depositar Documentation

Release 6.3.2

The depositar Team

Oct 25, 2018

Contents

CHAPTER 1

User guide

"depositar" is a public platform for storing, preserving, managing, and exploring research data. depositar is built with CKAN, which is an open source data management system, and extended with many useful features. The site is located at https://data.depositar.io.

This user guide covers using CKAN's web interface to organize, publish and find data. CKAN also has a powerful API (machine interface), which makes it easy to develop extensions and links with other information systems. The API is documented in http://docs.ckan.org/en/2.7/api/index.html.

Some web UI features relating to site administration are available only to users with sysadmin status, and are documented in http://docs.ckan.org/en/2.7/sysadmin-guide.html.

1.1 What is CKAN?

CKAN is a tool for making open data websites. (Think of a content management system like WordPress - but for data, instead of pages and blog posts.) It helps you manage and publish collections of data. It is used by national and local governments, research institutions, and other organizations who collect a lot of data. depositar is built with CKAN.

Once your data is published, users can use its faceted search features to browse and find the data they need, and preview it using maps, graphs and tables - whether they are developers, journalists, researchers, NGOs, citizens, or even your own staff.

1.1.1 Datasets and resources

For CKAN purposes, data is published in units called "datasets". A dataset is a parcel of data - for example, it could be the crime statistics for a region, the spending figures for a government department, or temperature readings from various weather stations. When users search for data, the search results they see will be individual datasets.

A dataset contains two things:

- Information or "metadata" about the data. For example, the title and publisher, date, what formats it is available in, what license it is released under, etc.
- A number of "resources", which hold the data itself. CKAN does not mind what format the data is in. A resource can be a CSV or Excel spreadsheet, XML file, PDF document, image file, linked data in RDF format, etc. CKAN can store the resource internally, or store it simply as a link, the resource itself being elsewhere on the web. A dataset can contain any number of resources. For example, different resources might contain the data for different years, or they might contain the same data in different formats.

1.2 Using `depositar`

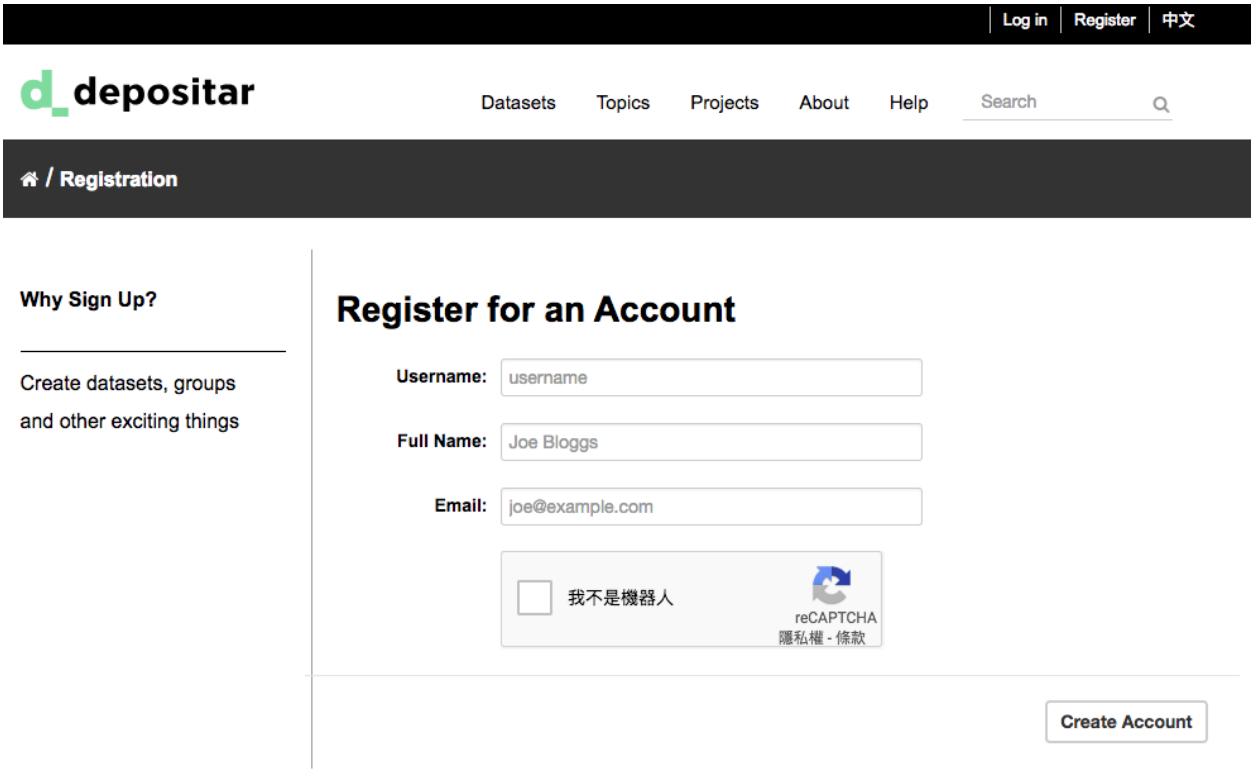
1.2.1 Registering and logging in

Note: Registration is needed for most publishing features and for personalization features, such as "following" datasets. It is not needed to search for and download data.

Hint: We provide a demo system at https://demo.depositar.io with the same features as `depositar` for evaluation purposes. You can create an account and try any functions provided by `depositar`. Please note that all data in this instance will be deleted occasionally.

To create a user ID, use the "Register" link at the top of any page. CKAN will ask for the following:

- *Username* – choose a username using only letters, numbers, - and _ characters. For example, "jbloggs" or "joe_bloggs93".
- *Full name* – to be displayed on your user profile
- *E-mail address* – this will not be visible to other users



If there are problems with any of the fields, CKAN will tell you the problem and enable you to correct it. When the fields are filled in correctly, we will receive an email to set your password as follows. Then you can use the "Log in" link at the top of any page to log in.

```
Dear OOO,

You have requested your password on depositar to be reset.

Please click the following link to confirm this request:

   https://data.depositar.io/user/reset/[token]

Have a nice day.

--
Message sent by depositar (https://data.depositar.io)
```

1.2.2 Features for publishers

Adding a new dataset

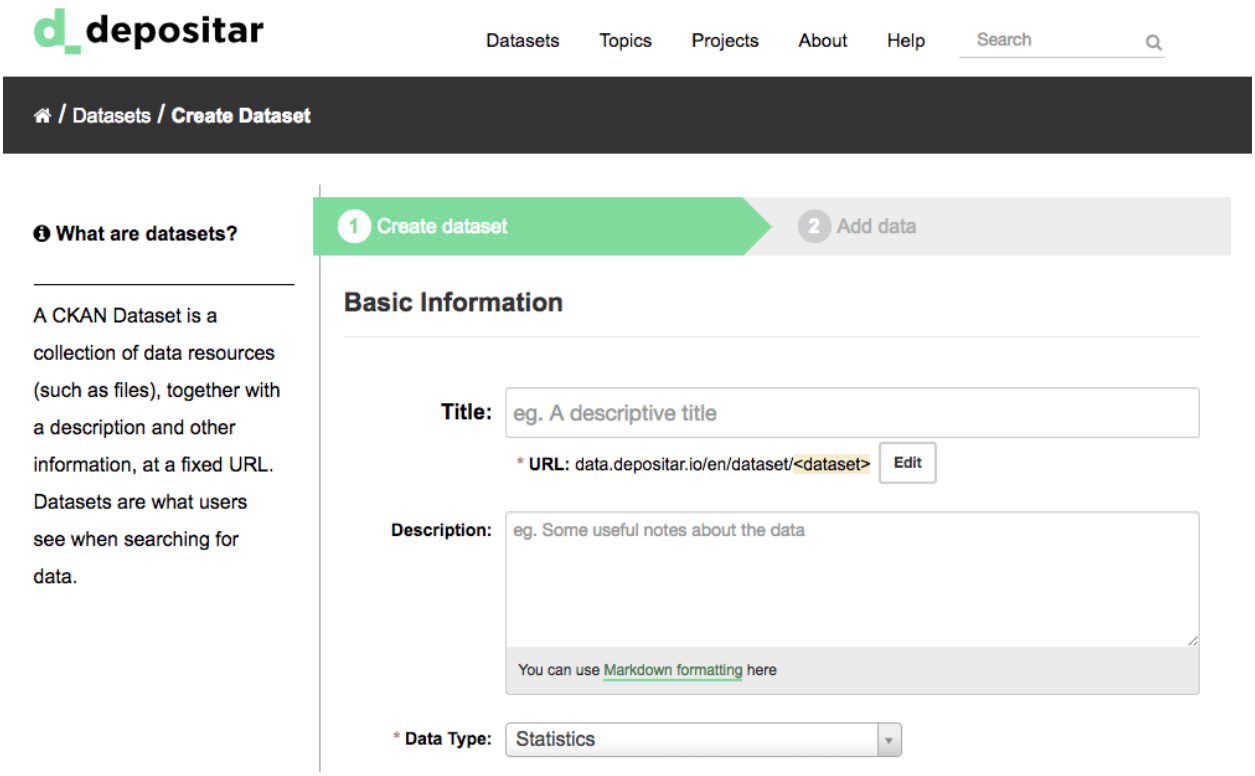
Note: You may need an user account in order to add and edit datasets.

Step 1. You can access CKAN's "Create dataset" screen in two ways.

1. Select the "Datasets" link at the top of any page. From this, above the search box, select the "Add Dataset" button.

2. Alternatively, select the "projects" link at the top of a page. Now select the page for the project that should own your new dataset. Provided that you are a member of this project, you can now select the "Add Dataset" button above the search box.

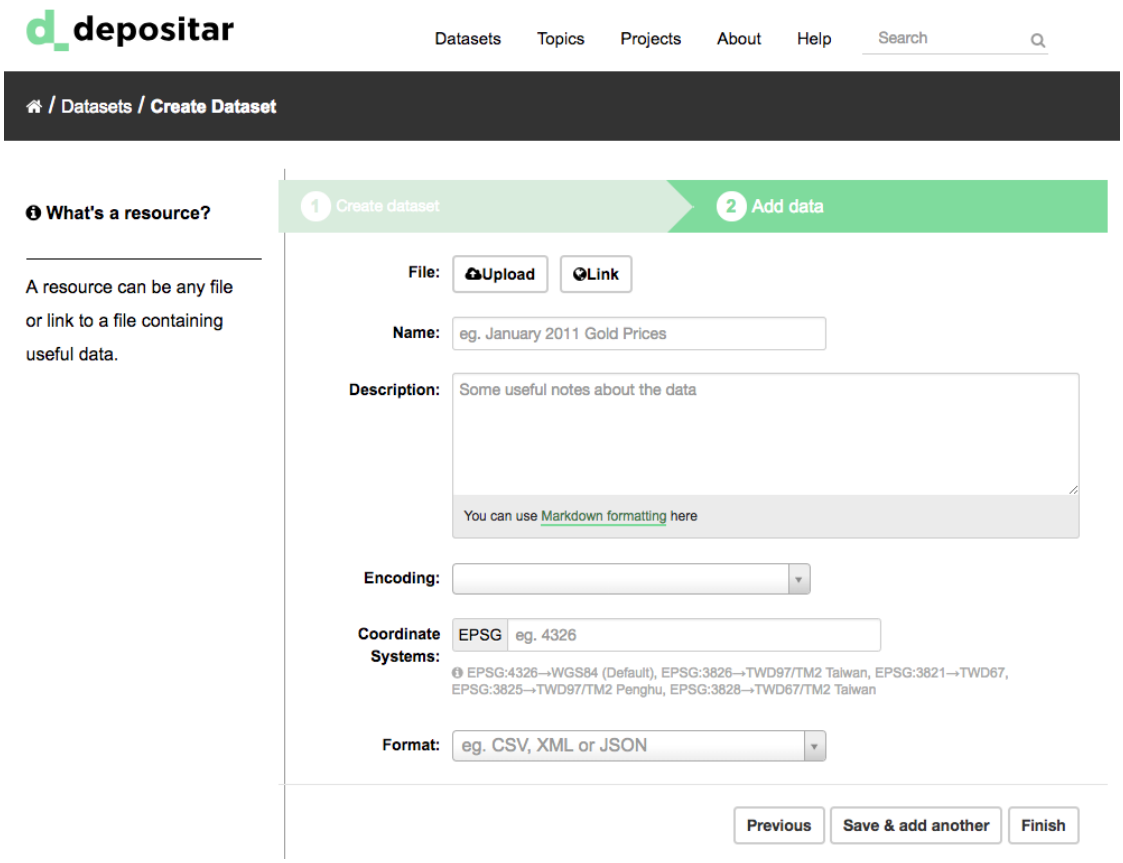
Step 2. CKAN will ask for the information about your data (See *Metadata at the dataset level*).



Note: By default, the only required field on this page is the title. However, it is good practice to include, at the minimum, a short description and, if possible, the license information. You should ensure that you choose the correct project for the dataset, since at present, this cannot be changed later. You can edit or add to the other fields later.

Step 3. When you have filled in the information on this page, select the "Next: Add Data" button. (Alternatively select "Cancel" to discard the information filled in.)

Step 4. CKAN will display the "Add data" screen.



This is where you will add one or more "resources" which contain the data for this dataset. Choose a file or link for your data resource and select the appropriate choice at the top of the screen:

- If you are giving CKAN a link to the data, like `http://example.com/mydata.csv`, then select "Link to a file" or "Link to an API". (If you don't know what an API is, you don't need to worry about this option - select "Link to a file".)
- If the data to be added to CKAN is in a file on your computer, select "Upload a file". CKAN will give you a file browser to select it.

Step 5. Add the other information on the page. (Please refer to *Metadata at the resource level*) CKAN does not require this information, but it is good practice to add it.

Step 6. If you have more resources (files or links) to add to the dataset, select the "Save & add another" button. When you have finished adding resources, select "Next: Additional Info".

Step 7. Select the 'Finish' button. CKAN creates the dataset and shows you the result. You have finished!

You should be able to find your dataset by typing the title, or some relevant words from the description, into the search box on any page in your CKAN instance. For more information about finding data, see the section *Finding data*.

Extended feature — Add a dataset to an existing topic

The topic is different from "Projects" feature as the latter is the way to control the visibility of datasets in CKAN and each dataset can belong to ONLY ONE project.

