ACEP Data Catalog User Guides

Jenae Matson

2024-08-05

Table of contents

Welcome

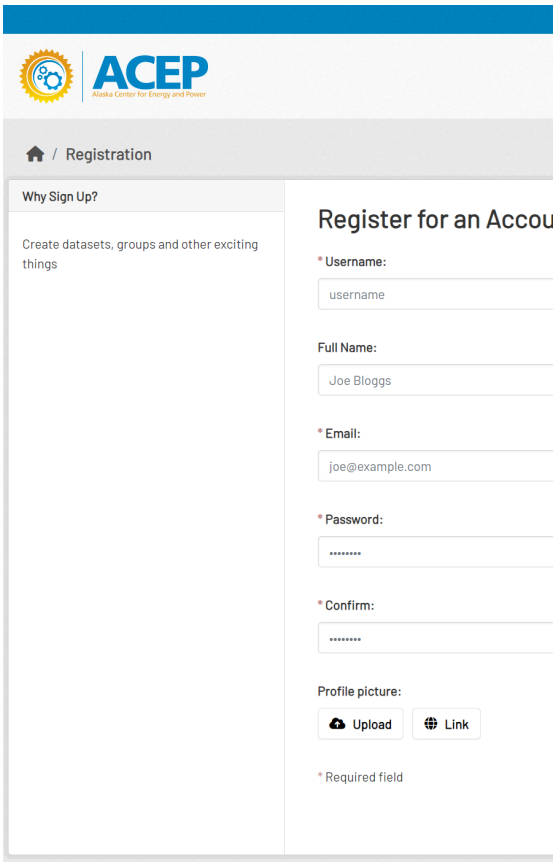
This documentation provides some helpful tutorials for working with the ACEP Data Catalog.

1 Getting Started

Here are some instructions for how to navigate the data catalog.

Making an Account

1. Click the **Register** button in the upper right corner of the screen.

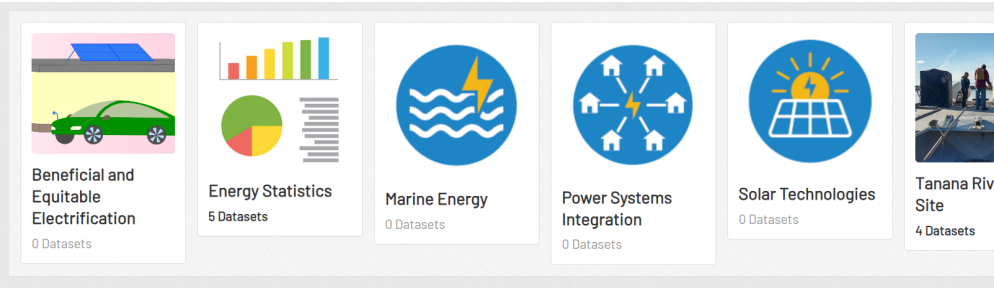


2. Fill out your information, using your UA email if you have one.
3. In order to see the Internal Use datasets or post datasets, you will need permissions granted to your account. If you are an ACEP employee, contact Liz or another admin.

Searching for Datasets

There are multiple ways to search for datasets

- Search Bar
  Use key words to find datasets, just like Google.
- Groups
  Browse through the groups that can be found at the **Groups** tab at the top of the screen



  or listed on the home page.
- Organizations

Submitting a Comment/Complaint

If you find a problem with a dataset or have a comment or suggestion for the data catalog, please contact the ACEP data team at uaf-acep-dcm-support@alaska.edu.

2 Researcher

This is some things researcher's posting stuff should know

Posting a Dataset

There are two ways to access the **Add Dataset** button:

- Click on the **Datasets** tab at the top of the screen
- Click on the **Organizations** tab at the top of the screen and select the organization you want to add a dataset to:
  - Choose **ACEP Open Data** if you are posting your own data or data that ACEP owns.
  - Choose **ACEP Internal Use** if you are posting a useful dataset that ACEP does not own.

Tagging a Dataset

Editing/Deleting a Dataset

Adding a Dataset to a Group

Creating a New Group

3 Admin

Here is some stuff admins should know how to do.

