to vary in size and spread from a very large spectrum (e.g. 1, 10, 100).
Colors *Choose the color palette of your heatmap*	• Click on the **Edit palette** button. • Choose your 5 colors. • Click on the **Apply** button.

Add and edit information

1. Click the **Information** tab.

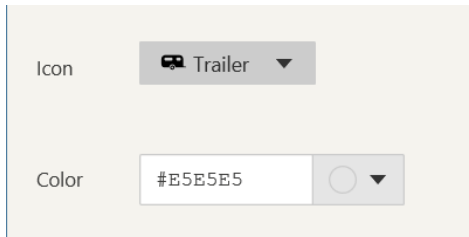*Title and description*

2. Rename your layer in the **Title** area, to make it clearer for the map viewers.
3. Add a **Description** in the provided area.

   Note: If you do not add your own description, it is the default one that will be displayed.
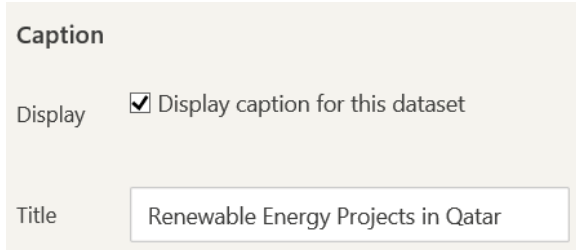
Icon

   You can add an icon to your layer, which will be displayed next to the layer's title.

4. Click on the **Icon** drop-down menu to choose the icon that represents your layer the best.
5. Click on the **Color** menu to add a color to the layer's icon.

Add and edit a caption

Caption

Display   ☑ Display caption for this dataset

Title   Renewable Energy Projects in Qatar

Captions are not mandatory however they are recommended because they add more information to your dataset, which means more clarity to the reading of your map.

1. Click on the **Caption** tab.
2. If it's not already the case, tick the **Display caption for this dataset** option.

   Note: Captions are activated by default. If you do not want to display a caption for your dataset, simply untick the **Display caption for this dataset** option.

3. In the provided area, write a **Title** for your layer's caption.

Delete a layer

◇ **Renewable Energy Projects in (**   Change filter   Back

   ✎ Display   **i** Information   🗑

Remove dataset from the map
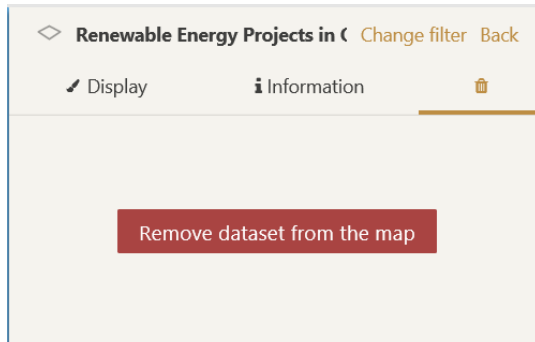
1. Click on the 🗑 icon.
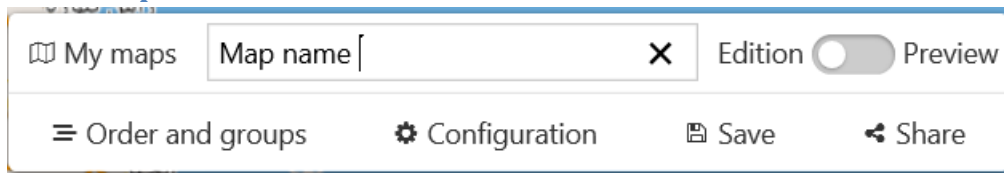2. Click on the red **Remove dataset from the map** button.

Rename and save a map

Rename a map
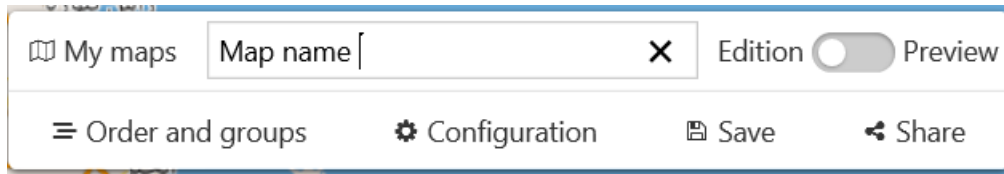
1. In the middle of the first row of the Map Builder menu, enter the name of your map in the provided area.
2. Click on the **Save** button.

Save a map

To save a map while working on it or after you're done, simply click on the **Save** button in the second row of the Map Builder menu. The list of all your saved maps is in the **Saved maps** tab of the **My Maps** interface (accessible from the Map Builder menu).

Manage your maps¶

Overview of the maps management interface: My Maps

All maps management actions (except saving and renaming, see Rename and save a map) are made from the **My Maps** interface. This interface is accessible from the Map Builder menu displayed at the very top of the map.

