DKAN Documentation

Release 1.17

DKAN

Jan 09, 2020

Contents

This is the central site for technical/developer documentation of DKAN. DKAN is a Drupal-based open data portal and catalog. Development sponsored by CivicActions .

CHAPTER 1

DKAN Overview

DKAN is a open data platform with a full suite of cataloging, publishing and visualization features that allows governments, nonprofits and universities to easily publish data to the public. DKAN is maintained by CivicActions.

DKAN is a Drupal-based open data portal based on CKAN, the first widely adopted open source open data portal software. CKAN stands for Comprehensive Knowledge Archive Network. It has inspired at least one other variant - JKAN, which is built on Jekyll.

1.1 About this documentation

What follows is a style guide for the DKAN documentation. Use it both to follow the conventions used throughout the site, and for your own contributions. DKAN's docs are written in a combination of Markdown (specifiically, CommonMark) and ReStructuredText (RST), and built with Sphynx. The docs live in the `/docs` folder of the DKAN Project; to suggest modifications, submit a pull request as you would for any suggested code change.

1.1.1 File types

Index files should always be in RST, to render correctly in the sidebar when built. Additional files can be in markdown or RST format depending on your preference. Currently, most DKAN documentation is in Markdown, mainly for historical reasons.

In some cases, `README.md` files are pulled into the docs site from elsewhere in the repository. This is accomplished with symbolic links in the docs folder.

1.1.2 Images

Screenshots should be taken at standard desktop resolution (no retina!) and avoid showing any browser chrome. If necessary they may contain arrows and annotations in red with sans-serif typeface.

1.1.3 Text conventions

Modules

Module names are written in Title Case with no additional styling. Quotes can be used if needed for clarity – for instance, it might be confusing to talk about how the "Data" module affects data on the site without quote marks. When possible, a module name is linked to its home page (on Drupal.org or Github) on its first mention in a page.

Entities and bundles

A specific content type or other entity bundle is written in italics, as in referring to a *dataset* node or a *choropleth* visualization. Entity types, like "node," require no additional styling.

Files

Filenames are written as inline code as in this example: `thisfile.txt` will do the trick.

Terminal commands

Terminal commands should be expressed in a full code block, with each line starting with$:

```
$ first -i "run" this-command
$ ../then.this --one
```

Code blocks

Code blocks are also expressed as... code blocks:

```
/**
 * Adds declared endpoint to list.
 *
 * This and hook_open_data_schema_map_load() are necessary so that modules can
 * declare more than one endpoint.
 */
function hook_open_data_schema_map_endpoints_alter(&$records) {
  $records[] = 'my_machine_name';
}
```

Code objects

When referring to **`$variables`**, **`function_names()`** and **`classNames`** inline, use bold inline code style. This can be achieved in markdown like this:

```
**`This text`** will be code-styled and bold
```

1.1.4 Building this documentation

If you contribute significantly to this documentation, at some point you will want to be able to build them locally to preview your formatting and other markup. This will require some degree of comfort with command-line tools but is otherwise fairly straightforward.

Sphinx

Sphinx <http://www.sphinx-doc.org/en/1.5.1/>_ is the Python application that generates the HTML from the documentation markup.

To work on sphinx documentation locally, install the Sphinx Python tools. This requires having the easy_install tool in your environment.

Install pip (the python package manager):

```
$ sudo easy_install pip
```

Then install sphinx

```
$ sudo pip install sphinx
```

Install the dependencies for this project. Make sure you are in the /docs directory:

```
$ cd docs
$ sudo pip install -r requirements.txt
```

Now you should be able to build the Sphinx site by typing

```
$ make html
```

The site will build in _build/html

Auto-build server

If you install the sphinx-autobuild package with pip, you can run a server that will build automatically when it senses a file change, and refresh your browser. Run

```
$ sudo pip install sphinx-autobuild
```

...then, from the /docs directory, run:

```
$ sphinx-autobuild ./ _build/html
```

The autobuild tool sometimes does not pick up changes to indexes very well. If you see issues with the sidebar table of contents, stop the server, delete the /_build directory and then re-start the server:

```
$ rm -rf _build && sphinx-autobuild ./ _build/html
```

1.2 Catalog Basics

Open data catalogs - or portals - are simple in purpose but may appear complicated at first glance. A homepage will often feature a slideshow, list of blog posts, or a visualization using data from the underlying catalog; these homepage assets can easily obscure the underlying structure of the data.