Giving User Permissions

Reviewing a Dataset

Deleting Datasets/Groups/Organizations

Restoring a Deleted User

When a user is deleted from the website, their information remains in the database with the `state` field set to `deleted`. To reactivate the user, you must set this field to `active` in the database.

1. Enter the `acep-db-cont` docker container
    - `> docker exec -it acep-db-cont /bin/bash`

2. Access the postgres database
    - `> psql -U postgres`

3. List all the databases and connect to the ckandb database.
    - `> \l`
    - `> \c ckandb`
4. List all the tables and list the columns of the user table.
    - `> \d`
    - `> \d user`
5. List all the users in the user table.

- `> SELECT * FROM public.user`

6. Find the deleted user with the username [username].
    - `> SELECT id, name, email, state FROM public.user WHERE name = ‘[username]’`
7. Update the user's state field.
    - `> UPDATE public.user SET state = ‘active’ WHERE name = ‘[username]’`
8. Find the user again and ensure that the state field is set to active.
    - `> SELECT id, name, email, state FROM public.user WHERE name = ‘[username]’`

For Developers

Documentation of the development of the ACEP Data Catalog.

For more information and guides, visit the official CKAN Documentation

Developing the Data Catalog

The ACEP Data Catalog is run on a VM hosted by RCS. Extensions can be updated by pushing to the acepportal-ckan Github repository. After pushing, changes take ~30 min to update on the main site.

Creating a Local Instance

Creating a local version of the data catalog is a useful tool for developing and testing new features.

1. Install Docker: https://www.docker.com/get-started/

2. Clone the ACEP CKAN repository from Github: https://github.com/UAF-RCS/acepportal-ckan.git

3. Create a replica of the main VM CKAN instance by copying over the source files, storage files, ckan.ini file, and database backups.
   These files are located on the VM inside `/opt/ckan/backups`. Use scp to copy the files onto your machine

   - `scp user@acepportal.rcs.alaska.edu:/opt/ckan/backups/app_[date].tar.bz2 .`
   - `scp user@acepportal.rcs.alaska.edu:/opt/ckan/backups/app_storage_[date].tar.bz2 .`
   - `scp user@acepportal.rcs.alaska.edu:/opt/ckan/backups/ckandb_[date].tar .`
   - `scp user@acepportal.rcs.alaska.edu:/opt/ckan/backups/datastore_[date].tar .`

4. Use tar to decompress the source and storage tar files
    - `tar -jxvf app_[date].tar.bz2`
    - `tar -jxvf app_[date].tar.bz2`
5. Place the source files and storage files inside of `acepportal-ckan/ckan-src`
6. Create a backups folder alongside the `acepportal-ckan` repository on your machine.
7. Create the `.env` file inside the main `acepportal-ckan` folder. Copy the contents from the `.env.example` file.
8. Specify the location of the source files, storage files, backups, etc. in the `.env` file. For example:

```
# CKAN Mounts Directory
CKAN_EXTENSIONS_MOUNT=./ckan-extension
SRC_EXTENSIONS_PATH=/srv/app/src_extensions
CKAN_SOURCE_MOUNT=./ckan-src/src
CKAN_STORAGE_MOUNT=./ckan-src/storage
CKAN_INI_MOUNT=./ckan-src/ckan.ini
```

9. Build the containers using,
    - `docker compose up`
10. Once the containers are up, use the `import_database.sh` bash script to import the database.
    - `bash import_database.sh`
11. Rebuild the CKAN search index.
    - `docker exec -it acep-ckan-cont /bin/bash`
    - `cd /srv/app`
    - `ckan search-index rebuild`

Create a New Extension

1. Run the following command inside the `acep-ckan-cont` Docker container.

- `ckan generate extension -o /srv/app/src/ckan-extension` This will create an extension in the `ckan-extension` folder which can be edited outside of the container.