The interface of **My Maps** is composed of:

- the **New map** button, to create a new map
- 2 submenus:
- **Saved maps**, a list of your named and manually saved maps
- **Drafts**, a list of your unnamed and automatically saved by Map Builder maps
- a search bar to find a map more easily in case you have quite a long list of maps

The **Saved maps** and **Drafts** submenus display information the same way. In each row of each tab, you will find:

- **Open map**, a link to open the selected map in the same tab
- information on the map, such as title, URL, and number of datasets
- reading and writing rights on the map
- **Duplicate**, a button to duplicate the map
- 🗑 to delete the map

Duplicate a map

1. Find the map or draft you want to duplicate, in the **Saved maps** or **Drafts** tab.
2. Click on the **Duplicate** button.

Delete a map

1. Find the map or draft you want to delete, in the **Saved maps** or **Drafts** tab.
2. Click on the 🗑 button. A *Delete map?* window appears.
3. Click on the red **Delete map** button.

Reorder and group layers in a map

Overview of the Order and groups interface

You can reorder your layers and group some of them together.

- Reordering layers can be handy, for instance if some of your layers are more important or demanded as others. You can thus move the most important and/or demanded layers at the very beginning of the list of layers, so that they can be found more easily.
- Grouping layers can also come in handy. For example, you might want to display different layers on your map but that need to be displayed together to be really relevant. If you group these layers, you make them stick together: or they're all displayed at the same time, or they're all hidden.
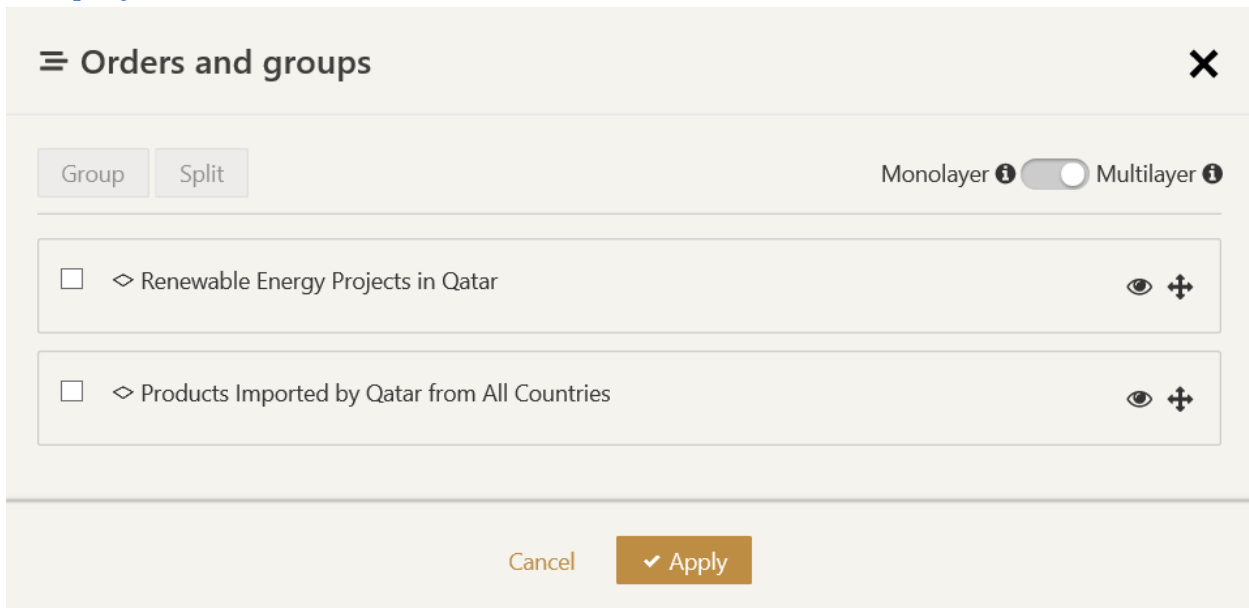
Order and groups management actions are made from the **Order and groups** interface. This interface is accessible from the Map Builder menu displayed at the very top of the map.
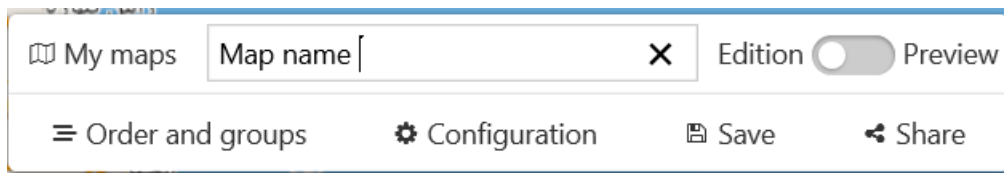
The interface of **Order and Groups** is composed of:

- **Group** button, to group layers
- **Split** button, to split groups
- a switch button, whether you want to be able to see several layers on your map (**Multilayer**) or just one at a time (**Monolayer**)
- the complete list of the layers of your map, with the following action buttons:
  - a checkbox to select a layer, useful to group layers or split groups
  - 👁 to make a layer visible or not
  - ✛ to move a layer when reordering

Group layers

1. Tick the boxes of the layers you want to group together.
2. Click on the **Group** button.
3. Click on the **Apply** button for your changes to be taken into account.