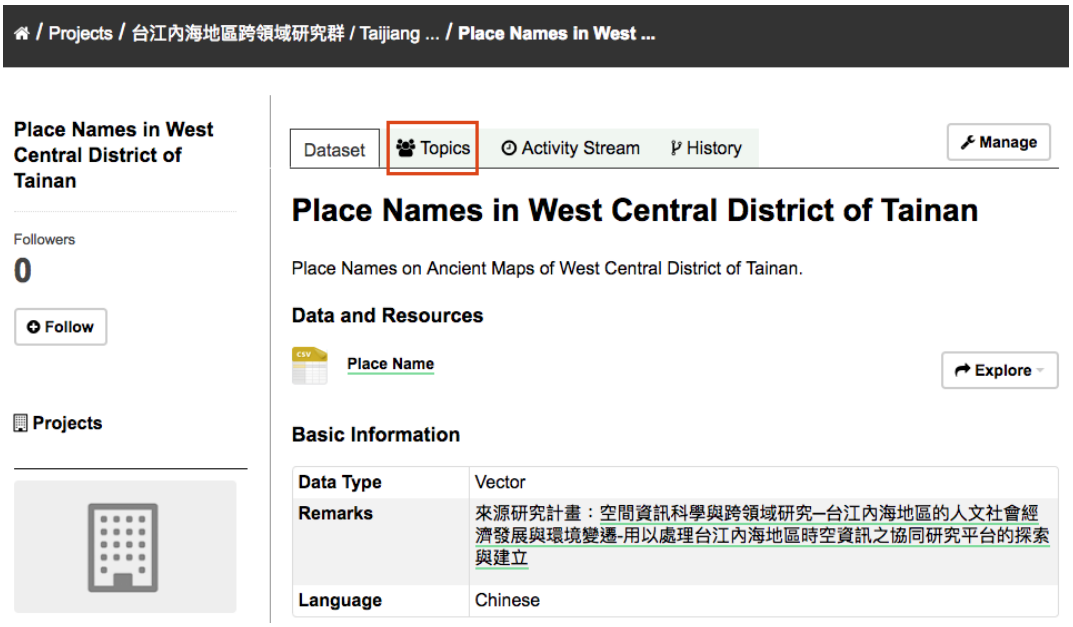
We refer the ISO19115 standard to define the following themes:

- **farming**: Rearing of animals or cultivation of plants, for example agriculture, irrigation, aquaculture, plantations, herding, pests and diseases affecting crops and livestock
- **biota**: Flora or fauna in natural environment, for example wildlife, vegetation, biological sciences, ecology, wilderness, sea life, wetlands, habitat, biological resources
- **boundaries**: Legal land descriptions, for example political and administrative boundaries, governmental units, marine boundaries, voting districts, school districts, international boundaries
- **climatologyMeteorologyAtmosphere**: Processes and phenomena of the atmosphere, for example cloud cover, weather, climate, atmospheric conditions, climate change, precipitation
- **economy** Economic activities, conditions, and employment, for example production, labor, revenue, business, commerce, industry, tourism and ecotourism, forestry, fisheries, commercial or subsistence hunting, exploration and exploitation of resources such as minerals, oil and gas
- **elevation** Height above or below seal level, for example altitude, bathymetry, digital elevation models, slope, derived products, DEMs, TINs
- **environment** Environmental resources, protection and conservation, for example environmental pollution, waste storage and treatment, environmental impact assessment, monitoring environmental risk, nature reserves, landscape, water quality, air quality, environmental modeling
- **geoscientificInformation** Information pertaining to earth sciences, for example geophysical features and processes, geology, minerals, sciences dealing with the composition, structure and origin of the earth's rocks, risks of earthquakes, volcanic activity, landslides, gravity information, soils, permafrost, hydrogeology, groundwater, erosion
- **health** Health, health services, human ecology, and safety, for example disease and illness, factors affecting health, hygiene, substance abuse, mental and physical health, health services, health care providers, public health
- **imageryBaseMapsEarthCover** Base maps, for example land/earth cover, topographic maps, imagery, unclassified images, annotations, digital ortho imagery
- **intelligenceMilitary** Military bases, structures, activities, for example barracks, training grounds, military transportation, information collection
- **inlandWaters** Inland water features, drainage systems and characteristics, for example rivers and glaciers, salt lakes, water utilization plans, dams, currents, floods and flood hazards, water quality, hydrographic charts, watersheds, wetlands, hydrography
- **location** Positional information and services, for example addresses, geodetic networks, geodetic control points, postal zones and services, place names, geographic names
- **oceans** Features and characteristics of salt water bodies (excluding inland waters), for example tides, tidal waves, coastal information, reefs, maritime, outer continental shelf submerged lands, shoreline
- **planningCadastre** Information used for appropriate actions for future use of the land, for example land use maps, zoning maps, cadastral surveys, land ownership, parcels, easements, tax maps, federal land ownership status, public land conveyance records
- **society** Characteristics of society and culture, for example settlements, housing, anthropology, archaeology, education, traditional beliefs, manners and customs, demographic data, tourism, recreational areas and activities, parks, recreational trails, historical sites, cultural resources, social impact assessments, crime and justice, law enforcement, census information, immigration, ethnicity
- **structure** Man-made construction, for example buildings, museums, churches, factories, housing, monuments, shops, towers, building footprints, architectural and structural plans
- **transportation** Means and aids for conveying persons or goods, for example roads, airports/airstrips, shipping routes, tunnels nautical charts, vehicle or vessel location, aeronautical charts, railways

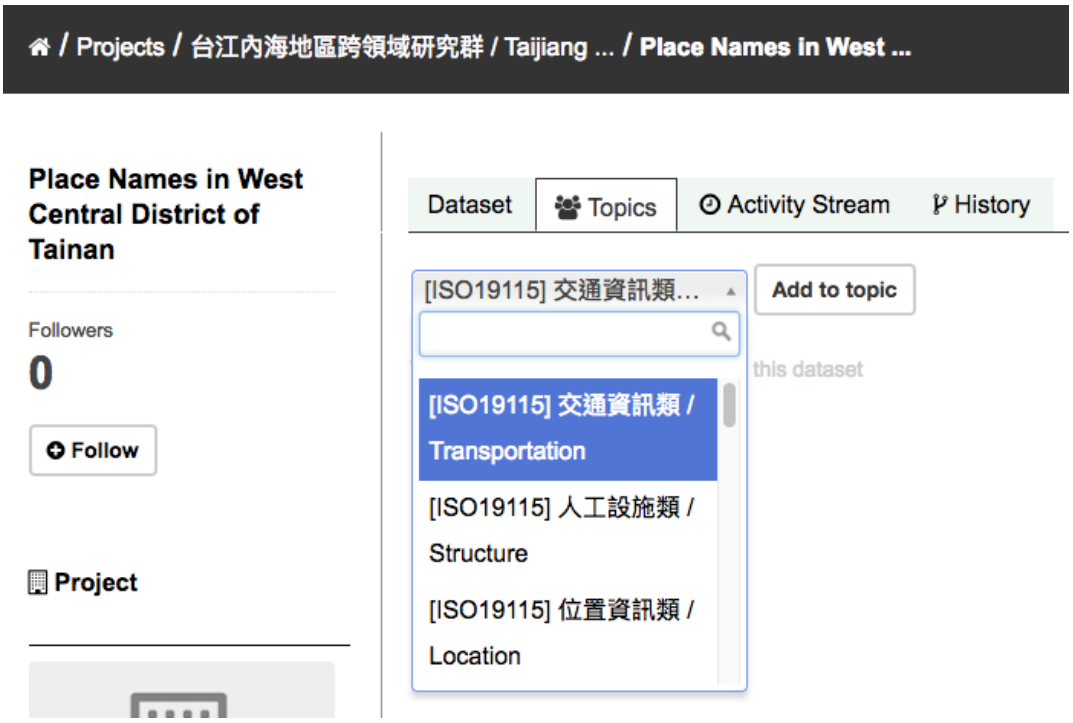
- **utilitiesCommunication** Energy, water and waste systems and communications infrastructure and services, for example hydroelectricity, geothermal, solar and nuclear sources of energy, water purification and distribution, sewage collection and disposal, electricity and gas distribution, data communication, telecommunication, radio, communication networks

Before adding a dataset to a theme, you should complete the upload process of the dataset (listed on the *Adding a new dataset*). Then do the following steps:

- Go to the dataset's page. You can find it by entering the title in the search box on any page.
- Select the "Topics" tab in the dataset's page.



- Select an existing topic and select the "Add to topic" button.

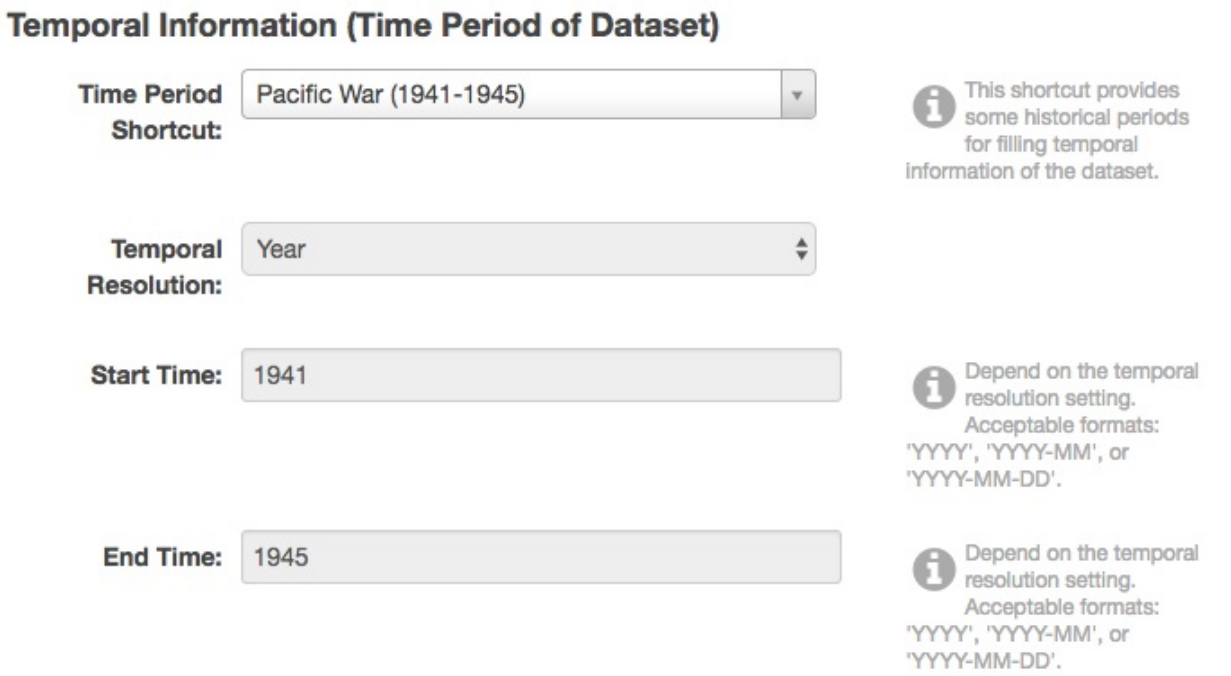


Extended feature — Fill-in snippet

- **Temporal Information**

The "temporal information" here means the time to events related to the dataset, not the time when the resources in the dataset were created.

- *Time Period Shortcut* – This shortcut provides some historical periods for filling temporal information of the dataset.
- *Temporal Resolution*[1] – This refers to the precision of a measurement with respect to time.
- *Start and End Time* – It depends on the temporal resolution setting. Acceptable formats: "YYYY", "YYYY-MM", or "YYYY-MM-DD".



- **Spatial Fields**

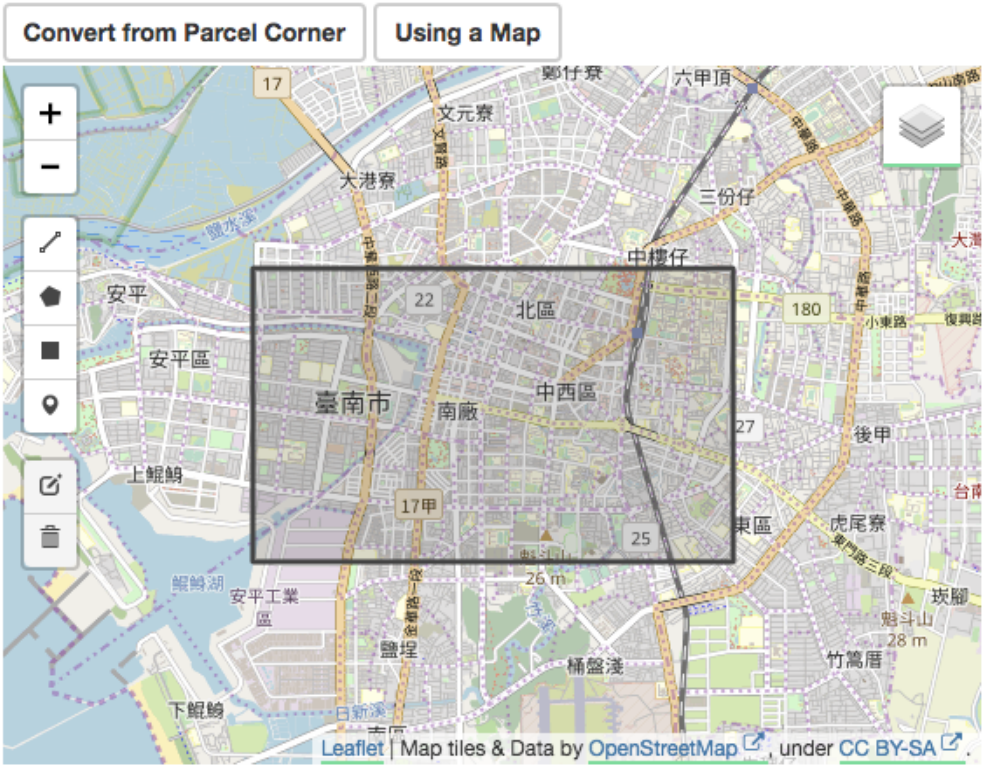
Here you can specify the spatial extent of the dataset for indexing, then the dataset can be found through *spatial search*.

You can use the following two methods to generate a valid spatial extent in GeoJSON format:

- *Convert from Parcel Corner* – If you already have the longitude and latitude of the corners for the parcel to describe the dataset, you can fill in the X.min, X.max, Y.mim, and Y.max fields, then select the "Convert from Parcel Corner" button to generate the spatial extent.
- *Using a Map* – You can also add the spatial extent through digitizing process. Select the "Using a Map" button and draw a polyline, polygon, rectangle, or marker on the expanded map to generate the spatial extent.

[1] This section uses material from the Wikipedia article Temporal resolution, which is released under the Creative Commons Attribution-Share-Alike License 3.0.

Spatial Fields



Spatial: {"type":"Polygon","coordinates":[[[120.1750659942627,22.97656943289965],
[120.1750659942627,23.003117971541474],
[120.22227287292482,23.003117971541474],
[120.22227287292482,22.97656943289965],
[120.1750659942627,22.97656943289965]]]}

This column is used for geo-indexing the dataset. Check http://geojson.org/ for details.

X.min: 120.1750659942627

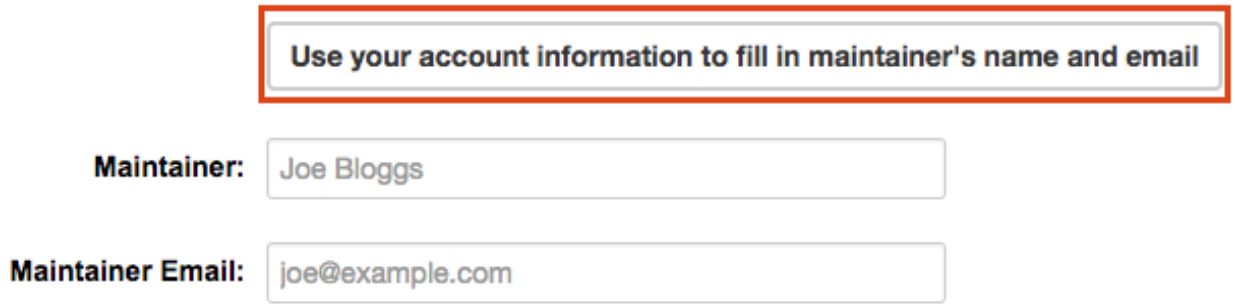
X.max: 120.22227287292482

Y.min: 22.97656943289965

Y.max: 23.003117971541474

- **Auto-completion of management metadata**

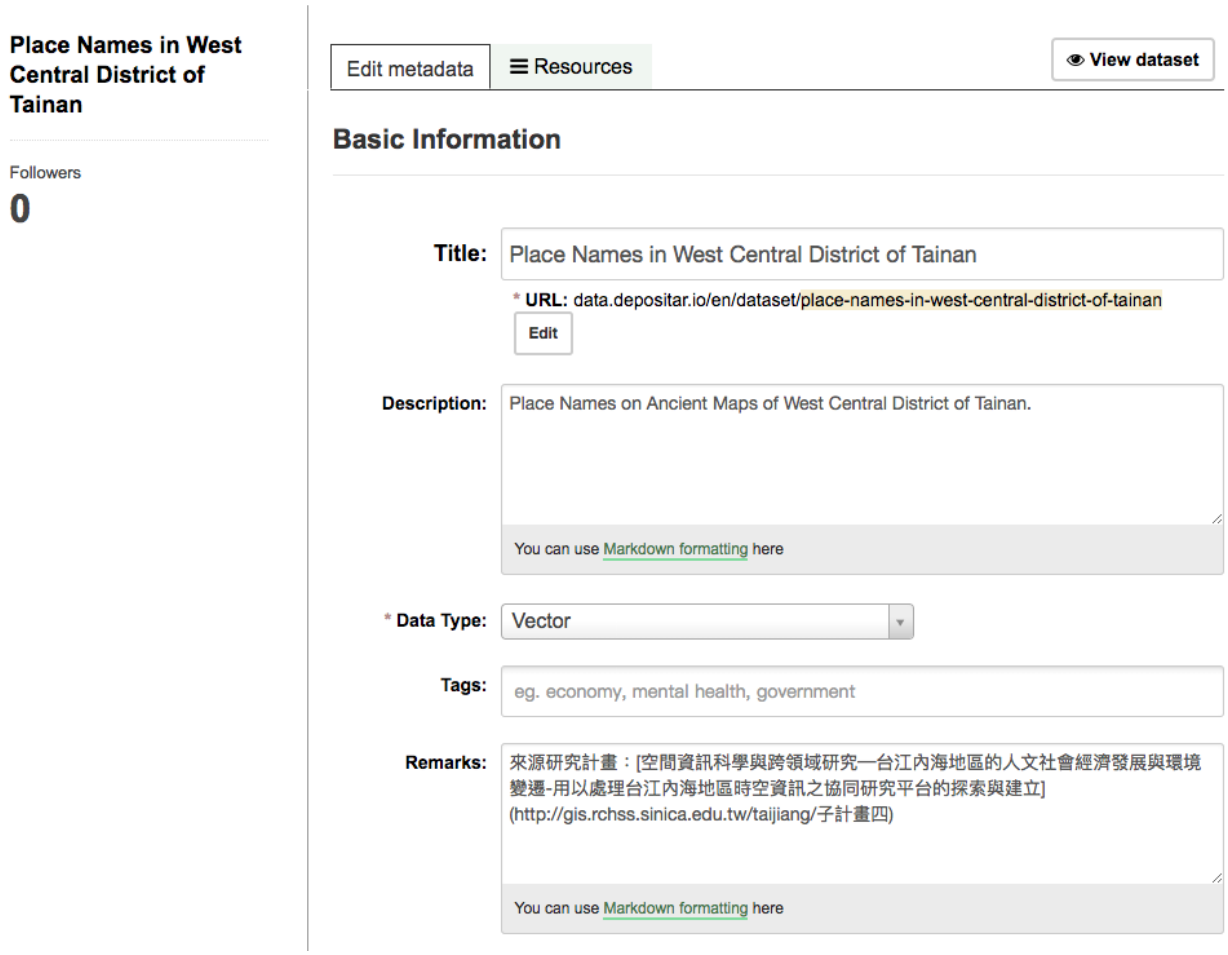
You can use the "Use your account information to fill in maintainer's name and email" button to automatically fill in the maintainer's information (`Maintainer` and `Maintainer Email`) using your account information (for account information, please refer to *Managing your user profile*).