Understanding the structure of an open data catalog can help users, developers, and stakeholders make the best use of open data tools.

1.2.1 Catalog

The catalog is a wrapper containing information, sorted into datasets. A catalog should provide the answers to basic questions such as: "Who is providing this data?" and "Under what conditions?"

DCAT - an RDF vocabulary designed to facilitate interoperability between data catalogs published on the Web - recommends providing the following fields: catalog record, dataset, description, homepage, language, license, publisher, release date, rights, spatial, themes, title and update date.

1.2.2 Dataset

A dataset contains individual resources as well as metadata. Metadata is the "Who, What, When, Where, Why" of each dataset. There are a number of specifications for dataset metadata and DKAN currently complies with the DCAT and Project Open Data schemas.

It is possible to add new fields to DKAN to conform to additional specifications or custom requirements.

1.2.3 Resource

Resources are the actual files, APIs or links that are being shared. Resource types include csv, html, xls, json, xlsx, doc, docx, rdf, txt, jpg, png, gif, tiff, pdf, odf, ods, odt, tsv, geojson and xml files. If the resource is an API, it can be used as a live source of information for building a site or application.

1.2.4 Datasets and Resources within DKAN and CKAN

The following images display how datasets and resources are structured within DKAN and CKAN. Screenshots are from Data.gov which is powered by CKAN. DKAN follows CKAN conventions where possible.

1.3 Catalog Features

1.3.1 Project Open Data Compliance

DKAN provides a "data.json" index to satisfy the US federal government's Project Open Data requirements. More information about the "slash data" or "data.json" requirements can be found in POD's Open Data Catalog Requirements and Common Core Metadata Schema pages.

The exact mapping of data (specifically, Drupal data tokens) from your DKAN site to the data.json index can be customized using the Open Data Schema Mapper.

1.3.2 DCAT-Compliant Markup

Project Open Data's schema is based on the DCAT open data vocabulary. DKAN also provides RDF endpoints and RDFa markup for all Datasets following the DCAT specification.

1.3.3 Public Catalog Listing API, Based on CKAN

For exposing more and better-structured machine-readable metadata than Project Open Data's data.json allows for, DKAN also ships with a public API based heavily on CKAN's. This includes APIs for viewing the contents of an entire catalog, as well as requesting the metadata for a single dataset.

1.4 Comparing DKAN and CKAN

CKAN is an open data catalog that has powered many high-profile portals, including the main open data portals for both the United Kingdom and the United States, among others. The makers of DKAN have enjoyed contributing to and deploying CKAN. So why DKAN?

Ultimately, DKAN is a complimentary offering to CKAN in the effort to make data more open and accessible.

1.4.1 Drupal and PHP Ecosystems

PHP powers a significant percentage of web pages and Drupal powers an estimated 2% of the Internet as a whole. This percentage is even higher among government entities who choose to publish open data. DKAN offers an easy option for those who have already adopted PHP or Drupal. DKAN can also be enabled in existing Drupal sites so that anyone using Drupal can easily start to publish open data in standards compliant ways.One of the design goals of DKAN is to make it easy for anyone with an inexpensive hosting environment to create an open data catalog. Thanks to the popularity of Drupal there are many resources to help install and host Drupal sites like DKAN.

1.4.2 Integrated Content Management System

CKAN has powerful publishing, auditing, and harvesting features for open datasets. Those using CKAN often choose to pair it with Drupal, Wordpress, Django, or other content management systems (CMS) or web publishing platforms to create pages, blogs and other content.

DKAN takes a different approach by integrating open data catalog features into an existing CMS. Datasets are treated as content that can unlock rich workflows. Drupal also provides a user interface for many site management activities. In turn, teams managing content only need to be trained on one system instead of two.

The fact that DKAN provides a single codebase is another benefit. Again, DKAN is a complementary effort to CKAN in enabling people to publish open data using open source tools.