Split groups of layers
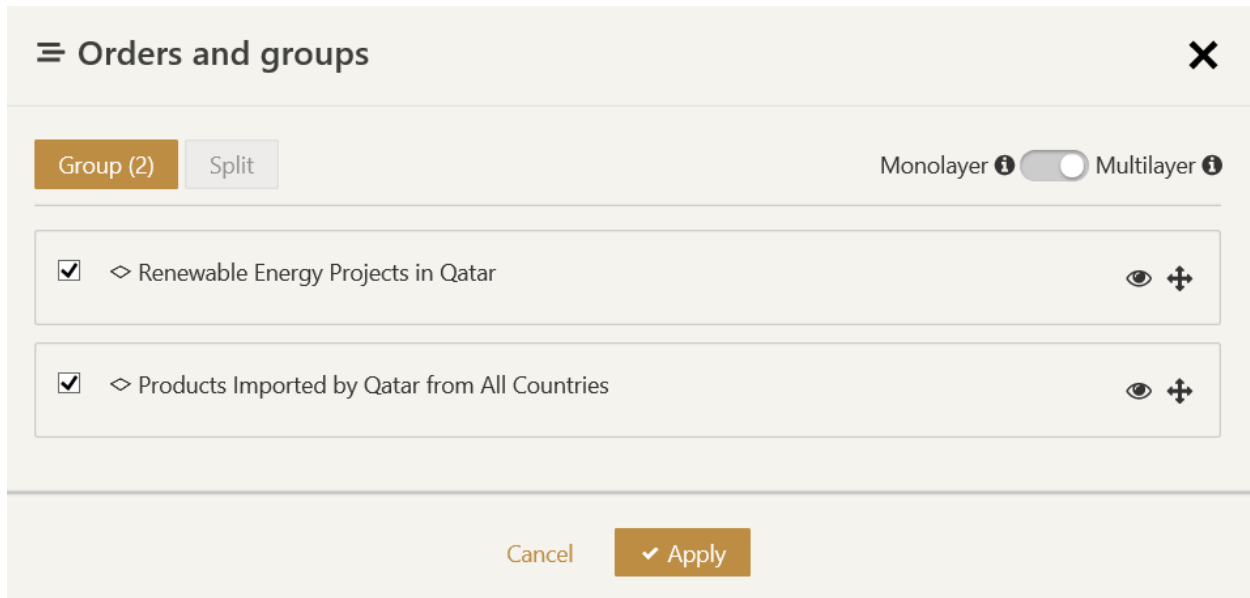
1. Tick the box of the group you want to split.
2. Click on the **Split** button.
3. Click on the **Apply** button for your changes to be taken into account.

Configure your map

Overview of the Configuration interface

From the **Configuration** interface, you can configure your map as seen in view or preview mode. This interface is accessible from the Map Builder menu displayed at the very top of the map.
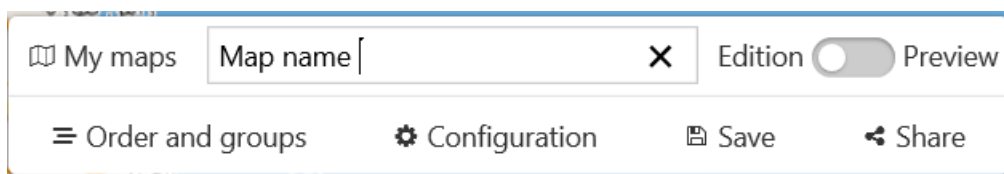
The interface of **Configuration** contains:

- the **URL** of your map
- a **Display search box** checkbox, to have the possibility to make research on your map
- a **Display full screen in view mode** checkbox, to display ⌐⌐on your map and thus have the possibility to display your map in full screen

- a **Geolocation** checkbox, to display ➶ on your map and thus have the possibility to focus the map on where you're located
- an **Auto-geolocation** checkbox, for the map to automatically focus on where you're located
- a **Display layers and groups visibility control** checkbox, to have the possibility to play with layers and groups of layers (hide them or display them) on the map

Share your map

Overview of the Share interface

You can share your map thanks to links and HTML codes related to your map, all available in the **Share** interface. This interface is accessible from the Map Builder menu displayed at the very top of the map.

The interface of **Share** is composed of 3 separate sections:
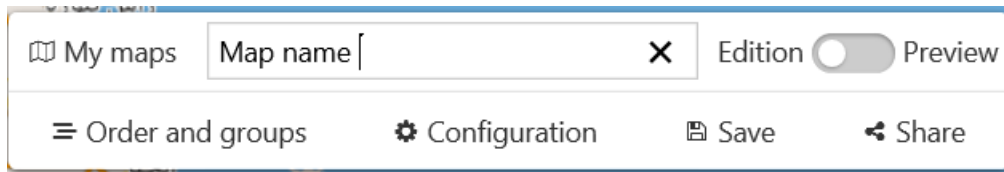
- **Direct link**, a simple direct URL to your map, through the Map Builder of your domain. You could for instance use a direct link to send your Map Builder map by email.
- **Embed map**, to embed your map as an iframe, for example in a blog post.
- **Use map as a widget**, to embed your map as a widget, for example within a page of your Qatar Open Data portal or using our Widgets library.