Editing a dataset

You can edit the dataset you have created, or any dataset owned by an project that you are a member of. (If a dataset is not owned by any project, then any registered user can edit it.)

1. Go to the dataset's page. You can find it by entering the title in the search box on any page.
2. Select the "Edit" button, which you should see above the dataset title.
3. CKAN displays the "Edit dataset" screen. You can edit any of the fields (Title, Description, Dataset, etc), change the visibility (Private/Public), and add or delete tags or custom fields. For details of these fields, see *Adding a new dataset*.
4. When you have finished, select the "Update dataset" button to save your changes.



Adding, deleting and editing resources

1. Go to the dataset's "Edit dataset" page (steps 1-2 above).
2. In the left sidebar, there are options for editing resources. You can select an existing resource (to edit or delete it), or select "Add new resource".
3. You can edit the information about the resource or change the linked or uploaded file. For details, see steps 4-5 of "Adding a new resource", above.
4. When you have finished editing, select the button marked "Update resource" (or "Add", for a new resource) to save your changes. Alternatively, to delete the resource, select the "Delete resource" button.

Deleting a dataset

1. Go to the dataset's "Edit dataset" page (see "Editing a dataset", above).
2. Select the "Delete" button.
3. CKAN displays a confirmation dialog box. To complete deletion of the dataset, select "Confirm".

Note: The "Deleted" dataset is not completely deleted. It is hidden, so it does not show up in any searches, etc. However, by visiting the URL for the dataset's page, it can still be seen (by users with appropriate authorization), and

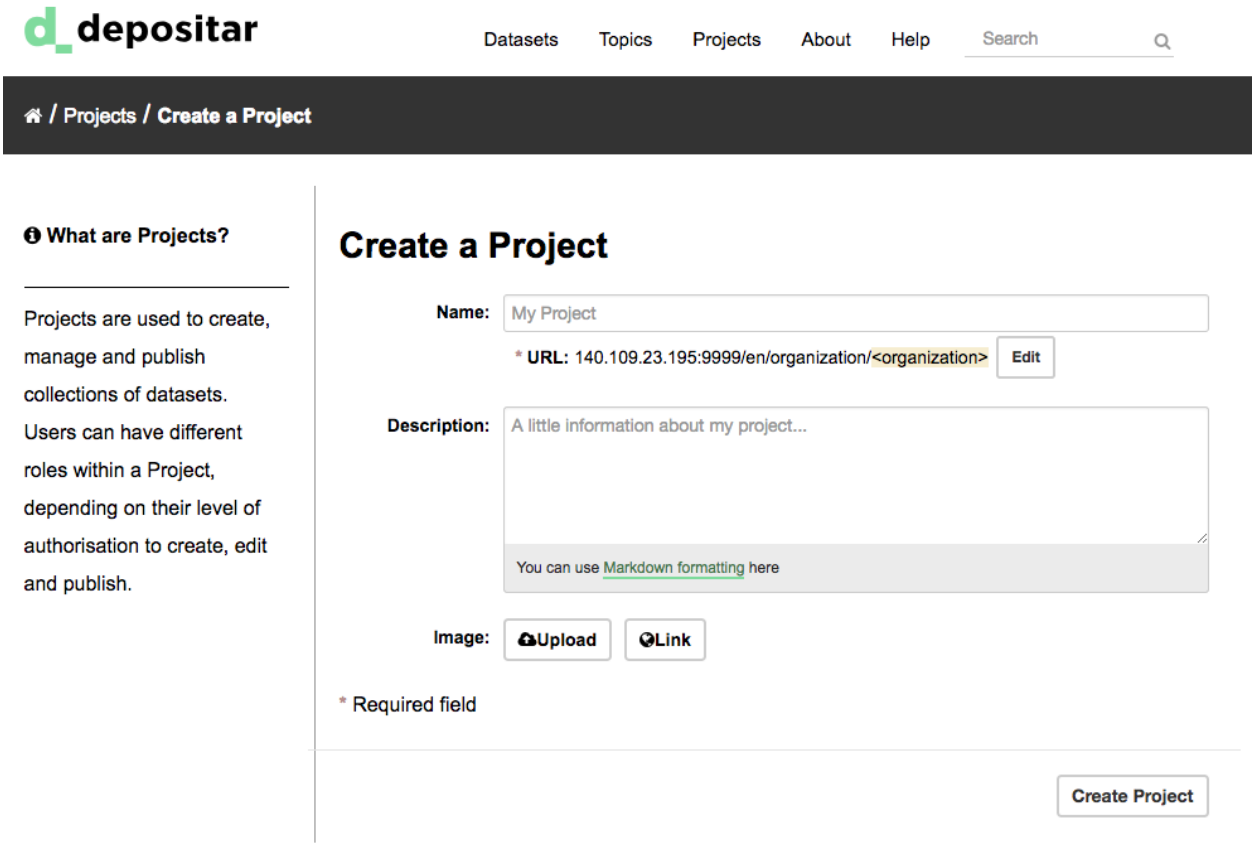
"undeleted" if necessary. If it is important to completely delete the dataset, contact your site administrator.

Creating a project

In general, each dataset is owned by one project. Each project includes certain users, who can modify its datasets and create new ones. Different levels of access privileges within a project can be given to users, e.g. some users might be able to edit datasets but not create new ones, or to create datasets but not publish them. Each project has a home page, where users can find some information about the project and search within its datasets. This allows different data publishing departments, bodies, etc to control their own publishing policies.

To create a project:

1. Select the "Projects" link at the top of any page.
2. Select the "Add Project" button below the search box.
3. CKAN displays the "Create a Project" page.
4. Enter a name for the project, and, optionally, a description and image URL for the project's home page.
5. Select the "Create Project" button. CKAN creates your project and displays its home page. Initially, of course, the project has no datasets.



You can now change the access privileges to the project for other users - see *Managing a project* below. You can also create datasets owned by the project; see *Adding a new dataset* above.

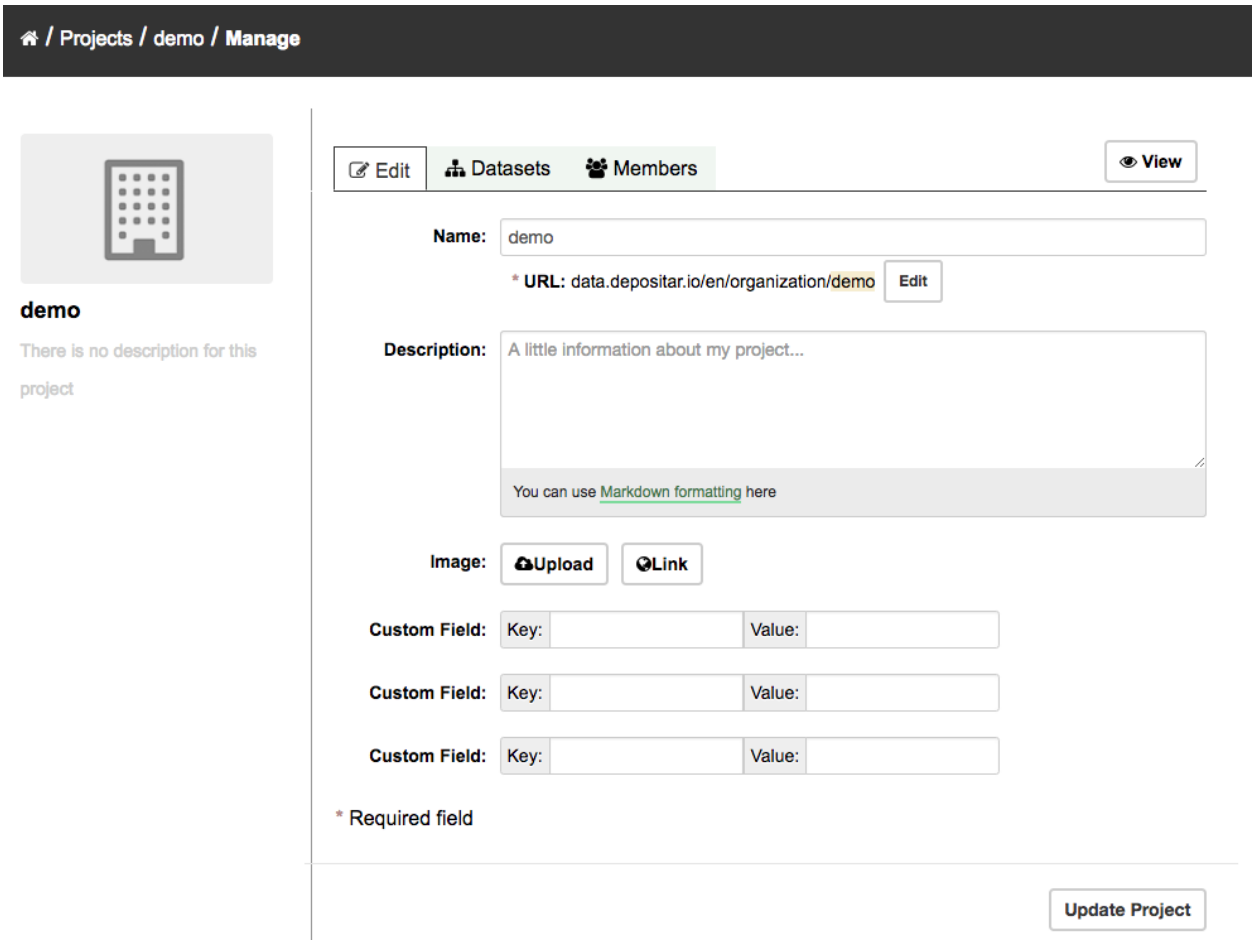
Note: You can learn how to fill in the information above by referring to existing projects. And, depending on how

CKAN is set up, you may not be authorized to create new projects. In this case, if you need a new project, you will need to contact your site administrator.

Managing a project

When you create a project, CKAN automatically makes you its "Admin". From the project's page you should see an "Admin" button above the search box. When you select this, CKAN displays the project admin page. This page has two tabs:

- *Info* – Here you can edit the information supplied when the project was created (title, description and image).
- *Members* – Here you can add, remove and change access roles for different users in the project. Note: you will need to know their username on CKAN.



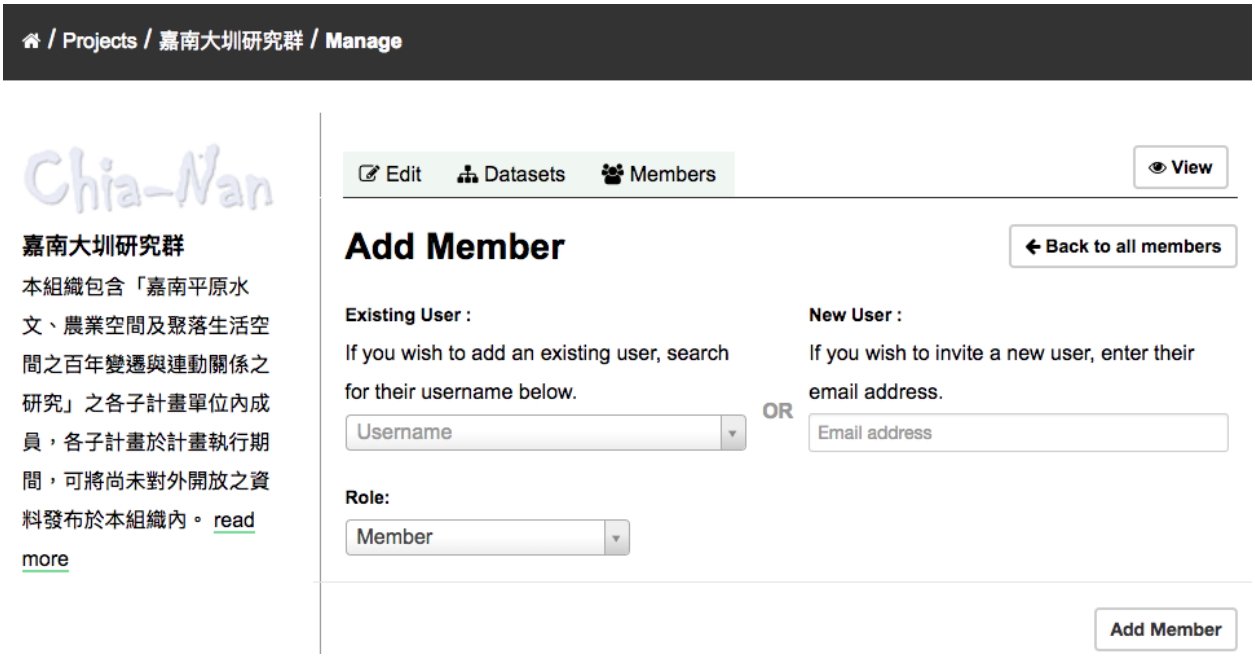
By default CKAN allows members of projects with three roles:

- *Member* – can see the project's private datasets
- *Editor* – can edit and publish datasets
- *Admin* – can add, remove and change roles for project members

Inviting others to project

If you want to invite others to collaborate on datasets, you can invite them to your project. From the project's page you should see an "Admin" button above the search box. When you select this, CKAN displays the project admin page.

Select the "Members" tab, and you will see the project members page. Then select the "Add Member" button.



You can invite an user to your project by his/her email or username in the "Existing User" section. Note that he/she must have an account.

Note: Due to the CKAN's privilege design, if the person you would like to invite does not have an account, please send an email with his/her email address to data.contact AT depositar.io. Then CKAN will send an invitation email to his/her.