1.5 Get DKAN

DKAN is open source and flexible: you can download it for free and run it on your own server or choose from one of our hosting partners below. Visit getdkan.org for more information.

1.5.1 Download and run DKAN on your server

DKAN is based on the open source Drupal content and application framework and runs almost anywhere Drupal is supported. Users unfamiliar with Drupal may be more comfortable trying one of the hosted options listed below, or contacting us to obtain a private demonstration instance. There is extensive information on how to install DKAN on your own in the installation section of this site.

1.5.2 Hosting Partners

Pantheon

Click here to install DKAN on Pantheon for free.

Pantheon provides reliable Drupal cloud hosting with a powerful development tools and web-based user interface designed to facilitate and encourage best development practices. With essentially a single click, you can spin up a new DKAN instance on Pantheon and log in to your new site in just a few minutes.

You can register for a free account here. Once you have access, create a new dkan site:

Pantheon will then build your new based site on the latest DKAN release. You will go through a normal Drupal install process, explained in detail in the installation instructions.

Using Terminus

Pantheon provides a command-line tool called Terminus for interacting with all aspects of site management on their platform. Once you have installed Terminus, you can spin up a new instance of DKAN with the command:

```
$ terminus site:create dkan-example-site "DKAN Example Site" d7370d7e-46fb-4b10-b79f-
↪942b5abf51de
```

Replace "DKAN Example Site" with the name of your new DKAN site. The last argument, `d7370d7e-46fb-4b10-b79f-942b5abf51de`, is Pantheon's internal ID for the DKAN upstream. After the command completes, you will see your new site on your dashboard.

Managing updates

Pantheon uses a modified version of Drupal Pressflow, which is publicly available on GitHub. Whenever a new version of the DKAN distribution is released, the changes are merged into a version of DKAN special-built for Pantheon, also available on GitHub.

However, Pantheon provides an easy way to update your instance of DKAN (or any Drupal distribution hosted with them). Each time the DKAN's Pantheon build is updated, an alert will appear in your Pantheon dashboard:

Usually, you will be able to use the "Apply Updates" button to merge those "upstream" changes directly into your copy of the codebase, alongside any changes you have already made to it. If you are developing locally using git, the next time you pull from your Pantheon repository, you'll receive the DKAN updates locally as well.

If you have modified any of the files included with DKAN, merging in upstream changes may produce conflicts. Pantheon's dashboard provides instructions for how to do the merge locally, to give you more control over resolving potential conflicts.

Troubleshooting

An error like the following is often seen at the end of the install process on Pantheon:

This will hopefully be fixed on future releases. However, the resulting site should still be fully installed and functional.

Acquia

Click here to install DKAN on Acquia for free.

Acquia offers a number of hosting tools built specifically for best maintaining Drupal sites. These include integrations with 3rd party systems like New Relic and Blaze Meter as well as reports on module updates, performance, and security reviews. Most importantly, Acquia offers a dashboard that makes it easy to move code (hosted by git), media files, and the database between development, testing, and production environments:

These tools allow a single site builder or team of developers to follow best practices, scale up if needed, and follow a rigorous QA process all without ever touching a server.

Single-click Installation

Acquia offers a "single-click" installation of DKAN. While this is labelled as a "Test drive," the environment offers the same dashboard tools as a full, paid account. Visit Acquia's website for complete instructions.

Maintaining a DKAN Site on Acquia

Updates to DKAN are released frequently. Acquia will not push these updates to your instance automatically, but you can keep your codebase up-to-date using your own workflow, or following our general Upgrade Instructions.

1.6 Updating and Maintaining DKAN

There are several strategies for maintaining your DKAN site. Maintaining a DKAN site does not differ substantially from maintaining other Drupal distributions.

Drupal distributions consist of a script that runs at the time of the installation as well as a set of modules, themes, and libraries that exist at `profiles/MY_DISTRIBUTION` directory. These modules, themes, and libraries work the same as any other modules, themes, or libraries that are added to Drupal sites. They are packaged together in the `profiles` directory to make it easier to install and maintain.

Tip: DKAN Starter is project containing a prebuilt version of DKAN and the tools CivicActions uses for our own implementations and deployments. Learn more advanced workflows in that project's documentation.