Share ✕

Direct link

URL

> https://qatar.opendatasoft.com/map/map_name/

Use this URL to share your map or post it on social networks.

Embed map

Map size (in pixels)

> 800 x 600 ⇕

Options
☐ Lock map's zoom and position
☐ Enable zooming in and out with the mouse wheel

iFrame code

> <iframe frameborder="0" width="800" height="600" src="https://qatar.opendatasoft.com/map/embed/map

Paste this code in a webpage to embed your map

Use map as widget

Widget code

```
<ods-dataset-context
context="renewableenergyprojectsinqatar0,productsimportedbyqatarfromallcountries"
renewableenergyprojectsinqatar0-dataset="renewable-energy-projects-in-qatar0"
productsimportedbyqatarfromallcountries-dataset="products-imported-by-qatar-from-all-countries">
    <ods-map no-refit="true" scroll-wheel-zoom="false" display-control="true" search-box="true"
toolbar-fullscreen="true" toolbar-geolocation="true" basemap="jawg.streets"
```

Navigate on a Map Builder map

Overview of the interactive map of Map Builder

The interactive map contains the same buttons as in any Qatar Open Data Portal map:
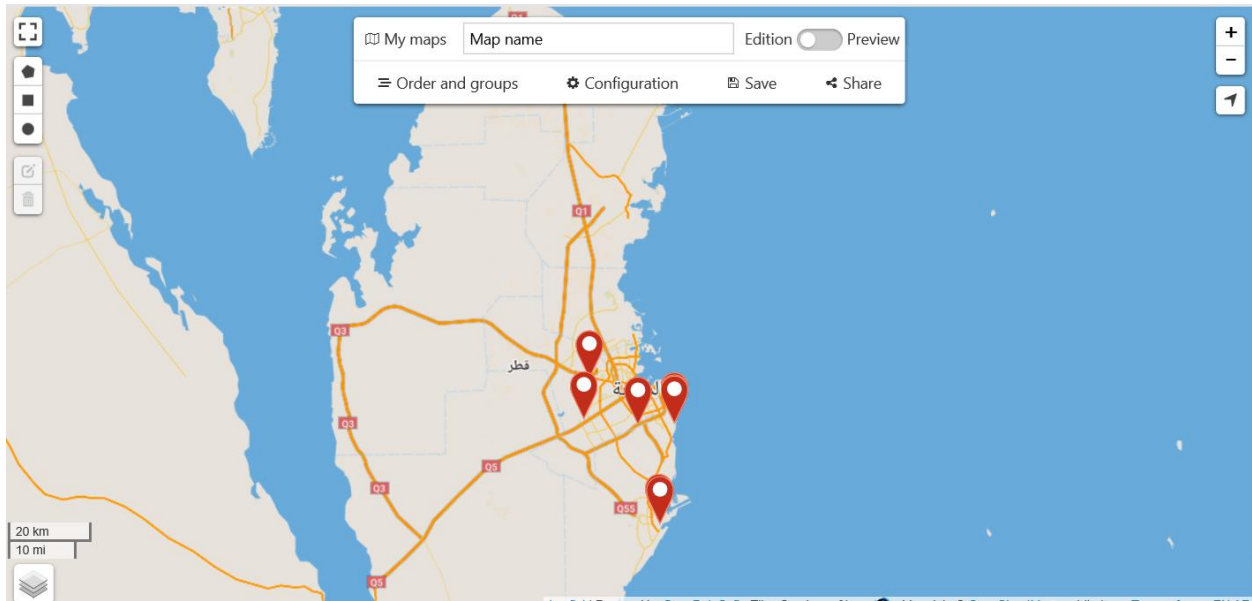
- polygon ⬠, circle ● and square ■ buttons to draw on the map and filter on certain areas, along with the edit shape ☑ and delete shape 🗑 buttons to go with them
- ⛶ to view your map in fullscreen mode

- zoom in ✛ and zoom out ━ buttons
- localize ➤ button to focus the map on your current location

However, any interactive map created with Map Builder also contains specific features, whether they are optional or not:

- a search bar, to make research on your map
- the list of all the layers contained in the map, which you can hide or display
- on the bottom right-hand corner of the map, a box containing the caption of each layer displayed, to better understand the map

Display or hide layers

By default, all your map layers are displayed on your map. The list of all the layers that compose your Map Builder map is displayed on the top right-hand corner of the map.

You can hide layers in case you only want to focus on specific layers or even on just one very layer. To hide a layer, simply click on it. You will see that the layer disappears from the map, and that its title and description in the list turn light gray and are detached from the active, displayed layers.