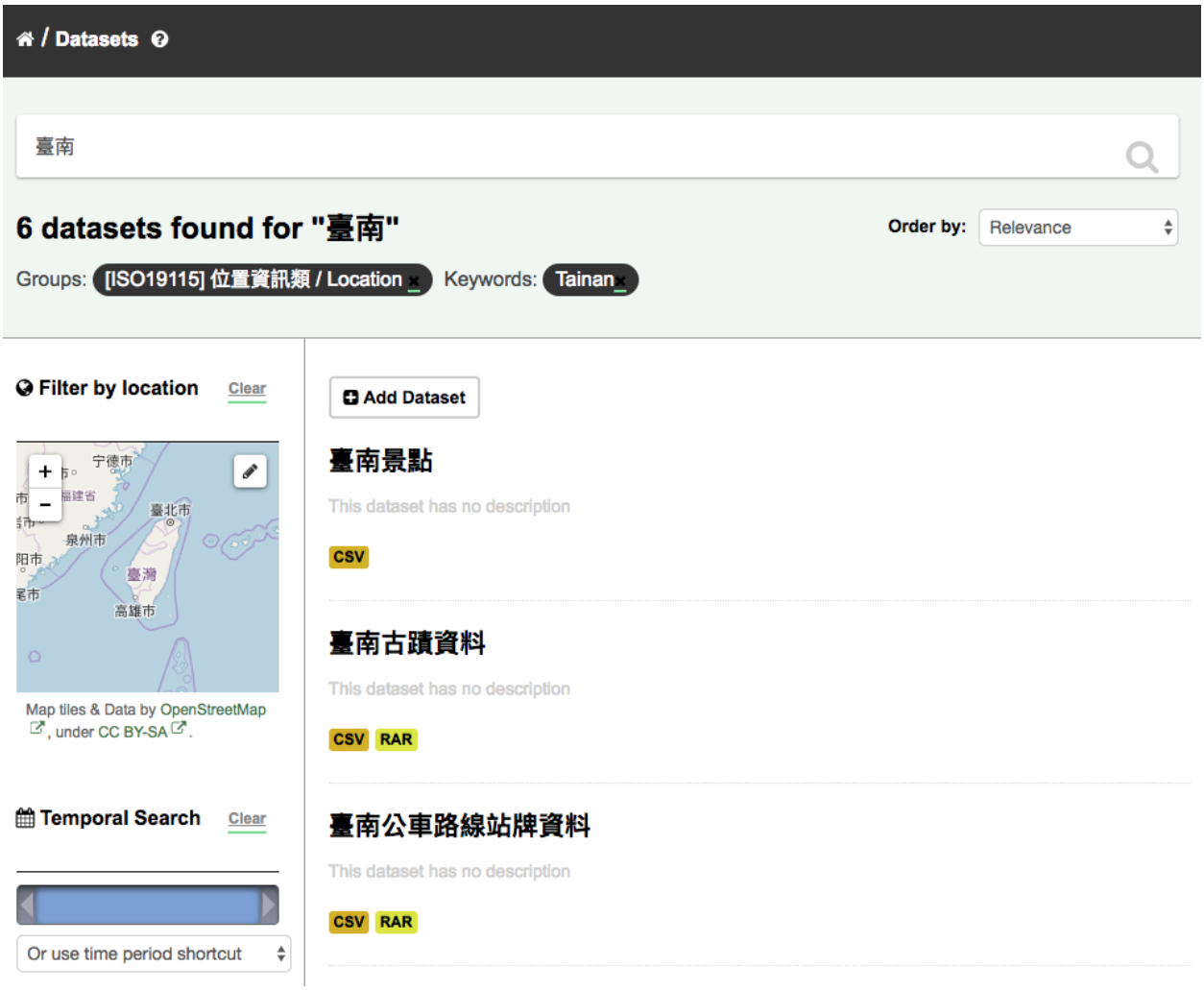
1.2.3 Finding data

Searching the site

To find datasets in CKAN, type any combination of search words (e.g. "health", "transport", etc) in the search box on any page. CKAN displays the first page of results for your search. You can:

- View more pages of results
- Repeat the search, altering some terms
- Restrict the search to datasets with particular tags, data formats, etc using the filters in the left-hand column

If there are a large number of results, the filters can be very helpful, since you can combine filters, selectively adding and removing them, and modify and repeat the search with existing filters still in place.

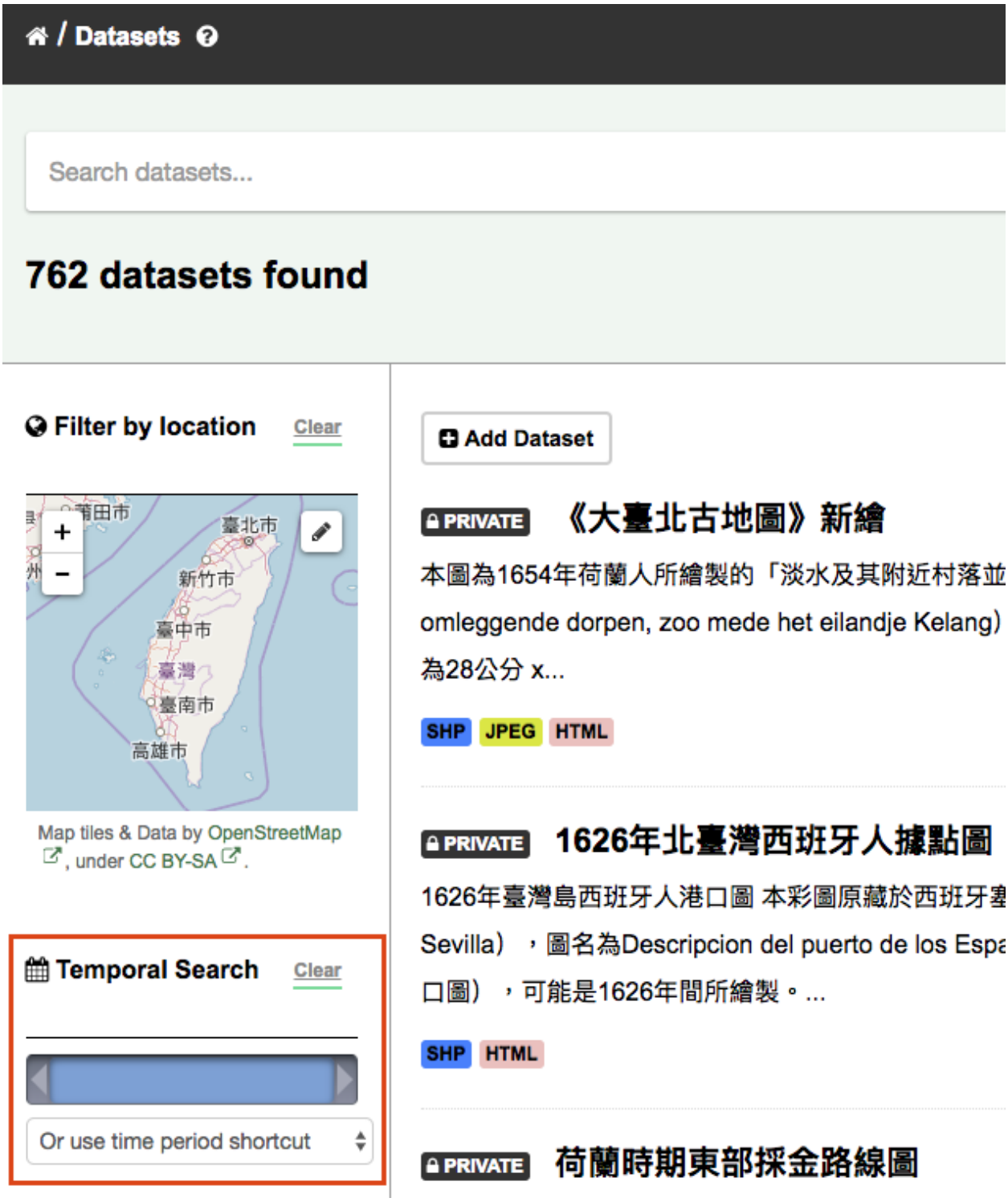


Extended feature — Temporal search

`depositar` has temporal search function. You can search for the datasets within a given date range.

You can find the temporal search widget from the left sidebar of the home page of datasets. You can do temporal search in two ways:

1. Use a range slider.
2. Use a time period shortcut which contains some historical periods.

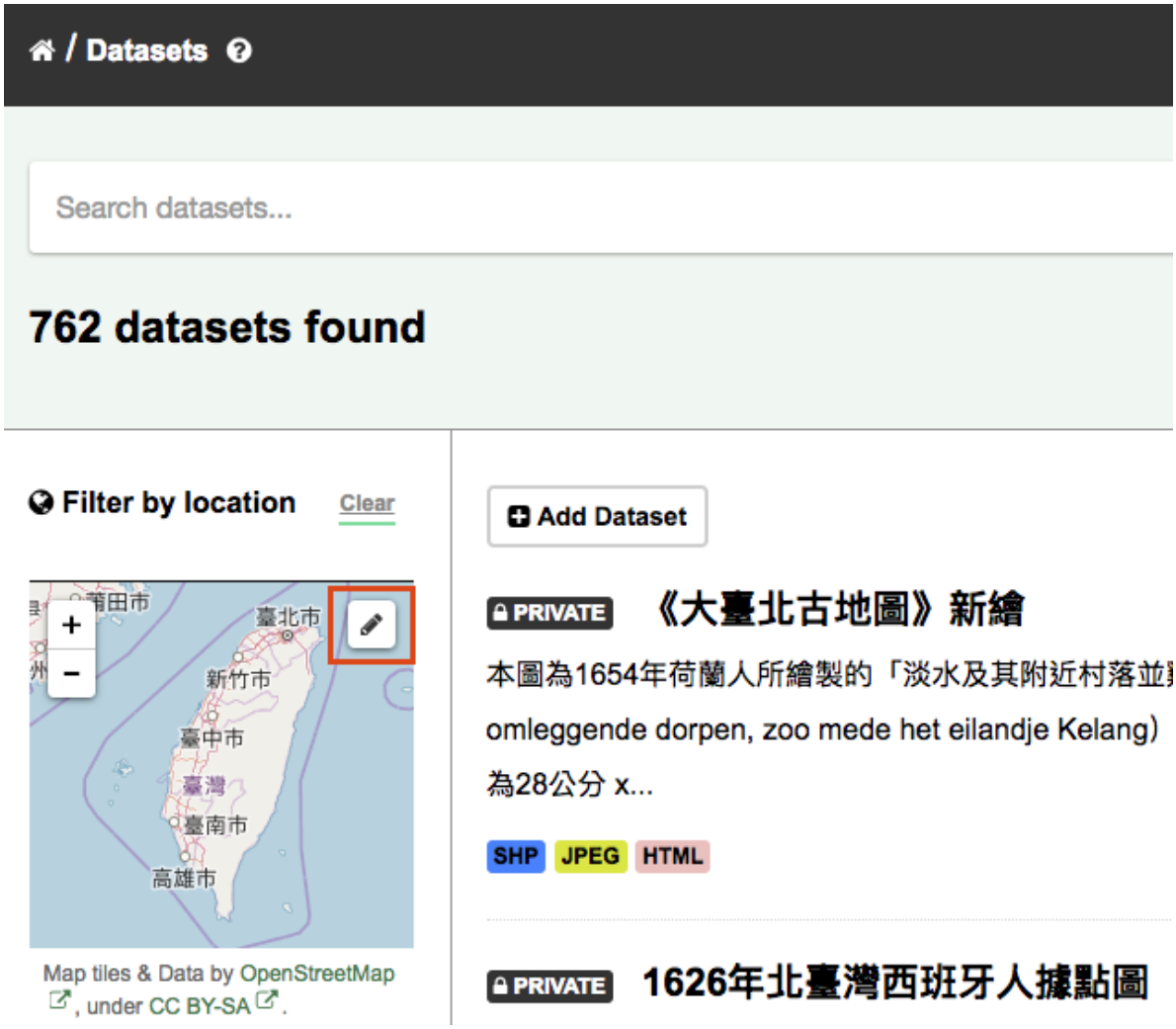


Extended feature — Spatial search

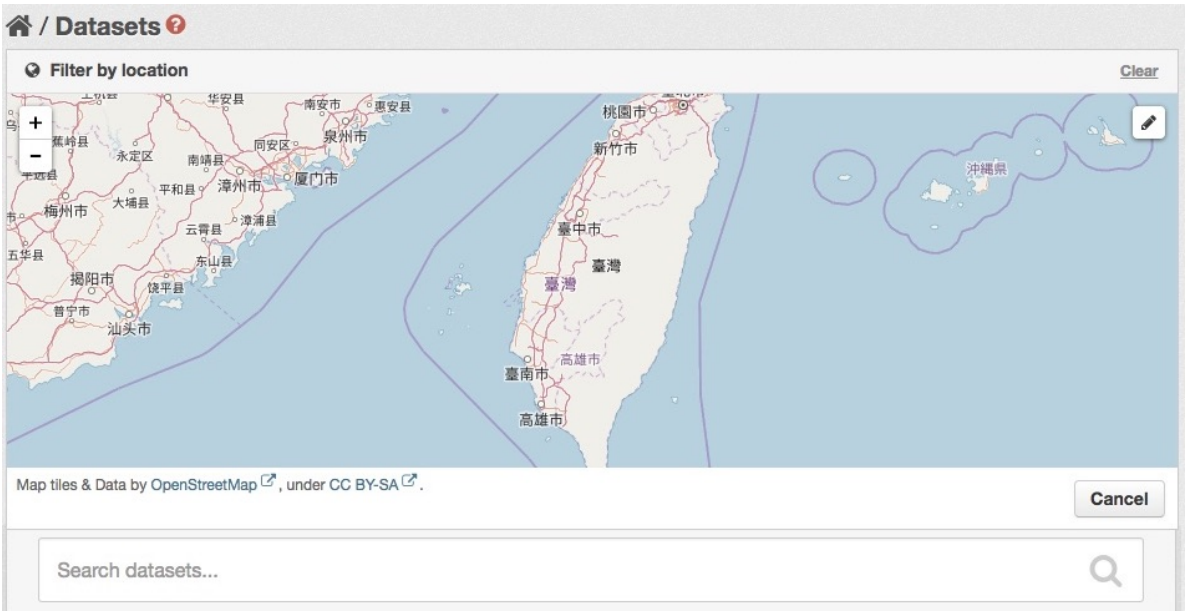
If datasets are tagged by geographical area in the `spatial` field (please refer to *Spatial Fields* for details), it is also possible to run CKAN with an extension which allows searching and filtering of datasets by selecting an area on a map.

You can find the spatial search widget from the left sidebar of the home page of datasets. You can do spatial search through the following steps:

1. Select the pencil icon in the upper-right corner:



2. Then you can draw a rectangle in the expanded map to specify a geographical area you are interested in:



3. The matched datasets will be shown up.
4. If you want to respecify a geographical area, please repeat step 1 and 2.

Searching within a project

If you want to look for data owned by a particular project, you can search within that project from its home page in CKAN.

1. Select the "Projects" link at the top of any page.
2. Select the project you are interested in. CKAN will display your project's home page.
3. Type your search query in the main search box on the page.

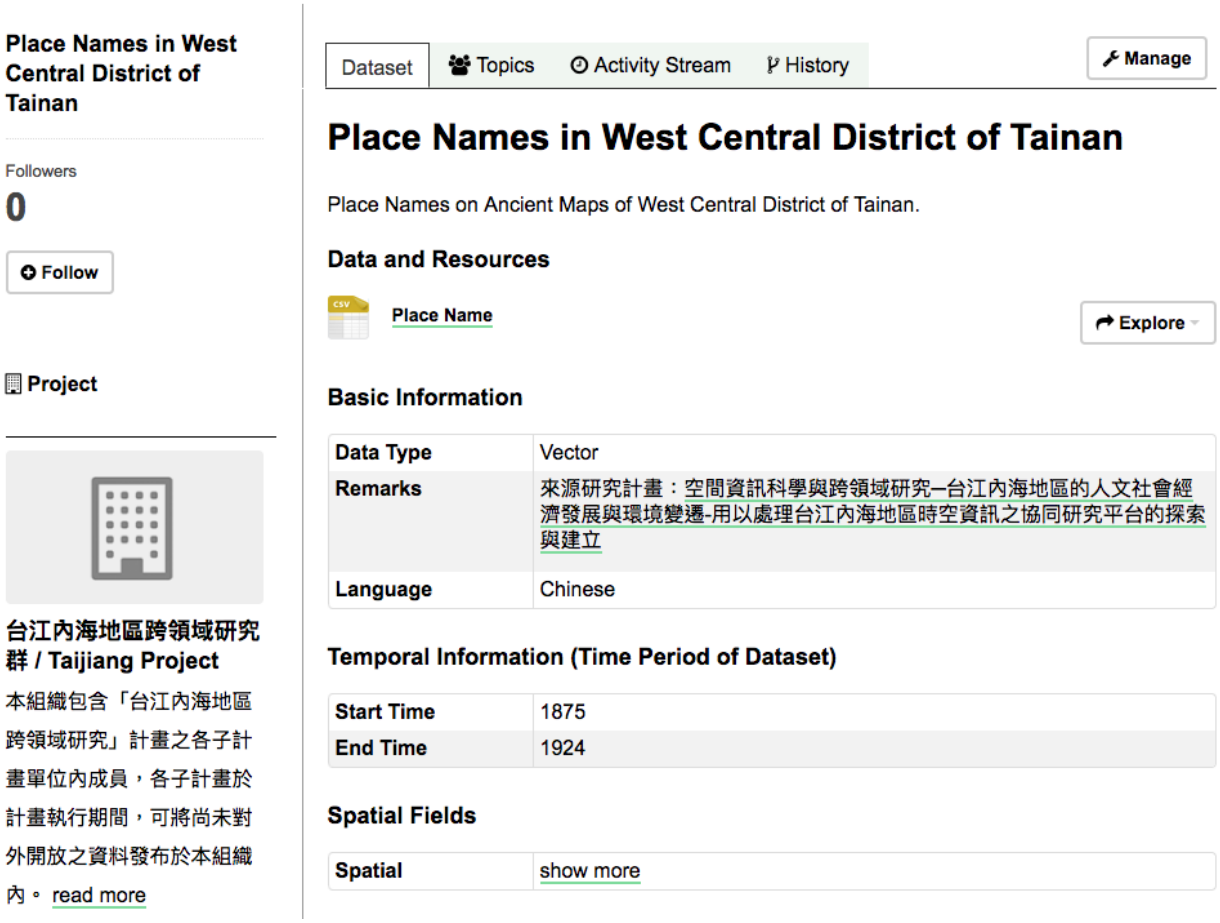
CKAN will return search results as normal, but restricted to datasets from the project.