2. Add the extension name to the `CKAN_PLUGINS` list in the `.env` file.
3. Run `docker compose up -d --build ckan`

Install an Extension

1. Ensure that the extension supports CKAN 2.10.4 and Python 3.10 Clone the extension repository into the `ckan-extension` folder.
2. Ensure that all dependencies for the extension are listed in `requirements.txt` or a similar file.
3. Add the extension name to the `CKAN_PLUGINS` list in the `.env` file.
4. Run `docker compose up -d --build ckan`

Extensions

Currently Installed

ckanext-customtheme

Author: Jenae Matson
Purpose: Add custom theming and features for the CKAN instance, including

- ACEP logos, colors, and fonts
- Home page layout, images, and featured dataset
- Changed font weight of Register button
- Added tags to search page display
- HTML file for About page text
- Removed social media links from dataset/resources pages
- Added support contact info to dataset sidebar
- Added default blank option to add-to-group dropdown menu

ckanext-faqpage

Author: Jenae Matson
Purpose: Create an FAQ page linked in the masthead with collapsible boxes for questions and answers.

ckanext-restrictpublish

Author: Jenae Matson
Purpose: Restrict the ability to ochange the visibility of a dataset to admins only. Datasets posted by editors default to private.

ckanext-geoview

Link: https://github.com/ckan/ckanext-geoview
Purpose: Created resource views for geojson and other geo-data file types. We have implemented the OpenLayers Viewer.

ckanext-package-group-permissions

Link: https://github.com/salsadigitalauorg/ckanext-package-group-permissions
Purpose: Allows all editors and admins to add datasets to any group, without having to be added as members to each group.
Modifications: This extension was created and works with CKAN 2.9. This instance is version 2.10, so the extension requires some small modifications to work. The following changes were made to the original extension:

- In the file `plugin.py`, change the `member_create` function to the following

```
def member_create(self, next_auth, context, data_dict):
    """
    This code is largely borrowed from /src/ckan/ckan/logic/auth/create.py
    With a modification to allow users to add datasets to any group
    :param context:
    :param data_dict:
    :return:
    """
    group = logic_auth.get_group_object(context, data_dict)

    authorized = False
    if not group.is_organization and data_dict.get('object_type') == 'package':
        authorized = helpers.user_has_admin_access(include_editor_access=True)

    if not authorized:
        # Fallback to the default CKAN behaviour
        return next_auth(context, data_dict)
    else:
        return {'success': True}
```

- In the the file `templates/package/group_list.html`, add the line `{ h.csrf_input() }` to the beginning of the two post forms, as follows

```
{% if groups %}
<form class="add-to-group" method="post">
    {{ h.csrf_input() }}
    ...
</form>
{% endif %}
```

```
{% if c.pkg_dict.groups %}
<form method="post">
    {{ h.csrf_input() }}
    ...
{% endif %}
```

ckanext-scheming

Link: https://github.com/ckan/ckanext-scheming
Purpose: Allows for the creation of alternate metadata templates (schemas) defined by .yaml or .json files.

Adding Alternate Schemas with ckanext-scheming

1. Create a .yaml or .json file in the folder `ckanext-scheming/ckanext/scheming` to define the metadata schema. See extension documentation for more information and examples.

2. In `ckan.ini`, add your schema to the `scheming.dataset_schemas` config option. For example:

```
scheming.dataset_schemas = ckanext.scheming:arctic_dataset.json
                           ckanext.scheming:geo_dataset.json
```

3. The new dataset creation form is located at a url defined by the schema type name. For example, the creation form for datasets of type `arctic-dataset` is located at `/arctic-dataset/new`. You can define a new Add Dataset button using this new url.

Attempted Extensions

ckanext-spatial

Link: https://github.com/ckan/ckanext-spatial
Purpose: This extension adds the ability to search for datasets on a map widget, as well as a dataset extent map widget on the dataset page, provided correct geospatial metadata.
Problems: This extension is not currently installed due to the following,

- Configuring map tiles for ckanext-spatial caused the map tiles for ckanext-geoview to disappear.
- Datasets with the required spatial metadata were not searchable on the map search widget, although the dataset extent widet worked correctly.

ckanext-oidc-pkce

Link: https://github.com/DataShades/ckanext-oidc-pkce/tree/master
Purpose: This extension allows for users to be authenticated through an external application when they login.
Problems: Ideally users on the ACEP Data Catalog would be able to login using their UA login credentials through Google Authentication. This extension installs correctly, but does not seem to support Google Authentication.