Understand the map: captions

For a better understanding of the map, each layer can have a related caption you can refer to. All available captions are displayed on the bottom right-hand corner of the map, under the list of layers. To go from one caption to another, simply click the ‹ and › buttons.
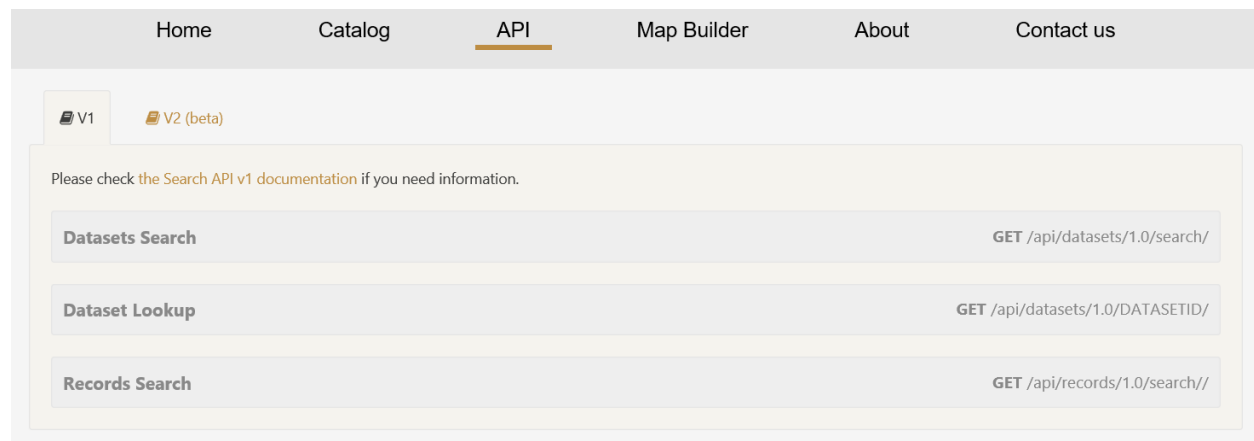
By default, the caption box is reduced to showing the most important information only, and to not take too much space on the map. You can only see the name of the layer it refers to and its captions. To expand this box and see more details, simply click the ▲ button.

Use the search bar

You can use the search bar to make research on your map. To do so, simply enter a keyword in the search bar at the top left-hand corner of the map. By default, the search bar will return location results. If you click on a location result, the map will automatically move to that selected location.

API

Search API v1

Available APIs

API Name	Description
Dataset search API	Search datasets in a catalog
Dataset lookup API	Find a dataset based on its identifier
Records search API	Search records within a given dataset
Records lookup API	Find a record based on its identifier

Analysis API	Build advanced aggregations of records from a given dataset
Download API	Efficiently download a large number of records from a given dataset
Geo clustering API	Build geo clusters of records from a given dataset
Real time push API	Push new records to a given dataset in real time
Multimedia download API	Download multimedia content attached to a dataset or a record

These APIs return JSON by default, except:

- the download API that returns CSV by default but supports several output formats like JSON and geographic formats
- the multimedia download API that depends on the file

The real time push API is very specific and will not be detailed in the following documentation. Please refer to the platform documentation for more information about this API.

These APIs support cross-domain access from a browser using CORS. For older browsers or other specific purposes, JSONP is supported when returning JSON content by adding a `callback` parameter.

Finding a dataset identifier

To access a dataset directly via the dataset lookup API or record related APIs, its **identifier** must be provided. This identifier is found in the information tab of each dataset.

HTTP Methods

All API endpoints support both GET and POST requests. GET methods are preferred, but POST methods can be used to work around browser, library or proxy limitations regarding the size of HTTP URLs.

Security

All API endpoints are protected by the same authentication and authorization model.

Anonymous access and authenticated users can be restricted to:

- a subset of the domain's datasets
- a subset of records in a given dataset

All API endpoints are available in HTTPS, which use is highly recommended wherever possible.

The following authentication modes are available:

- **HTTP Basic authentication:** via an account login and password (https://en.wikipedia.org/wiki/Basic_access_authentication)
- **API key authentication:** via an API key generated from the account settings page
- **Session authentication:** API calls performed from a browser will authenticate logged users via the OpenDataSoft session cookie

Quotas

All API endpoints are subject to quota-based limitations. According to the domain configuration, authenticated users may have extended quotas compared to anonymous access. Please contact the domain administrator for more information about a user's quotas.

The API response contains three headers to indicate the current state of a user's quota:

- **X-RateLimit-Limit** indicates the total number of API calls the user can do in a single day (resets at midnight UTC)
- **X-RateLimit-Remaining** indicates the remaining number of API calls for the user until reset
- **X-RateLimit-Reset** indicates the epoch of the next reset time

Errors handling

When an error occurs, a JSON object describing the error is returned by the API.

Authentication

An authenticated user can be granted access to restricted datasets and benefit from extended quotas for API calls. The API features an authentication mechanism for users to be granted their specific authorizations.