If the project is of interest, you can opt to be notified of changes to it (such as new datasets and modifications to datasets) by using the "Follow" button on the project page. See the section *Managing your news feed* below. You must have a user account and be logged in to use this feature.

Exploring datasets

When you have found a dataset you are interested and selected it, CKAN will display the dataset page. This includes

- The name, description, and other information about the dataset
- Links to and brief descriptions of each of the resources



The resource descriptions link to a dedicated page for each resource. This resource page includes information about the resource, and enables it to be downloaded. Many types of resource can also be previewed directly on the resource page. .CSV and .XLS spreadsheets are previewed in a grid view, with map and graph views also available if the data is suitable. The resource page will also preview resources if they are common image types, PDF, or HTML.

The dataset page also has two other tabs:

- *Activity stream* – see the history of recent changes to the dataset
- *Topics* – see any topic associated with this dataset.

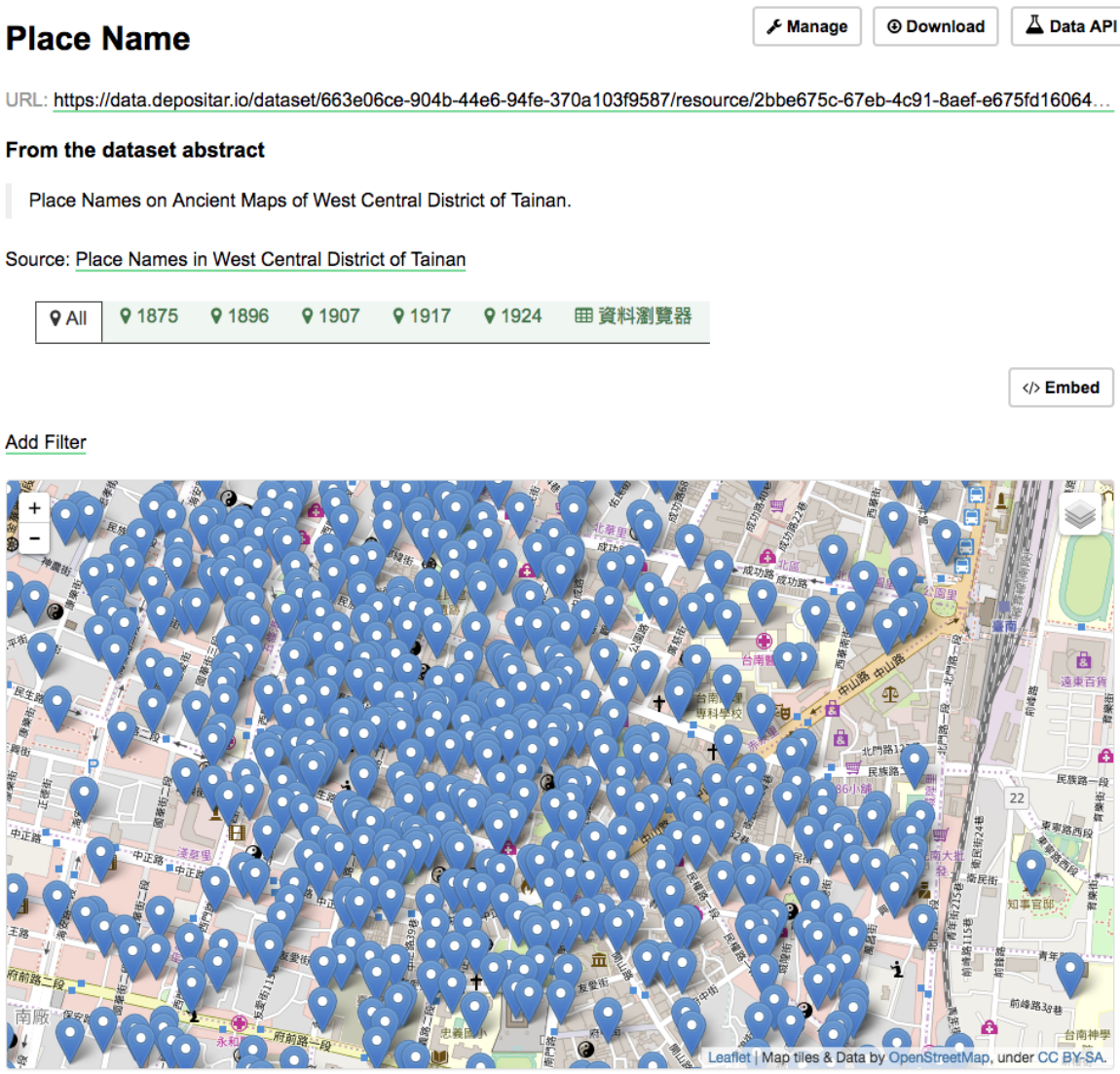
If the dataset is of interest, you can opt to be notified of changes to it by using the "Follow" button on the dataset page. See the section *Managing your news feed* below. You must have a user account and be logged in to use this feature.

Extended feature — Data preview and visualization

CKAN's data preview allows you learn the data without the need to download the entire file first:

1. Go to the dataset's page. You can find it by entering the title in the search box on any page.
2. Select the "Preview" button inside the "Explore" button beside a resource in the "Data and Resources" section:

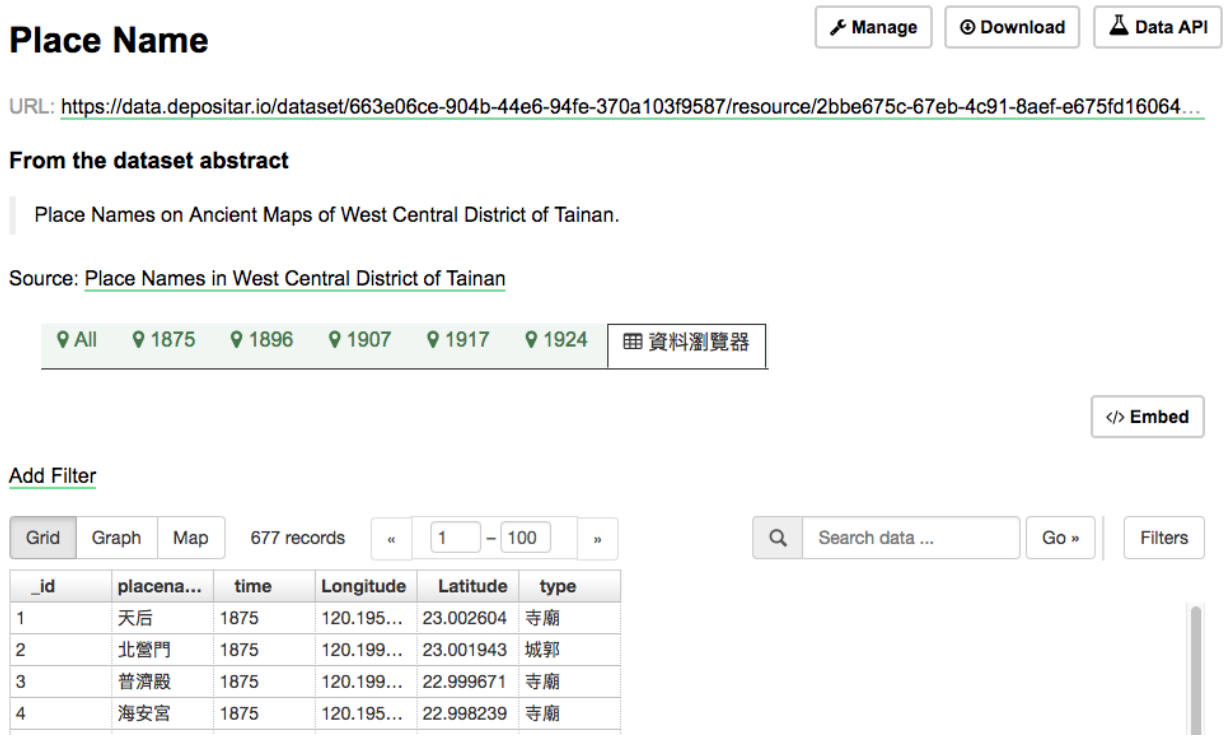
3. Then you can preview the resource:



The data preview function will check the *Format* field to specify a proper `resource view`. Please refer to step 5 of *Adding a new dataset*. `depositar` can preview the following formats:

- Text: txt, html, xml, json, and geojson
- Image: png, jpg, jpeg, and gif
- Table: csv and xls(x)
- Spatial data: WMTS, WMS, and Shapefile[2]
- Others: PDF and web page

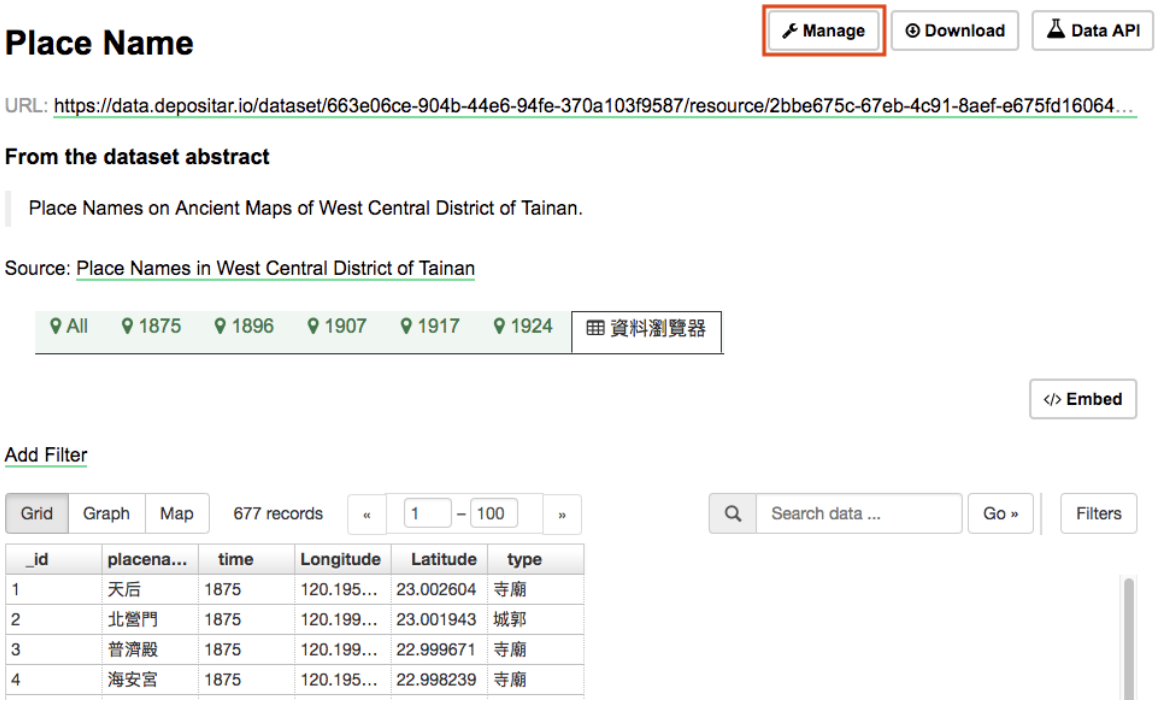
[2] Please specify the shapefile as "shp" in the `Format` field when filling out its information, otherwise it can not be visualized (Please refer to *Metadata at the resource level*).



One resource can have multiple views of the same data (for example a grid and some graphs for tabular data).

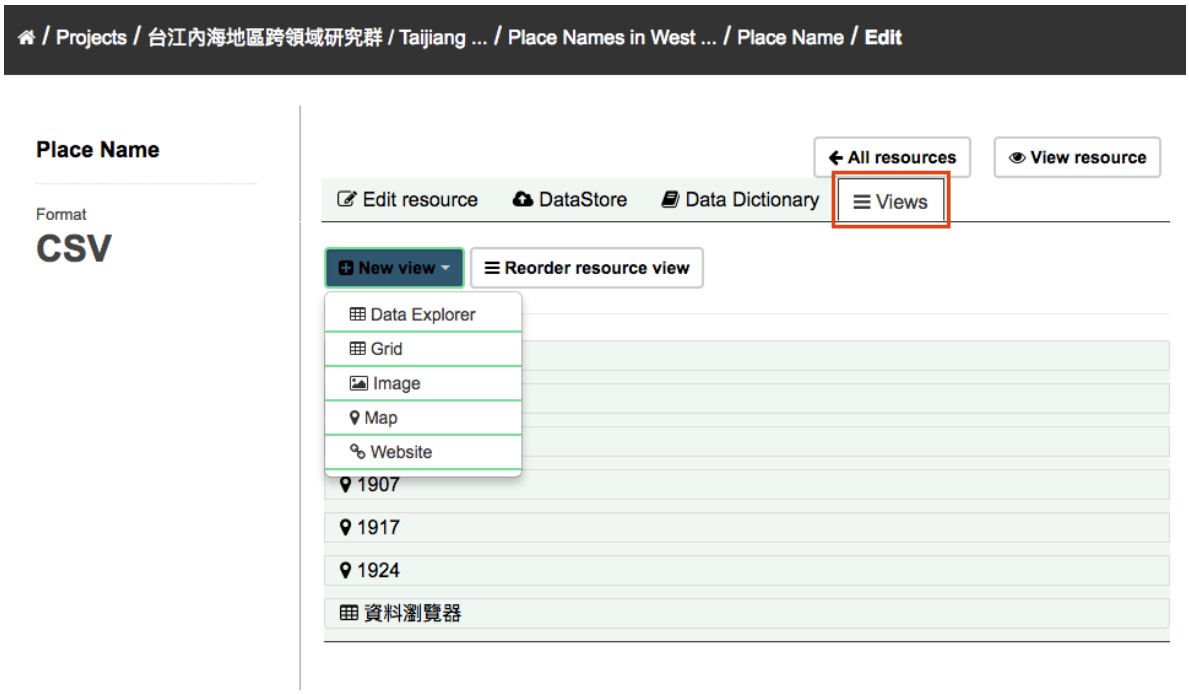
You can add a new resource view through the following steps:

1. Go to the resource's page.
2. Select the "Manage" button (You must have the right to edit the resource).



3. Select the "Views" tab in the next page. From here you can create new views, update or delete existing ones and reorder them. Available view plugins are:

- Data Explorer: It allows querying, filtering, graphing and mapping data.
- Grid: Displays a filterable, sortable, table view of structured data.
- Map: Shows data stored on the DataStore in an interactive map. It supports plotting markers from a pair of latitude / longitude fields or from a field containing a GeoJSON representation of the geometries.
- Image: If the resource format is a common image format like PNG, JPEG or GIF, it adds an `<img>` tag pointing to the resource URL.
- Web page: Adds an `<iframe>` tag to embed the resource URL.