1.6.1 Filesystem Conventions

With Drupal's inheritance model mentioned above, it should not be necessary to place custom code or modules in the `profiles/dkan` directory. Additional modules, themes, or libraries, or newer versions of ones already present in `profiles/dkan`, can be placed in `sites/all`.

If it is necessary or expedient to overwrite files in the `profiles/dkan` directory, it is recommended to keep a patch of the changes. A patch will make it possible to re-apply changes once a newer version of DKAN is added to the 'profiles/dkan' directory.

If DKAN's extensions and customizations of core Drupal are isolated in `profiles/DKAN`, and your site's particular configuration, files, and overrides and customizations of DKAN are isolated in `sites/`, maintaining your DKAN site will be much easier.

1.6.2 Primary Maintenance Tasks

By "maintenance" we mean three specific tasks

- **Upgrading** DKAN to receive new features and bug-fixes

- **Adding** additional modules or features
- **Overriding** current modules or functionally

1.6.3 Getting DKAN Updates

DKAN uses Drupal versioning standards, with one modification. *Minor* upgrades to DKAN are released approximately every 4-6 weeks. For instance, a minor release would move from DKAN 7.x-1.11 to 7.x-1.12. Starting with version 7.x-1.12, we are adding *patch* releases for security and bug fixes. For instance, the first patch release between 7.x-1.12 and 7.x-1.13 will be 7.x-1.12.1.

Please note *you can not use* `drush up` *with DKAN*. This is because DKAN is not packaged on Drupal.org.

Basic Upgrades

The least complex way to update your DKAN codebase is similar to an update of Drupal itself.

1. Back up your database (just in case!)
2. Copy your `sites` folder somewhere safe.
3. Replace your entire codebase with the latest fully built version of DKAN from DKAN DROPS-7.
4. Check the new versions' release notes to see if there are any special instructions for updating. (If you are several releases behind, you may need to follow instructions for several releases).
5. Replace the `sites` folder in your new codebase with your old `sites` folder.
6. Now navigate to *http://yoursite.com/update.php* or run `drush updatedb`.
7. Clear caches by visiting */admin/performance* or running `drush cache-clear all`.
8. Revert all features by visiting */admin/structure/features* or running `drush features-revert-all` (Use with caution, as this may overwrite any DKAN configuration you have overridden and not exported to code; see Features for more information.)

Note: Occasionally a DKAN component will be moved to a new directory. This should be explained in the release notes for that version. But if you get errors related to incorrect location of module files, you may want to try rebuilding the registry.

Using drush make

We are developing an easier workflow to update DKAN on the command line. For the time being, the recommended method for updating using the `drush make` instructions described in the Installation Instructions is similar to the process described above.

Assuming you have followed the instructions for `drush make` and have a `webroot` folder inside a main clone of the DKAN repo:

1. Back up your database
2. Copy your `sites` folder somewhere safe.
3. Remove your webroot folder: `rm -rf webroot` (use with caution!)
4. Check out the new version of DKAN you want to update to: `git checkout tags/7.x-1.12`
5. `drush make drupal-org-core.make webroot --yes`
6. `rsync -av . webroot/profiles/dkan --exclude webroot`

7. `drush make --no-core --contrib-destination=./ drupal-org.make webroot/profiles/dkan --no-recursion --yes`
8. Replace the `sites` folder in your new codebase with your old `sites` folder.
9. Check the new versions' release notes to see if there are any special instructions for updating. (If you are several releases behind, you may need to follow instructions for several releases).
10. `drush updatedb`.
11. `drush cache-clear all`
12. `drush features-revert-all` (use with caution).

You can also use this method to upgrade to the most recent "bleeding-edge" development version of DKAN. Instead of checking out a specific tag, check out the `7.x-1.x` branch in step 3.

1.6.4 Features Module

DKAN packages much of its configuration using the Features module.

After DKAN is upgraded DKAN site maintainers may wish to revert some features in order to take advantage of new functionality. We recommend using the Features Override module to capture overridden features elements to make it easier to revert Features from DKAN when desired. More documentation on this to come.