For the platform to authenticate a user, you need to either:

- be logged in a portal so a session cookie authenticating your user is passed along your API calls
- provide your username and password as HTTP Basic authentication tokens
- provide an **API key** as a request parameter

Finding and generating API keys

API keys are managed via your user profile page at `https://qatar.opendatasoft.com/account/` or by clicking on your name in the header.
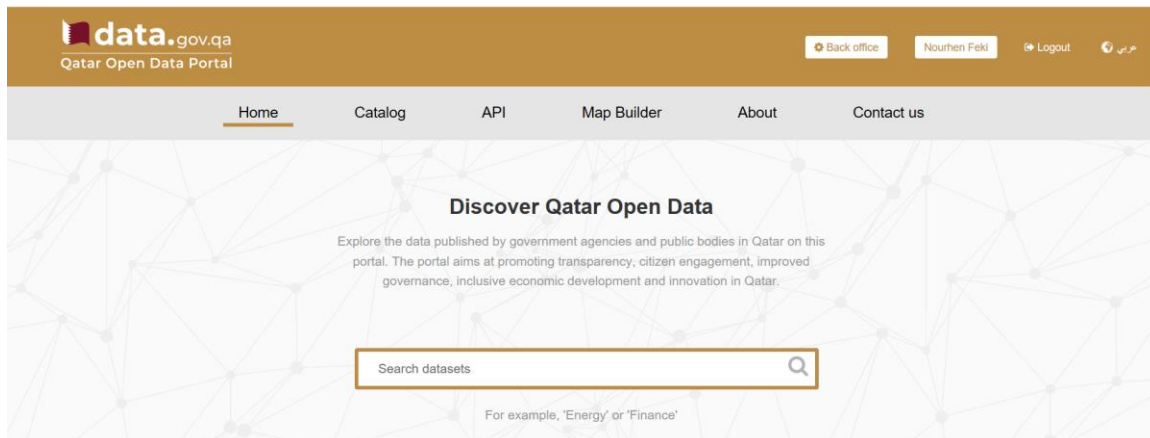
Go to the tab named **My API keys** to see your existing API keys, revoke them and create new ones.
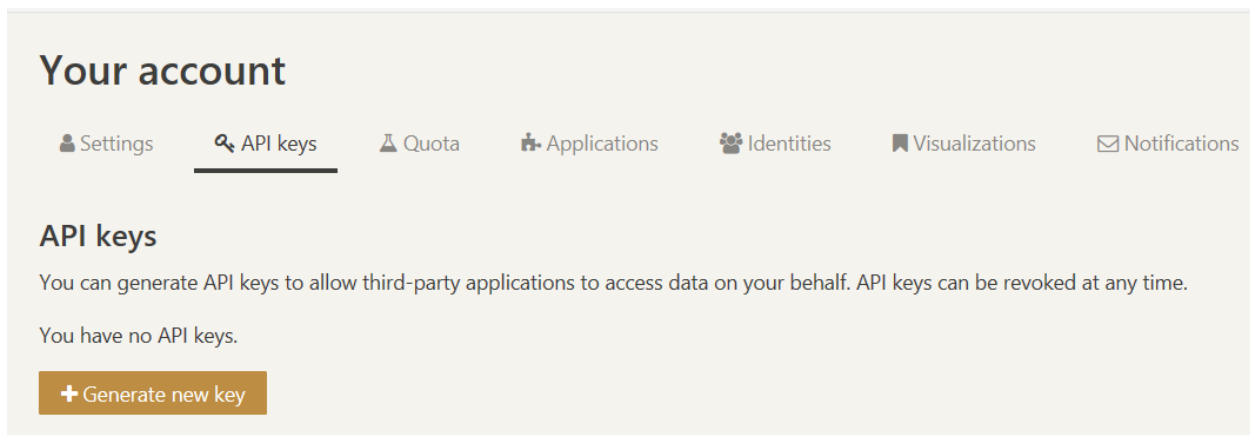
Every API key authenticates requests as coming from your user, which means they grant the same rights (yours) to any person using them. Therefore, you should not share your keys.

Providing API keys within requests

API keys are passed along requests through the query parameter `apikey`.

For example, accessing a private portal's catalog unauthenticated will return a `401 Unauthorized` error.

But passing the API key of an authorized user will return the JSON response with the list of accessible datasets for this user on the portal.

Using OAuth2 authorization

Overview

Qatar Open Data implements the OAuth2 authorization flow, allowing third party application makers to access the data hosted on an OpenDataSoft platform on behalf of a user while never having to deal with a password, thus avoiding any user credential to be compromised.

The OAuth2 authorization flow is compliant with RFC 6749 and makes use of Bearer Tokens in compliance with RFC 6750.

Application developers who want to use the APIs with OAuth2 must go through the following steps, which will be explained in this section.

1. Register their application with the platform.
2. Request approval from users via an OAuth2 authorization grant.
3. Request a bearer token that will allows them to query the platform APIs for a limited amount of time.
4. Refresh the Bearer Token when it expires.

Currently, applications are registered on a specific domain and can only access data on this domain.

Register an application for OAuth2 authentication

1. Go to the **Applications** tab of your account page on the domain you want to register the application on.
2. Fill the registration form with the following information:
    o   **Application name:** the name of the application
    o   **Type:**
        ▪   confidential: client password is kept secret from the user and only used from a trusted environment (e.g: a web service, where the client password is stored server-side and never sent to the user)
        ▪   public: client password is embedded in a client-side application, making it potentially available to the world (e.g: a mobile or desktop application)
    o   **Redirection URL:** the URL users will be redirected to after they have granted you permission to access their data
3. Store the resulting **client ID** and **client secret** that will be needed to perform the next steps.