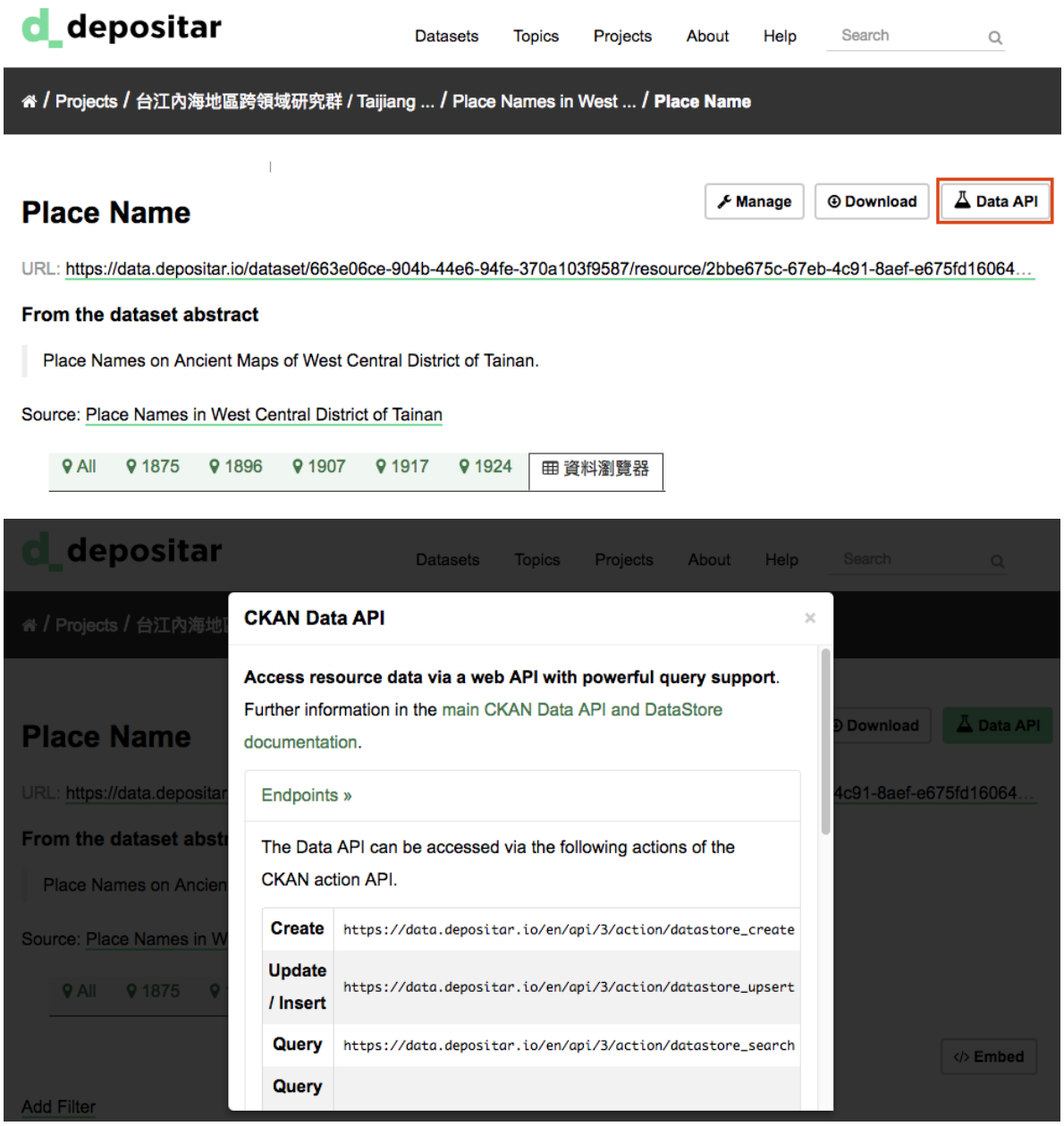
4. Select the "Add" button to save the new view. You can also take a sneak peek at the view by clicking the "Preview" button.

1.2.4 DataStore API

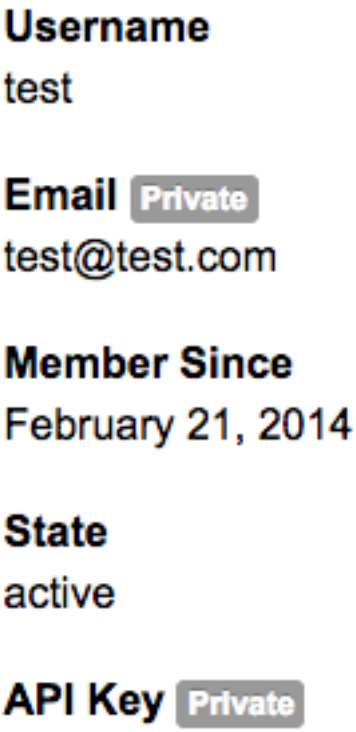
The CKAN DataStore extension provides an ad hoc database for storage of structured data from CKAN resources. It also offers an API for reading, searching and filtering data without the need to download the entire file first.

You can get access to DataStore API through the following steps:

1. Go to the resource's page.
2. Select the "Data API" button, a pop-up window will show how to use the API and provide some examples.



3. Some API functions require an API key. You can get your key from the user profile page using the "User" link at the top of any page:

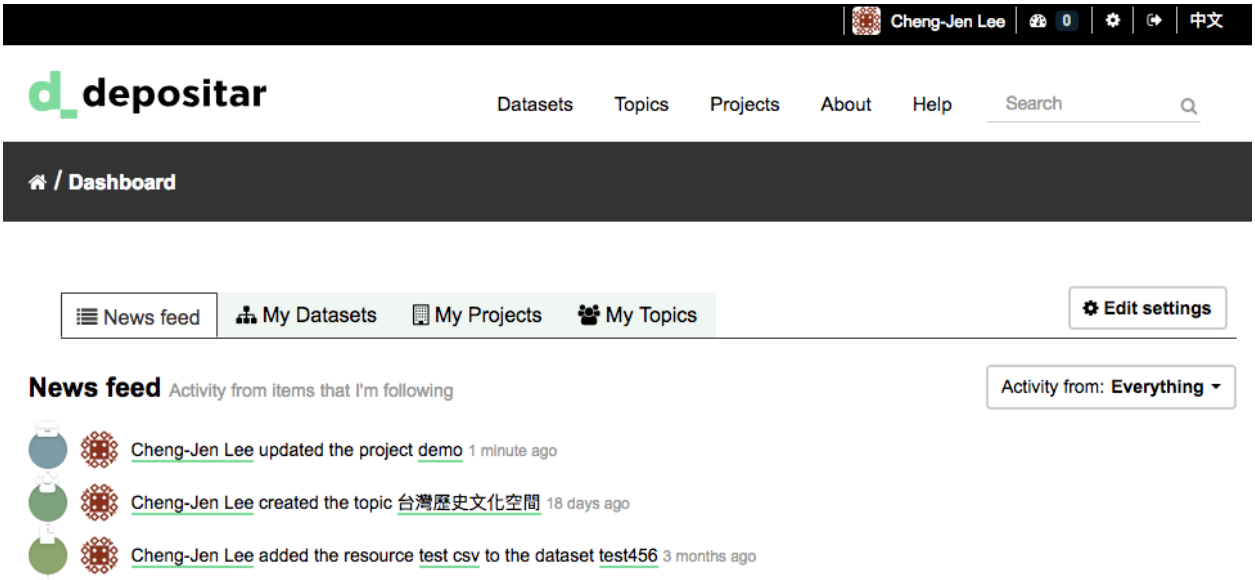


1.2.5 Personalization

CKAN provides features to personalize the experience of both searching for and publishing data. You must be logged in to use these features.

Managing your news feed

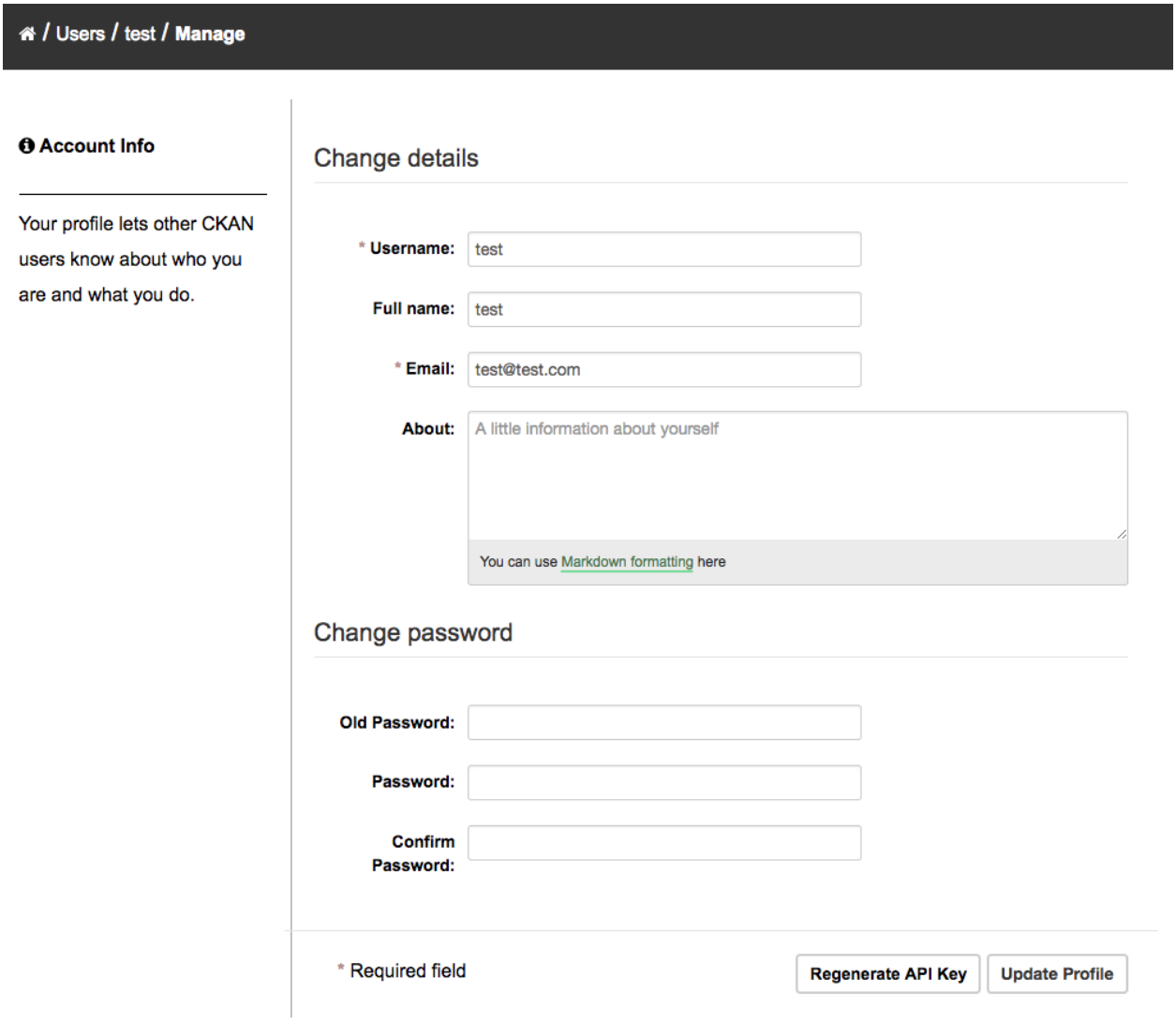
At the top of any page, select the dashboard symbol (next to your name). CKAN displays your News feed. This shows changes to datasets that you follow, and any changed or new datasets in projects that you follow. The number by the dashboard symbol shows the number of new notifications in your News feed since you last looked at it. As well as datasets and projects, it is possible to follow individual users (to be notified of changes that they make to datasets).



If you want to stop following a dataset (or project or user), go to the dataset's page (e.g. by selecting a link to it in your News feed) and select the "Unfollow" button.

Managing your user profile

You can change the information that CKAN holds about you, including what other users see about you by editing your user profile. (Users are most likely to see your profile when you edit a dataset or upload data to a project that they are following.) To do this, select the gearwheel symbol at the top of any page.



CKAN displays the user settings page. Here you can change:

- Your username
- Your full name
- Your e-mail address (note: this is not displayed to other users)
- Your profile text - an optional short paragraph about yourself
- Your password

Make the changes you require and then select the "Update Profile" button.

Note: If you change your username, CKAN will log you out. You will need to log back in using your new username.

1.2.6 System Limitation

- File size limit: up to around 1 GB.

- File size limit for data preview: up to around 20 MB for general format. Up to dozens of MB for PDFs.
- Filename length: 3 to 100 characters (including the filename extension).

CHAPTER 2

Maintainer's guide

Note: This manual is translated and adapted from User guide — CKAN 2.7.4 documentation by Open Knowledge International and contributors licensed under Creative Commons Attribution-ShareAlike 3.0 Unported.

2.1 Installing CKAN

`depositar` is installed from source at present.

Please refer to *Installing CKAN from source* and *Deploying a source install*.

2.1.1 Installing CKAN from source

This section describes how to install CKAN used by `depositar` from source on an Ubuntu 16.04 server.

1. Install the required packages

```
sudo apt-get install build-essential libxslt1-dev libxml2-dev python-dev postgresql␣
↪libpq-dev python-pip python-virtualenv git-core openjdk-8-jdk redis-server
```

2. Install CKAN into a Python virtual environment

1. Create a Python virtual environment (virtualenv) to install CKAN into, and activate it:

```
sudo mkdir -p /usr/lib/ckan/default
sudo chown `whoami` /usr/lib/ckan/default
virtualenv --no-site-packages /usr/lib/ckan/default
. /usr/lib/ckan/default/bin/activate
```

Important: The final command above activates your virtualenv. The virtualenv has to remain active for the rest of the installation and deployment process, or commands will fail. You can tell when the virtualenv is active because its name appears in front of your shell prompt, something like this:

```
(default) $ _
```

For example, if you logout and login again, or if you close your terminal window and open it again, your virtualenv will no longer be activated. You can always reactivate the virtualenv with this command:

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

2. Install the recommended setuptools version:

   Important: Please run all the commands below under the *ckan* directory:

   ```
   cd /usr/lib/ckan/default/
   ```

   ```
   pip install setuptools==36.1
   ```

3. Install the CKAN source code and customized extesion into your virtualenv.

   ```
   pip install -e 'git+https://github.com/depositar-io/ckanext-data-depositario.git
   ↪#egg=ckanext-data-depositario'
   ```

4. Install the Python modules that customized extesion requires into your virtualenv:

   ```
   pip install -r /usr/lib/ckan/default/src/ckanext-data-depositario/requirements.txt
   ```

5. Install the Python modules that CKAN requires into your virtualenv:

   ```
   pip install -r /usr/lib/ckan/default/src/ckan/requirements.txt
   ```

6. Install other required Python modules into your virtualenv:

   ```
   pip install -r /usr/lib/ckan/default/src/ckanext-spatial/pip-requirements.txt
   pip install -r /usr/lib/ckan/default/src/ckanext-scheming/requirements.txt
   ```

3. Install DataPusher into a Python virtual environment

Note: This DataPusher is a service that automatically uploads data to the DataStore from suitable files (like CSV or Excel files), whether uploaded to CKAN's FileStore or externally linked.

The CKAN DataStore extension provides an ad hoc database for storage of structured data from CKAN resources. Data can be pulled out of resource files and stored in the DataStore.