1.6.5 Advanced Workflows

Using a Custom Make file

DKAN is "built" using a make file and `drush make`. The drupal-org.make file in DKAN contains a list of most of the modules installed in DKAN.

When developing a website for production, it is recommended to keep a make file for all custom modules added to DKAN. Instead of using `drush pm-download` or other means of downloading and adding modules to `sites/all`, a make file is kept that has a list of the sites modules. This enforces some best practices about not overwriting contributed modules, maintaining patches, and reusability. This make file along with DKAN's makefiles also provide a reusable recipe for your site.

More documentation and automation scripts regarding this process are under active development and can be viewed here: DKAN Starter Documentation.

Adding additional modules or features

New modules, themes, or libraries should be added to the 'sites/all' directory. For modules or themes it is often useful to differentiate "custom" modules from "community" modules. We often have a directory structure for modules like:

Location	Contents
`sites/all/modules/contrib`	community or contributed modules
`sites/all/modules/custom`	custom modules
`sites/all/libraries`	Additional libraries

Overriding current DKAN modules or functionality

Drupal has an inheritance model that makes it easy to override modules added to distributions as well as the functionality of other modules.

Any modules or themes added to `sites/all` will override the same named module as one that is placed in `profiles/dkan/`.

If a DKAN site maintainer wishes to update a module supplied by DKAN that module can be placed in "sites/all". For example if one wished to update the Date module, if there is a security update or new release with a certain functionality, add it to `sites/all`:

Location	Version
`profiles/dkan/modules/contrib/date`	7.x-1.4
`sites/all/modules/contrib/date`	7.x-1.5

In this case, DKAN will use the version 7.x-1.5 and ignore 7.x-1.4.

If, later, you update your site to a version of DKAN that uses Date v. 7.x-1.5, the version in `sites/all` should be removed. Be careful to review your overrides in `sites/all` after every DKAN update to ensure you are not missing important module updates.

Note that moving to a different location for an existing, installed module will require a Registry Rebuild to prompt Drupal to refresh all module paths.

1.7 DKAN sites

A partial list of DKAN sites around the world.

Submit an issue to add sites not on this list.

1.7.1 Multinational

Organization	URL
International Organization of Migration (CTDC)	https://www.ctdatacollaborative.org/
The Asian Development Bank	https://data.adb.org/
United Nations (Open Data System Inventory)	http://data.un.org/
The World Bank	http://climatesmartplanning.org
Votes Without Violence	http://www.voteswithoutviolence.org

1.7.2 U.S.-based

Organization	URL
U.S. Department of Health and Human Services (USA)	https://healthdata.gov
National Democratic Institute	https://nditech.org/project/dkan
USDA National Agricultural Library	https://data.nal.usda.gov
California	https://data.ca.gov
Oklahoma	https://data.ok.gov
North Dakota	https://gishubdata.nd.gov
Georgia (Governor's Office of Student Achievement)	https://schoolgrades.georgia.gov
Massachusetts	https://docs.digital.mass.gov/
Nebraska	https://nebraskamap.gov/
Louisville, KY	https://data.louisvilleky.gov
OpenOakland	http://data.openoakland.org
DigitalC	http://civicinsights.org
Department of Conservation, Missouri	https://research.mdc.mo.gov/
Open Puerto Rico	http://abrepr.org

1.7.3 Africa

Organization	URL
Pan-Africa	http://transformagriculture.org
Sierra Leone	http://opendata.gov.sl
Nigeria	http://dataportal.visualdata.com.ng
Ghana	http://data.gov.gh
Namibia	http://www.namopendata.com/demo/
South Africa	http://data.gov.za
Local Development Research Initiative, Kenya	http://transformagriculture.org
Sayada, Tunisia	http://opendata.sayada.tn/fr
Edo, Nigeria	http://data.edostate.gov.ng/

1.7.4 Asia

Organization	URL
Saudi Arabia	http://data.gov.sa/
Bangladesh	http://data.gov.bd/
G0V, Taiwan	http://data.g0v.tw/
Ikoma, Japan	http://data.code4ikoma.org/
Kyoto, Japan	https://data.city.kyoto.lg.jp/
Urban Data Challenge, Japan	http://udct-data.aigid.jp
Banda Aceh City Council, Indonesia	http://data.bandaacehkota.go.id/
Gerakan Anti Corruption Project, Indonesia	http://data.gerakaceh.id/
Hutanriau, Indonesia	http://Hutanriau.org
Telangana Open Data, India	http://data.telangana.gov.in/