Getting an authorization grant

To get an authorization grant from a user:

1. Redirect them to `/oauth2/authorize/` with the appropriate query parameters.
2. The user will then be authenticated in the platform and redirected to a page identifying your application.
3. From there, the user will review the information you filled in the form described above and the scope of the requested access, and grant your application the right to access their data.
4. Once the user has accepted those terms, they will be redirected to your application's redirection URL with query parameters describing your authorization grant.

The query parameters you need to supply when redirecting the user are the following:

*   `client_id`: the client ID you were given during registration
*   `redirect_uri`: the redirect URI you provided during registration
*   `response_type`: this should always be set to `code`
*   `scopes` *(optional)*: a list of space-separated requested scopes. Currently only `all` is supported
*   `state` *(optional)*: a random string of your choice

The state parameter is not mandatory, but providing one is recommended for security reasons to verify the returned value provided in the authorization grant redirect

The authorization grant redirect will have these values:

*   `code`: a 30-characters-long authorization code
*   `state`: the state passed in the request described above

The 30-character authorization code must now be converted into a bearer token within 1 hour before expiring.

Converting an authorization grant to a bearer token

To receive a bearer token, convert the previously obtained authorization grant via a POST request to `/oauth2/token/` with the following parameters:

- `client_id`: the client ID you were given during registration
- `client_secret`: the client secret you were given during registration
- `redirect_uri`: the redirect URI you provided during registration
- `grant_type`: this should always be set to `authorization_code`
- `code`: the 30-character authorization code received as an authorization grant
- `scopes` *(optional)*: a list of space-separated requested scopes. Currently only `all` is supported
- `state` *(optional)*: a random string of your choice

Alternatively, you can pass your client ID and client secret through the Authorization header

The response to this request is a JSON representation of a bearer token, which contains the following values:

- `access_token`: the token you can use to access the user's data.
- `expires_in`: the number of seconds before token expiration
- `token_type`: the type of the token. It will always be `Bearer`
- `state`: the state passed in the request described above
- `scope`: the list of scopes of this authorization code
- `refresh_token`: a refresh token that can be used to renew this bearer token when expired

Unlike the access token, that can be used any number of times until expiration, the refresh token doesn't expire but can only be used once

Using the bearer token

The bearer token can be passed along requests for authentication in three different ways:

- as a query parameter of the request
- in the request's `Authorization` header
- in the request body

Refreshing a bearer token

To refresh an expired bearer token, send a request to the `/oauth2/token/` endpoint, with the following query parameters:

- `client_id`: the client ID you were given during registration
- `client_secret`: the client secret you were given during registration
- `refresh_token`: the refresh token returned in the bearer token response
- `grant_type`: this should always be set to `refresh_token`
- `scopes`: a list of space-separated requested scopes. Currently only `all` is supported
- `state` *(optional)*: a random string of your choice

The response to this request is identical to the bearer token response.

Query Language and Geo Filtering

Filtering features are built in the core of Qatar Open Data Portal API engine. Many of the previously listed APIs can take as a parameter filters for constraining the list of returned datasets or records.

Note that a given filtering context can simply be copied from one API to another. For example, you can easily build a user interface which first allows the user to visually select the records their are interested in, using full text search, facets and geo filtering, and then allowing them to download these records with the same filtering context.

Query language

The query language makes it possible to express complex boolean conditions as a filtering context.

The user query can most of the time be expressed through the `q` HTTP parameter.

Full-text search

The query language accepts full text queries.

If a given word or compounds is surrounded with double quotes, only exact matches are returned (modulo an accent and case insensitive match).

Boolean expressions

The query language supports the following boolean operators `AND`, `OR` and `NOT`.

Parenthesis can be used to group together expressions and alter the default priority model:

- `NOT`
- `AND`
- `OR`

Field queries

One of the major feature of the query language is to allow per field filtering. You can use field names as a prefix to your queries to filter the results based on a specific field's value.

For the dataset search API, the list of available fields corresponds exactly to available metadata. By default

Field name	Description
Publisher	The dataset publisher
Title	The dataset title
Description	The dataset description
license	The dataset license
records_count	The number of records in the dataset
modified	The last modification date of the dataset
language	The language of the dataset (iso code)
theme	The theme of the dataset
references	The references for the dataset

The domain administrator might define a richer metadata template, thus giving access to a richer set of filtering fields.

For the record search API, the list of available fields depends on the schema of the dataset. To fetch the list of available fields for a given dataset, you may use the search dataset or lookup dataset APIs.

Multiple operator fields can be used between the field name and the query:

- :, -, ==: return results whose field exactly matches the given value (granted the fields are of text or numeric type)
- >, <, >=, <=: return results whose field values are larger, smaller, larger or equal, smaller or equal to the given value (granted the field is of date or numeric type)
- [start_date TO end_date]: query records whose date is between start_date and end_date

Date formats can be specified in different formats: simple (YYYY[[/mm]/dd]) or ISO 8601 (YYYY-mm-DDTHH:MM:SS)

Query language functions

Advanced functions can be used in the query language. Function names need to be prefixed with a sharp (#) sign.