1. Create a Python virtual environment (virtualenv) to install DataPusher into, and activate it:

```
sudo mkdir -p /usr/lib/ckan/datapusher
sudo chown `whoami` /usr/lib/ckan/datapusher
virtualenv --no-site-packages /usr/lib/ckan/datapusher
. /usr/lib/ckan/datapusher/bin/activate
```

2. Install the DataPusher source code into your virtualenv:

Important: Please run all the commands below under the *ckan* directory:

```
cd /usr/lib/ckan/datapusher/
```

Install the DataPusher

```
pip install -e 'git+https://github.com/ckan/datapusher.git#egg=datapusher'
```

3. Install the Python modules that DataPusher requires into your virtualenv:

```
pip install -r /usr/lib/ckan/datapusher/src/datapusher/requirements.txt
```

4. Create the FireStore directory

Note: When enabled, CKAN's FileStore allows users to upload data files to CKAN resources. Please refer to *User guide* for details.

```
sudo mkdir -p /var/lib/ckan/default
sudo chown `whoami` /var/lib/ckan/default
sudo chmod u+rwx /var/lib/ckan/default
```

5. Setup a PostgreSQL database

1. Create a database user:

```
sudo -u postgres createuser -S -D -R -P ckan_default
```

2. Create a new database:

```
sudo -u postgres createdb -O ckan_default ckan_default -E utf-8
```

3. Install the PostGIS:

```
sudo apt-get install postgresql-9.5-postgis-2.2 python-dev libxml2-dev libxslt1-
↪dev libgeos-c1v5
sudo -u postgres psql -d ckan_default -f /usr/share/postgresql/9.5/contrib/
↪postgis-2.2/postgis.sql
sudo -u postgres psql -d ckan_default -f /usr/share/postgresql/9.5/contrib/
↪postgis-2.2/spatial_ref_sys.sql
sudo -u postgres psql -d ckan_default -c 'ALTER VIEW geometry_columns OWNER TO␣
↪ckan_default;'
sudo -u postgres psql -d ckan_default -c 'ALTER TABLE spatial_ref_sys OWNER TO␣
↪ckan_default;'
```

4. Create a new database user and a new database for DataStore:

   Note: The CKAN DataStore extension provides an ad hoc database for storage of structured data from CKAN resources. Data can be pulled out of resource files and stored in the DataStore.

```
sudo -u postgres createuser -S -D -R -P -l datastore_default
sudo -u postgres createdb -O ckan_default datastore_default -E utf-8
```

5. (For `depositar` administrator) Restore database backup:

```
gunzip -c main_db.sql.gz | sudo -u postgres psql ckan_default
gunzip -c datastore_db.sql.gz | sudo -u postgres psql datastore_default
```

6. Create a CKAN config file

1. Create a directory to contain the site's config files:

```
sudo mkdir -p /etc/ckan/default
sudo chown -R `whoami` /etc/ckan/
```

2. Create the CKAN config file via paster:

   Important: (For `depositar` administrator) Please ignore the following step. c and use `production.ini` the in the `configs.tar.gz`.

   Important: The virtualenv has to remain active when running the paster command. You can always reactivate the virtualenv with this command:

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

```
paster make-config ckan /etc/ckan/default/development.ini
```

3. Edit the development.ini file in a text editor, changing the following options:

   Note:

   - The settings below is the minimum requirements to run the CKAN.

```
## Database Settings
## This should refer to the database we created in 5. Setup a PostgreSQL␣
↪database above
## Replace pass with the CKAN database password that you created
sqlalchemy.url = postgresql://ckan_default:pass@localhost/ckan_default
## Replace pass with the CKAN database password that you created
ckan.datastore.write_url = postgresql://ckan_default:pass@localhost/
↪datastore_default
## Replace pass with the DataStore database password that you created
ckan.datastore.read_url = postgresql://datastore_default:pass@localhost/
↪datastore_default
```

```
## Site Settings
ckan.site_url = http://127.0.0.1:5000

## Plugins Settings
ckan.plugins = data_depositario wikidatakeyword stats datastore datapusher
               resource_proxy recline_view text_view image_view
               webpage_view recline_grid_view recline_map_view
               pdf_view spatial_metadata spatial_query
               geo_view geojson_view wmts_view shp_view
               scheming_datasets repeating

## Front-End Settings
licenses_group_url = file:///usr/lib/ckan/default/src/ckanext-data-
↪depositario/ckanext/data_depositario/public/license_list.json

## Storage Settings
ckan.storage_path = /var/lib/ckan/default

## Datapusher Settings
ckan.datapusher.url = http://0.0.0.0:8800/

## Schema Settings
## Add these settings
scheming.presets = ckanext.scheming:presets.json
                   ckanext.repeating:presets.json
                   ckanext.data_depositario:presets.json
                   ckanext.wikidatakeyword:presets.json
scheming.dataset_schemas = ckanext.data_depositario:scheming.json

## Spatial Settings
## Add these settings
ckanext.spatial.search_backend = solr-spatial-field

## ckanext-data-depositario Settings
## Add these settings
## GMAP_AKI_KEY is the API key for Google Maps
ckanext.data_depositario.gmap.api_key = GMAP_AKI_KEY
## GA_ID is the id for Google Analytics
ckanext.data_depositario.googleanalytics.id = GA_ID
```

7. Setup Solr (with Chinese and spatial search support)

Note: This section is adapted from How To Install Solr 5.2.1 on Ubuntu 14.04 by DigitalOcean™ Inc. licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International.

1. Download and extract the service installation file:

```
cd ~
wget http://archive.apache.org/dist/lucene/solr/5.5.5/solr-5.5.5.tgz
tar xzf solr-5.5.5.tgz solr-5.5.5/bin/install_solr_service.sh --strip-components=2
```

2. Install Solr as a service using the script:

```
sudo bash ./install_solr_service.sh solr-5.5.5.tgz
```

3. Create the Solr configset for CKAN:

```
sudo -u solr mkdir -p /var/solr/data/configsets/ckan/conf
sudo ln -s /usr/lib/ckan/default/src/ckanext-data-depositario/solr/schema.xml /
↪var/solr/data/configsets/ckan/conf/schema.xml
sudo -u solr cp /opt/solr/server/solr/configsets/basic_configs/conf/solrconfig.
↪xml /var/solr/data/configsets/ckan/conf/.
sudo -u solr touch /var/solr/data/configsets/ckan/conf/protwords.txt
sudo -u solr touch /var/solr/data/configsets/ckan/conf/synonyms.txt
```

4. Download Chinese tokenizer `Mmseg4j` and copy it to the Solr directory:

```
wget http://central.maven.org/maven2/com/chenlb/mmseg4j/mmseg4j-core/1.10.
0/mmseg4j-core-1.10.0.jar
wget http://central.maven.org/maven2/com/chenlb/mmseg4j/mmseg4j-solr/2.3.
1/mmseg4j-solr-2.3.1.jar
sudo cp mmseg4j-*.jar /opt/solr/server/solr-webapp/webapp/WEB-INF/lib/.
```

5. Download geometry library JTS Topology Suite 1.13 (or above) and copy it to the Solr directory:

```
wget -O jts-1.13.jar https://search.maven.org/remotecontent?filepath=com/
↪vividsolutions/jts/1.13/jts-1.13.jar
sudo cp jts-1.13.jar /opt/solr/server/solr-webapp/webapp/WEB-INF/lib/.
```

6. Restart Solr:

```
sudo service solr restart
```

7. Create a new Solr core called `ckan` by entering the following link in a web browser:

   http://127.0.0.1:8983/solr/admin/cores?action=CREATE&name=ckan&configSet=ckan

8. Open http://127.0.0.1:8983/solr/#/ckan in a web browser, and you should see the Solr front page.
9. Modify /etc/ckan/default/development.ini with Solr url:

```
solr_url = http://127.0.0.1:8983/solr/ckan
```

8. Create database tables

Important: (For `depositar` administrator) Please ignore this step.

1. Create the database tables via paster:

```
paster --plugin=ckan db init -c /etc/ckan/default/development.ini
```

2. You should see Initialising DB: SUCCESS.
3. Then you can use this connection to set the permissions for DataStore:

```
paster --plugin=ckan datastore set-permissions -c /etc/ckan/default/development.
↪ini | sudo -u postgres psql --set ON_ERROR_STOP=1
```

9. Link to `who.ini`

```
ln -s /usr/lib/ckan/default/src/ckan/who.ini /etc/ckan/default/who.ini
```

10. Creating a sysadmin user

Important: (For `depositar` administrator) Please ignore this step.

You have to create your first CKAN sysadmin user from the command line. For example, to create a user called *admin* and make him a sysadmin:

```
paster --plugin=ckan sysadmin add admin email=admin@localhost -c /etc/ckan/default/
↪development.ini
```

11. Serve CKAN under a development server

1. Run the DataPusher:

   ```
   . /usr/lib/ckan/datapusher/bin/activate
   JOB_CONFIG='/usr/lib/ckan/datapusher/src/datapusher/deployment/datapusher_
   ↪settings.py' python /usr/lib/ckan/datapusher/src/datapusher/wsgi.py
   ```

2. Open another terminal and use the Paste development server to serve CKAN from the command-line:

   ```
   . /usr/lib/ckan/default/bin/activate
   paster serve /etc/ckan/default/development.ini
   ```

3. Open http://127.0.0.1:5000/ in a web browser, and you should see the CKAN front page.

Now that you've installed CKAN.

2.1.2 Deploying a source install

Since CKAN is written mainly in Pylons and supports WSGI, CKAN can be used with a number of different web server and deployment configurations.

This guide explains how to deploy CKAN using Gunicorn and proxied with Nginx on an Ubuntu server. These instructions have been tested on Ubuntu 16.04.

1. Create a `production.ini` file

Important: (For `depositar` administrator) Please ignore this step and use `production.ini` the in the `configs.tar.gz`.

```
cp /etc/ckan/default/development.ini /etc/ckan/default/production.ini
```

2. Modify the `production.ini` file

Important: (For `depositar` administrator) Please ignore this step.

Edit the production.ini file in a text editor, changing the [server:main] and [app:main] sections as follows:

```
[server:main]
#use = egg:Paste#http
#host = 0.0.0.0
#port = 5000
use = egg:gunicorn#main
bind = unix:/var/run/gunicorn/ckan_socket.sock
preload = true
## The Error log file to write to.
errorlog = /etc/ckan/default/ckan.log
loglevel = warning
## USER is the owner of /etc/ckan/default
user = USER
group = www-data
umask = 0113

[app:main]
...
## Site Settings

ckan.site_url = http://127.0.0.1
```

3. Install Gunicorn

Install Gunicorn into a Python virtual environment:

```
. /usr/lib/ckan/default/bin/activate
pip install gunicorn

. /usr/lib/ckan/datapusher/bin/activate
pip install gunicorn
```

4. Set the startup script for CKAN

1. Create a Systemd service for CKAN:

   ```
   sudo vi /etc/systemd/system/ckan.service
   ```

2. In the vi editor, add the following contents:

   ```
   [Unit]
   Description=Gunicorn instance to serve CKAN
   After=network.target

   [Service]
   WorkingDirectory=/usr/lib/ckan/default/src/ckan
   RuntimeDirectory=gunicorn
   ```

(continues on next page)

(continued from previous page)

```
ExecStart=/usr/lib/ckan/default/bin/gunicorn --paste /etc/ckan/default/production.
↪ini
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s TERM $MAINPID
StandardError=syslog
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

3. Start the Systemd service:

```
sudo systemctl enable ckan
```

4. To start the installed service, run the following command:

```
sudo service ckan start
```

5. You can check the site status via:

```
sudo service ckan status
```

You should now be able to see the following output:

```
 ckan.service - Gunicorn instance to serve CKAN
   Loaded: loaded (/etc/systemd/system/ckan.service; enabled; vendor preset:␣
↪enabled)
   Active: active (running) since Thr 2017-12-14 14:36:37 CST; 2s ago
  Process: 20152 ExecStop=/bin/kill -s TERM $MAINPID (code=exited, status=0/
↪SUCCESS)
 Main PID: 20191 (gunicorn)
    Tasks: 2
   Memory: 88.0M
      CPU: 1.596s
   CGroup: /system.slice/ckan.service
           ├─20191 /usr/lib/ckan/default/bin/python2 /usr/lib/ckan/default/bin/
↪gunicorn --paste /etc/ckan/default/production.ini
           └─20198 /usr/lib/ckan/default/bin/python2 /usr/lib/ckan/default/bin/
↪gunicorn --paste /etc/ckan/default/production.ini
```

6. You can stop the Systemd service by:

```
sudo service ckan stop
```

5. Set the startup script for DataPusher

Note: This DataPusher is a service that automatically uploads data to the DataStore from suitable files (like CSV or Excel files), whether uploaded to CKAN's FileStore or externally linked.

1. Create a Systemd service for DataPusher:

```
sudo vi /etc/init/datapusher.conf
```

2. In the vi editor, add the following contents:

```
[Unit]
Description=Gunicorn instance to serve DataPusher
After=network.target

[Service]
RuntimeDirectory=gunicorn
Environment=JOB_CONFIG=/usr/lib/ckan/datapusher/src/datapusher/deployment/
↪datapusher_settings.py
ExecStart=/usr/lib/ckan/datapusher/bin/gunicorn wsgi:app
ExecReload=/bin/kill -s HUP $MAINPID
ExecStop=/bin/kill -s TERM $MAINPID
StandardOutput=null
StandardError=null
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

3. Start the Systemd service:

```
sudo systemctl enable datapusher
```

4. To start the installed service, run the following command:

```
sudo service datapusher start
```

5. You can check the status via:

```
sudo service datapusher status
```

You should now be able to see the following output:

```
 datapusher.service - Gunicorn instance to serve DataPusher
   Loaded: loaded (/etc/systemd/system/datapusher.service; enabled; vendor␣
↪preset: enabled)
   Active: active (running) since Thr 2017-12-14 14:48:44 CST; 2min 44s ago
  Process: 20571 ExecStop=/bin/kill -s TERM $MAINPID (code=exited, status=0/
↪SUCCESS)
 Main PID: 20626 (gunicorn)
    Tasks: 2
   Memory: 46.0M
      CPU: 1.790s
   CGroup: /system.slice/datapusher.service
           ├─20626 /usr/lib/ckan/datapusher/bin/python2 /usr/lib/ckan/datapusher/
↪bin/gunicorn wsgi:app
           └─20673 /usr/lib/ckan/datapusher/bin/python2 /usr/lib/ckan/datapusher/
↪bin/gunicorn wsgi:app
```

6. You can stop the Systemd service by:

```
sudo service datapusher stop
```

6. Install and setup Nginx

1. Install Nginx:

```
sudo apt-get install nginx
```

b. Create your site's Nginx config file at /etc/nginx/sites-available/ckan, with the following contents:

```
proxy_cache_path /tmp/nginx_cache levels=1:2 keys_zone=cache:30m max_
↪size=250m;

server {
    listen 80;
    server_name 127.0.0.1;
    client_max_body_size 1000M;
    access_log /var/log/nginx/ckan_access.log;
    error_log /var/log/nginx/ckan_error.log error;

    location / {
        try_files $uri @proxy_to_app;
    }

    location @proxy_to_app {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        # enable this if and only if you use HTTPS
        # proxy_set_header X-Forwarded-Proto https;
        proxy_set_header Host $http_host;
        # we don't want nginx trying to do something clever with
        # redirects, we set the Host: header above already.
        proxy_redirect off;
        proxy_pass http://unix:/var/run/gunicorn/ckan_socket.sock;
    }
}
```

3. To prevent conflicts, disable your default Nginx sites. Finally, enable your CKAN site in Nginx:

```
sudo rm /etc/nginx/sites-enabled/default
sudo ln -s /etc/nginx/sites-available/ckan /etc/nginx/sites-enabled/ckan
```

4. Restart Nginx:

```
sudo service nginx restart
```

7. Test the site

You should now be able to visit your server (at http://127.0.0.1) in a web browser and see your new CKAN instance.

2.2 Translating `depositar`

2.2.1 String internationalization

Our customized extension can be internationalized. This guide shows how to internationalize strings.

Note: This is the simplified version. For details please refer to String internationalization in CKAN's documentation.

1. Internationalizing strings in Jinja2 templates

   To internationalize a string put it inside a `_()` function:

   ```
   {% set hello = _('Hello World!') %}
   ```

2. Internationalizing strings in Python code

   To internationalize a string put it inside a `_()` function:

   ```
   my_string = _("This paragraph is translatable.")
   ```

3. Internationalizing strings in JavaScript code

   To internationalize a string put it inside a `this._()` function:

   ```
   this._('Something that should be translated')
   ```

2.2.2 Extract strings and edit translations

Before editing translations, you should extract strings from the customized extension.

Important: The virtualenv has to remain active for the rest of the installation and deployment process, or commands will fail. You can tell when the virtualenv is active because its name appears in front of your shell prompt, something like this:

```
(default) $ _
```

For example, if you logout and login again, or if you close your terminal window and open it again, your virtualenv will no longer be activated. You can always reactivate the virtualenv with this command:

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

Important: Please run all the commands below under the directory where the customized extension is installed.

```
cd /usr/lib/ckan/default/src/ckanext-data-depositario
```

---

1. Extract strings to be translated:

   ```
   python setup.py extract_messages
   ```

2. Create translation files for a locale:

   Note: We will create translation files for the `zh_TW` locale.

   Since the `zh_TW` locale already exists, we run the `update_catalog` command to keep the translated strings. If you want to create a new locale, please use `init_catalog` instead.

   ```
   python setup.py update_catalog -l zh_TW
   ```

3. Open the generated translation file and add translations for it by editing the `msgstr` section:

```
vi ckanext/data_depositario/i18n/zh_TW/LC_MESSAGES/ckanext-data_depositario.po
```

4. Compiling the catalog:

```
python setup.py compile_catalog
```

5. Restart CKAN:

Note: In this tutorial we are assuming that you have finished *Deploying a source install*.

```
sudo stop ckan && sudo start ckan
```

2.3 Writing documentation

Note: For details please refer to Writing documentation.