1.7.5 Canada

Organization	URL
Open Data Windsor Essex, Canada	http://odwe.ca/

1.7.6 The Caribbean

Organization	URL
Jamaica	http://data.gov.jm
Saint Lucia	http://data.govt.lc

1.7.7 Central and South America

Organization	URL
El Salvador	http://datoselsalvador.org
Buenos Aires, Argentina	http://dkan.puertasdebuenosaires.com
Rosario, Argentina	http://datos.rosario.gob.ar
IT PAMI, Argentina	https://it.pami.org.ar/
Bogotá Secretariat of Culture, Recreation, and Sport, Colombia	http://sispru.scrd.gov.co/siscred/dataset
Narino, Colombia	http://datos.narino.gov.co/
IMSS, Mexico	http://datos.imss.gob.mx/
CONCYTEC, Peru	http://datos.concytec.gob.pe/
Ministry of Education, Peru	http://datos.minedu.gob.pe/
Ministry of Health, Peru	http://datos.minsa.gob.pe/
National Jury of Elections, Peru	http://jnedatosabiertos.pe/
Ministry of Labor, Peru	http://datos.trabajo.gob.pe/
São Paulo State's Department of Education, Brazil	https://dados.educacao.sp.gov.br/
La Paz, Bolivia	https://datosabiertos.cedla.org/

1.7.8 Europe

Organization	URL
Tirana, Albania	https://opendata.tirana.al/
Cambridgeshire, UK	http://opendata.cambridgeshireinsight.org.uk
Marine Scotland, UK	http://marinedata.scotland.gov.uk
Detail Data Northern Ireland, UK	http://data.nicva.org
UK Financial Conduct Authority	Internal site - no public access
Bank of England	Internal site - no public access
INPTDAT (Leibniz Institute for Plasma Science and Technology), Germany	https://www.inptdat.de
Bonn, Germany	https://opendata.bonn.de
Cologne, Germany	https://offenedaten-koeln.de
Mülheim an der Ruhr, Germany	https://geo.muelheim-ruhr.de
Bielefeld University, Germany	https://dszbo-portal.uni-bielefeld.de
Wuppertal, Germany	https://offenedaten-wuppertal.de
RNV, Germany	http://opendata.rnv-online.de
KDZV Frechen Rhein Erft Rur, Germany	http://offenedaten.kdvz-frechen.de
Montpellier, France	http://data.montpellier3m.fr
Italy	http://dati.gov.it
Genova, Italy	http://dati.comune.genova.it
Pompeii, Italy	http://open.pompeiisites.org
Formia, Italy	http://dati.comune.formia.lt.it
Torino, Italy	http://aperto.comune.torino.it
Confiscati Bene, Italy	http://www.confiscatibene.it
Donneche Contano, Italy	http://donnechecontano.it
Data Hub, Hungary	http://data-hub.hu
Ministry of Finance, Czech Republic	http://data.mfcr.cz
Ministry of Defence, Czech Republic	http://data.army.cz
Ministry of Regional Development, Czech Republic	http://data.mmr.cz
Czech Telecommunications Office, Czech Republic	http://data.ctu.cz
Děčín Municipality, Czech Republic	http://data.mmdecin.cz
Opava Municipality, Czech Republic	http://kod.opava-city.cz

Continued on next page

Table 1.1 – continued from previous page

Organization	URL
Decentralised Administration of Crete, Greece	http://apdkritis.gov.gr/en
Bosnia and Herzegovina	http://opendata.ba
Belarus	http://opendata.by
Russia	http://data.gov.ru
Moscow Region, Russia	http://data.mosreg.ru
Instituto Politécnico de Bragança, Portugal	http://observatoriottm.ipb.pt
Barcelona Provicial Diputation, Spain	https://dadesobertes.diba.cat/
Castilla–La Mancha, Spain	https://datosabiertos.castillalamancha.es/
Düsseldorf, Germany	https://opendata.duesseldorf.de/
The Republic of Cyprus	http://www.data.gov.cy/