Function name	Description
Now	Return the current date. This function should be

	called as a query value for a field
Null	Search for records where no value is defined for the given field
Exact	Search for records with a field exactly matching a given value

Available parameters for the #now function

- years, months, weeks, days, hours, minutes, seconds, microseconds: these parameters add time to the current date.
- year, month, day, hour, minute, second, microsecond: can also be used to specify an absolute date.
- weekday: Specifies a day of the week. This parameter accepts either an integer between 0 and 6 (where 0 is Monday and 6 is Sunday) or the first two letters of the day (in English) followed by the cardinal of the first week on which to start the query.

Geo Filtering

Records search APIs accept geofilter parameters to filter in records which are located in a specific geographical area.

The following parameters may be used.

Parameter Name	Description
geofilter.distance	Limit the result set to a geographical area defined by a circle center (WGS84) and radius (in meters): latitude, longitude, distance
geofilter.polygon	Limit the result set to a geographical area defined by a polygon (points expressed in WGS84): ((lat1, lon1), (lat2, lon2), (lat3, lon3))

Using facets

A facet can be considered as a valued tag associated with a record. For instance, let's say a dataset has a facet "City". A record in this dataset could have the value "Paris" for the "City" facet.

Facets are for instance used for building the left navigation column, both for dataset catalog exploration page and dataset records exploration page.

Facets are especially useful to implement guided navigation in large result sets.

Identifying facets

By default, in dataset and record APIs, faceting is disabled. Faceting can be enabled by using the "facet" API parameter, specifying the name of the facet to retrieve.

In the dataset APIs, facets are the same for all datasets and are the following:

Facet Name	Description
modified	Last modification date
publisher	Producer
issued	First publication date
accrualperiodicity	Publication frequency
language	Language
license	Licence
granularity	Data granularity
dataquality	Data quality
theme	Theme
keyword	Keywords
created	Creation date
creator	Creator
contributor	Contributors

In the records API, facets are defined at field level. A field facet can be available depending on the data producer choices. Fields (retrieved for instance from the Dataset Lookup API) for which faceting is available can be easily identified as shown in the example on the right.

When faceting is enabled, facets are returned in the response after the result set.

Every facet has a display value ("name" attribute) and a refine property ("path" attribute) which can be used to alter the query context.

Facets are hierarchical, for instance, a year facet will contain months facets and a month facet will contain days facets.

Every facet contains two additional information:

- the `count` attribute contains the number of hits that have the same facet value
- the `state` attribute defines whether the facet is currently used in a `refine` or in an `exclude`. Possible values for the state attribute are:
  - **displayed**: no refine nor exclude

o **refined**: refine
o **excluded**: exclude

Refining

It is possible to limit the result set by refining on a given facet value. To do so, use the following API parameter: `refine.FACETNAME=FACETVALUE`.

For example: https://public.opendatasoft.com/api/datasets/1.0/search?refine.modified=2013.

In the returned result set, only the datasets modified in 2013 will be returned.

As the refinement occurs on the "year" and as the "modified" facet is hierarchical, the sub-level is returned. Results are dispatched in the "month" sub value.

Excluding

Using the same principle as above, it is possible to exclude from the result set the hits matching a given value of a given facet. To do so, use the following API parameter:
`exclude.FACETNAME=FACETVALUE`.

For example: https://public.opendatasoft.com/api/datasets/1.0/search?exclude.modified=2013

Only results that have not been modified in 2011 will be returned.

Disjunctive faceting

By default, faceting is conjunctive. This means that the following context will lead down to no results:
https://public.opendatasoft.com/api/datasets/1.0/search?refine.modified=2013&refine.modified=2014.

You can enable disjunctive faceting using the following API parameter:
`disjunctive.FACETNAME=true`.

For example:
https://public.opendatasoft.com/api/datasets/1.0/search?refine.modified=2013&refine.modified=2014&disjunctive.modified=true

Dataset Catalog APIs

Dataset Search API

This API provides a search facility in the dataset catalog. Full text search as well as multi-criteria field queries are made possible and results facetting is provided as well.

Parameters

Parameter	Description
`q`	Full-text query performed on the result set
`Facet`	Activate faceting on the specified field (see list of fields in the [Query Language](#) section). This parameter can be used multiple times to activate several facets. By default, faceting is disabled
`refine.<FACET>`	Limit the result set to records where `FACET` has the specified value. It can be used several times for the same facet or for different facets
`exclude.<FACET>`	Exclude records where `FACET` has the specified value from the result set. It can be used several times for the same facet or for different facets
`sort`	Sorts results by the specified field. By default, the sort is descending. A minus sign – may be used to perform an ascending sort. Sorting is only available on numeric fields (int, double, date and datetime) and on text fields which have the `sortable` annotation
`Rows`	Number of results to return in a single call. By default, `10` results are returned. While you can request for up to `10 000` results in a row, such requests are not optimal and can be throttled so you should consider splitting them into smaller ones.
`Start`	Index of the first result to return (starting at 0). Use in conjunction with `rows to implement paging

Dataset Lookup API

This API makes it possible to fetch an individual dataset information.

Parameters

The dataset identifier is passed as a part of the URL as indicated by the `<dataset_id>` placeholder in the example on the right.