2.3.1 Getting started

Install documentation into a Python virtual environment

Create a Python virtual environment (virtualenv) to install documentation into, and activate it:

```
sudo apt-get install python-dev libpq-dev python-pip python-virtualenv
virtualenv --no-site-packages pyenv
. pyenv/bin/activate
pip install -e 'git+https://github.com/depositar-io/ckanext-data-depositario.git
↪#egg=ckanext-data-depositario'
cd pyenv/src/ckanext-data-depositario
pip install -r requirements-docs.txt
cd ../ckan
pip install -r requirements.txt
pip install setuptools==36.1
```

Edit the reStructuredText files

`depositar`'s documentation is created using Sphinx. To make changes to the documentation, use a text editor to edit the `.rst` files in `pyenv/src/ckanext-data-depositario/doc`. Some useful links to bookmark:

- Sphinx's reStructuredText Primer
- reStructuredText cheat sheet
- reStructuredText quick reference
- Sphinx Markup Constructs is a full list of the markup that Sphinx adds on top of Docutils.

2.3.2 Build the docs

You should now be able to build the CKAN documentation locally. Make sure your virtual environment is activated, and then run the these commands:

```
cd ../ckanext-data-depositario
python setup.py build_sphinx
```

Now you can open the built HTML files in `build/sphinx/html`.

Important: When you build the docs, Sphinx prints out warnings about any broken cross-references, syntax errors, etc. We aim not to have any of these warnings, so when adding to or editing the docs make sure your changes don't introduce any new ones.

It's best to delete the `build` directory and completely rebuild the docs, to check for any warnings:

```
rm -rf build; python setup.py build_sphinx
```

2.3.3 Publish the docs

`depositar`'s documentation is published using ReadTheDocs. ReadTheDocs will detect each change to the `ckanext-data-depositario` repository and build a new version.

CHAPTER 3

Appendix

3.1 Metadata at the dataset level

CKAN allows administrators to customize the metadata on the basis of the characteristics of collected datasets. In the `depositar`, we have organized the metadata at the dataset level into three categories: `Basic Information`, `Descriptive Information`, and `Management Information`.

3.1.1 Basic Information

Field Name	Description	Mandatory (M)/Optional (O)	Maximum Occurrence	Data Type[1]	Validators and Converters[2]
Title	It is recommended to make it brief but specific. E.g. "Taiwan population density by region" is better than "Population figures".	O	1	gco:CharacterString	*if_empty_same_as(name) unicode*
URL[3]	This URL will be unique across CKAN. Only letters, numbers, - and _ characters are accepted.	M	1	gco:CharacterString	*not_empty unicode package_name_validator*
Description	You can add a longer description of the dataset here, including information such as where the data is from and any information that people will need to know when using the data.	O	1	gco:CharacterString	
Data Type	The type of the dataset. Fields will be changed according to the data type.	M	1	Data_type	*scheming_required scheming_choices*
Tags[4]	Here you may add tags that will help people find the data and link it with other related data.	O	N	gco:CharacterString	*ignore_missing tag_string_convert*
Remarks	You can put some supplementary information for the dataset here.	O	1	gco:CharacterString	

3.1.2 Descriptive Information

Field Name	Description	Mandatory (M)/Optional (O)	Maximum Occurrence	Data Type	Validators and Converters
Language	The language of the dataset (e.g., Chinese or Japanese).	O	1	Language_type	*scheming_required schem-ing_choices*
Temporal Resolution	Please refer to *Fill-in snippet for temporal information*	O	1	Temp_res_type	*scheming_required schem-ing_choices*
Start Time	Please refer to *Fill-in snippet for temporal information*	O	1	gco:Date	*ignore_empty temp_res_validator*
End Time	Please refer to *Fill-in snippet for temporal information*	O	1	gco:Date	*ignore_empty temp_res_validator*
Spatial	Please refer to *Fill-in snippet for spatial fields*	O	1	GeoJSON	*ignore_empty json_validator re-move_blank_wrap*
X.min	Please refer to *Fill-in snippet for spatial fields*	O	1	gco:Decimal	*ignore_empty long_validator*
X.max	Please refer to *Fill-in snippet for spatial fields*	O	1	gco:Decimal	*ignore_empty long_validator*
Y.min	Please refer to *Fill-in snippet for spatial fields*	O	1	gco:Decimal	*ignore_empty lat_validator*
Y.max	Please refer to *Fill-in snippet for spatial fields*	O	1	gco:Decimal	*ignore_empty lat_validator*
Keywords[5][6]	The short term to describe the contents of the dataset.	O	N	gco:CharacterString	*wikidata_keyword*
Books[7]					
ISBN-13		O	1	gco:CharacterString	
ISSN		O	1	gco:CharacterString	
Journal		O	1	gco:CharacterString	

Continued on next page

[1] For details please refer to appendix: *Data Type*.

[2] CKAN has the validator mechanism to check if the given value is valid. CKAN also comes with converters to transform the given value into a valid value.

[3] The URL will be generated automatically when you input the title of the dataset. If there is no letter or number in the title, a random hash will be generated. You can modify the generated URL afterwards.

[4] Please only use this field when there is no proper entry in the Wikidata to describe the dataset. Otherwise, use the `Keywords` field below instead.

Table 1 – continued from previous page

Field Name	Description	Mandatory (M)/Optional (O)	Maximum Occurrence	Data Type	Validators and Converters
Volume		O	1	gco:CharacterString	
Proceeding		O	1	gco:CharacterString	
Location		O	1	gco:CharacterString	
Publisher		O	1	gco:CharacterString	
Publication Year		O	1	gco:CharacterString	
Book Query		O	1	gco:CharacterString	
URL		O	1	gco:CharacterString	
Historical Material		O	N	Hist_material_type	*scheming_multiple_choice*
Village of Research Area		O	1	gco:CharacterString	
Religion of Research Area		O	1	gco:CharacterString	
Family of Research Area		O	1	gco:CharacterString	
Reservoir of Research Area		O	1	gco:CharacterString	
Industry of Research Area		O	1	gco:CharacterString	
Notes		O	1	gco:CharacterString	
Pictures[8]					
Original Source		O	1	gco:CharacterString	
Scan Size	Size (cm) of source (e.g., 60x72)	O	1	gco:CharacterString	
Scanning Resolution	Resolution (DPI) of source (e.g., 300)	O	1	gco:Integer	*ignore_empty* *is_positive_integer*
Spatial Resolution	Spatial resolution (m) of source	O	1	gco:CharacterString	*ignore_empty* *positive_float_validator*
Scale Denominator	Scale denominator of data	O	1	gco:Integer	*ignore_empty* *is_positive_integer*
Preprocessing	Steps of data generating process	O	1	gco:CharacterString	

[5] We use Wikidata entries as the source for keywords. Wikidata entries are multilingual, which means the language of keywords may align with the site language setting.

You can also search and select keywords by an autocomplete dropdown list as shown below:

[6] Use the `Tags` field above when there is no proper entry in the Wikidata to describe the dataset.

[7] The corresponding fields for the `Books` data type (See the "Basic Information" above).

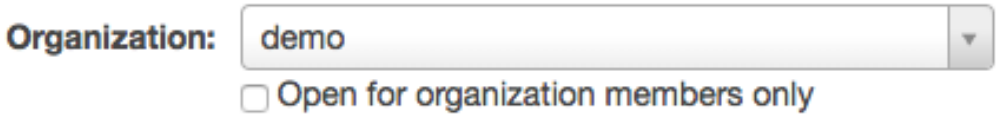
[8] The corresponding fields for the `Pictures` data type (See the "Basic Information" above).

Chianan Irrigation

ID	Label	Description
Q31069	Chianan Irrigation	system of irrigation from Chinian Plain, Taiwan
Q24040887	Chianan Irrigation Drainage Basin, Taiwan	

3.1.3 Management Information

Field Name	Description	Mandatory (M)/Optional (O)	Maximum Occurrence	Data Type	Validators and Converters
License[9]	It is important to include license information so that people know how they can use the data.	M	1	License_code	
Author	The name of the person or project responsible for producing the data.	M	1	gco:CharacterString	
Created Time	The time when the resources in the dataset were created.	O	1	gco:Date	*ignore_empty* *date_validator*
Project[10][11]	If you are a member of any projects, this drop-down will enable you to choose which one should own the dataset.	O	1	gco:CharacterString	*owner_org_validator* *unicode*
Maintainer	If necessary, the name for a second person responsible for the data.	O	1	gco:CharacterString	
Maintainer Email	If necessary, the email for a second person responsible for the data.	O	1	gco:CharacterString	
Maintainer Phone	If necessary, the phone number for a second person responsible for the data.	O	1	gco:CharacterString	
Identifier	The unique identifier of this dataset in its source.	O	1	gco:CharacterString	



3.2 Metadata at the resource level

Field Name	Description	Mandatory (M)/Optional (O)	Maximum Occurrence	Data Type[1]	Validators and Converters[2]
URL	The url of the resource.	O	1	gco:CharacterString	*ignore_missing unicode remove_whitespace*
Name	The title of the resource.	O	1	gco:CharacterString	
Description	You can add a longer description of the resource here.	O	1	gco:CharacterString	
Encoding	The character encoding of the resource (e.g., UTF-8 or Big5).	O	1	Encoding_type	*scheming_required scheming_choices*
Coordinate Systems[3]	The coordinate systems of the spatial resource.	O	1	gco:Integer	*ignore_empty is_positive_integer*
Format[4]	The file format of the resource (e.g., CSV[5], XLS, JSON, or PDF).	O	1	gco:CharacterString	*if_empty_guess_format ignore_missing clean_format unicode*

3.3 Manual for Validators and Converters

CKAN has the validator mechanism to check if the given value is valid. CKAN also comes with converters to transform the given value into a valid value.

[9] If you need to use a license not on the list, please select the "Other Licenses" and mark the license in the `Remarks` field above.

[10] If you select "No project", this dataset will not be owned by any project and will be opened to the public.

[11] If you check the "Open for project members only" box below this field, this dataset will only be seen by members of the project owning the dataset and will not show up in searches by other users. Otherwise, the dataset will be public and can be seen by any user of the site.

[1] For details please refer to appendix: *Data Type*.

[2] CKAN has the validator mechanism to check if the given value is valid. CKAN also comes with converters to transform the given value into a valid value.

[3] The EPSG (European Petroleum Survey Group) system has been used.

[4] The data preview function will check this field to specify a proper resource view. Please refer to *Extended feature — Data preview and visualization*.

[5] Comma-separated values

3.3.1 Internal Validators and Converters

ckanext.data_depositario.converters.**remove_blank_wrap**(*value*, *context*)
Remove blank and text wrap in the value.

3.3.2 External Validators and Converters

if_empty_same_as(name) Return the value in the `name` field if the provided value is empty.

unicode Checks that the provided value (if it is present) is a valid unicode string.

not_empty Only check if the provided value is empty.

package_name_validator Check that no package with the given name already exists and limit the length of the name from 2 characters to 100 characters.

scheming_required If the field is required, apply the `not_empty` validator. Otherwise, apply the `ignore_missing` validator.

scheming_choices Must be empty or one of the field choices values.

ignore_missing By putting `ignore_missing` at the start of the schema list for a field, you can allow users to post a dataset or resource without the field and the dataset or resource will pass validation. But if they post a dataset or resource that does contain the field, then any validators after ignore_missing in the dataset's or resource's schema list will be applied.

tag_string_convert Check that if the tag is a valid unicode string, -, _, or . characters. And limit the length of the tag from 1 characters to 100 characters.

ignore_empty Accept the empty string.

wikidata_keyword Must be in the form of Python list (e.g., `["Q1", "Q2"]`) or string (e.g., `"Q1, Q2"`).

scheming_multiple_choice Must be in the form of Python list (e.g., `["Q1", "Q2"]`) or string (e.g., `"Q1, Q2"`). And the values must be in the field choices values.

is_positive_integer Must be a postive integer.

owner_org_validator Must be "no project" or an existing project.

remove_whitespace Remove the leading and trailing whitespace characters in the string.

if_empty_guess_format Guess file format if it is empty.

clean_format Convert the filename extension to lower case.

3.4 Data Type

gco:CharacterString[1] String (text)

gco:Date Date (ISO8601 standard)

GeoJSON See http://geojson.org/

gco:Decimal Decimal value

gco:Integer Integer value

Data_type See the table below for details

[1] gco stands for Geographic COmmon extensible markup language

Statistics
Books
Pictures (Non spatial)
Pictures (Spatial)
Vector
3D Model
Multimedia

Language_type Based on ISO639 standard

Chinese
Hakka Chinese
Min Nan Chinese
Yue Chinese (Cantonese)
Literary Chinese
Japanese
English
Spanish (Castilian)
Dutch
Siraya
Portuguese

Temp_res_type See the table below for details

Year
Decade
Century
Date
Month

Hist_material_type See the table below for details

Local chronicles in Qing dynasty
Japan officials
Officials in postwar period
Inscriptions
Newspapers
Maps
Taiwanese Governor Office Files
Dutch Formosa
Field Researches

License_code See the table below for details

License Not Specified
Public Domain
CC0 1.0
CC-BY 4.0
CC-BY-SA 4.0
CC-BY-NC-SA 4.0
ODbL 1.0
GFDL
TWOGDL
Other Licenses

Encoding_type Based on IANA Character Sets

See the table below for details

Big5
UTF-8
ISO-8859-1
GB2312
GB18030
Shift_JIS
EUC-JP

CHAPTER 4

Changelog

4.1 v6.3.2 2018-10-25

- Update: UI hotfix.

4.2 v6.3.1 2018-10-25

- Update: Miscellaneous UI improvements.

4.3 v6.3.0 2018-10-23

- Update: Revamped look.

And, registration is open to the public as of today.

4.4 v6.2.1 2018-08-24

- Update: Email confirmation required to create an account.
- Update: Correct some errors in documentation.
- Update: Update licenses to match https://licenses.opendefinition.org/. Add CC-BY-NC-SA 4.0 license.
- Remove: News block in the home page.

4.5 v6.2.0 2018-07-20

- Improvement: Add a “License Details” tool beside all Licenses filters.

- Update: CKAN core version 2.7.4.
- Other improvements and UI adjustments.

4.6 v6.1.3 2018-07-06

- Add: English documentation in footer.
- Improvement: Move the language selector to the top-right corner.
- Improvement: Fix an issue where the `Preprocessing` dataset level field cannot be displayed correctly (#2).
- Improvement: Correct some errors in Chinese documentation.

4.7 v6.1.2 2018-05-10

- Update: CKAN core version 2.6.6.

4.8 v6.1.1 2018-04-23

- Add: Documentation in footer (Chinese only at present).

4.9 v6.1.0 2018-03-23

- Add: Site status in footer.
- Improvement: Fix the wrong positive_float_validator validator.
- Improvement: Apply the suitable validators to schema fields.
- Improvement: Add LineString support to map for filling spatial extent.
- Improvement: Add edit and delete tools to map for filling spatial extent.
- Update: Leaflet.draw 0.4.1.
- Update: CKAN core version 2.6.5.
- Move the Wikidata-powered keyword function to an extension: https://github.com/depositar-io/ckanext-wikidatakeyword.
- Other improvements and UI adjustments.

4.10 v6.0 2017-11-03

- Add: A Keywords field, which integrates wikidata entries, replaces the old theme and spatial keywords.
- Add: System will generate a hash if the new dataset's title can not be slugfied.
- Update: CKAN core version 2.6.4.
- Other improvements and UI adjustments.

4.11 v5.0.x 2017-09-05

- Improvement: Simplified metadata with three categories – basic information, descriptive Information, and management information. Add Remarks to replace Reference and Sub Project. Move Encoding to resource level. Remove some fields which are not often used.
- Improvement: After a user fills in Spatial field using a map, system will generate geojson value and parcel corner and lock those fields.
- Improvement: Maintainer and Maintainer Email can be filled in with logged-in account information.
- Improvement: Add a checkbox to open a dataset for organization members only.
- Improvement: Separate translations for our custom extension from CKAN core thanks to CKAN 2.5's translation capabilities for extensions.
- Update: ckanext-pages verison with zh_TW language.
- Update: CKAN core version 2.6.3.
- Other improvements and UI adjustments.

Python Module Index

C

Index

C

R