1.7.9 Oceania / Pacific Islands

Organization	URL
American Samoa	https://americansamoa-data.sprep.org
Transport for New South Wales, Australia	https://opendata.transport.nsw.gov.au/
Maps and Data, Cape York, Australia	http://maps.capeyorknrm.com.au
Cook Islands	https://cookislands-data.sprep.org
Federated States of Micronesia	https://fsm-data.sprep.org
Fiji	https://fiji-data.sprep.org
Kiribati	https://kiribati-data.sprep.org
Republic of the Marshall Islands	https://rmi-data.sprep.org
Nauru	https://nauru-data.sprep.org
New Caledonia	https://newcaledonia-data.sprep.org
Niue	https://niue-data.sprep.org
Palau	https://palau-data.sprep.org
Papua New Guinea	https://png-data.sprep.org
Samoa	https://samoa-data.sprep.org
Solomon Islands	https://solomonislands-data.sprep.org
Tonga	https://tonga-data.sprep.org
Tuvalu	https://tuvalu-data.sprep.org
Vanuatu	https://vanuatu-data.sprep.org
Pacific Environment Data Portal	https://pacific-data.sprep.org

CHAPTER 2

Installation

This document contains instructions for installing the DKAN open data publishing software on your webserver. If you're not comfortable installing and maintaining server software, you may wish to *use web-based tools to deploy to Pantheon* instead.

Please note that we are in the process of revamping our installation and upgrade guide. The instructions here will work, but please bear with us as we develop better documentation and processes.

2.1 Installation Basics

Note: This page is essentially legacy documentation, but still provides some useful information for alternative methods of downloading and working with DKAN. For a more robust local development for working on DKAN core, see *Local Development Environment*. For the most up-to-date documentation on managing website projects with DKAN, see DKAN Starter documentation.

Before getting started, it's recommended that you familiarize yourself with:

- Drush, the command line tool
- Drupal's installation process
- Drupal's upgrade process
- Drupal profiles and distributions

What you will find in the main DKAN Repository is a Drupal *installation profile*. To set up a working website using DKAN, you will need to acquire or build a full DKAN distribution of Drupal.

Tip: DKAN Tools contains scripts that CivicActions uses for our own implementations and deployments.

2.1.1 Requirements

Operating Environment

DKAN is based on Drupal software and – generally – runs anywhere Drupal is supported. This document assumes installation on a Linux-based Apache webserver using MySQL as a back-end database (aka LAMP server). For other environments, please see our Alternative Environment Support.

- MySQL: minimum version 5.0.15+ with PDO
- before installation, please create one MySQL database and associated user.
- PHP: minimum version 5.3.x
- Apache: minimum version 2.x
- Git

Hardware

DKAN has been successfully tested in limited-resource environments, such as Amazon's "micro" AWS instance, for development.

- Minimum RAM: 1GB for development, 2GB or more recommended for production.
- Minimum Disk: 64M for base installation, recommended 1GB or more for production.

DKAN is based on Drupal and follows the same basic installation procedure as any Drupal distribution. More information about various requirements can be located in the Drupal Installation Guide.

2.1.2 Pre Installation

Using fully made version

At the moment, our supported fully-made DKAN codebase is the DKAN DROPS-7 repository, which is optimized to run on the Pantheon platform. You can build a DKAN site with a single click on Pantheon here. (We also offer one-click installation on Acquia)

Download and unzip the latest version of the "DKAN DROPS" codebase on your server webroot.

if you want to do this with git instead:

```
$ git clone --branch master https://github.com/GetDKAN/dkan-drops-7.git dkan
```

Build your own

This "builds" a full DKAN website codebase from the bleeding-edge development version of DKAN, by downloading Drupal and all the additional modules that DKAN needs to run. You may want to use this method to get recent changes that have not yet been included in an official release, or to use a branch or forked version of the DKAN profile.

Note that `rsync` is used to copy the DKAN profile inside the Drupal `/profiles` folder. You may wish to modify this process to fit your own development practices.

Requires drush version 8.x.

```
$ git clone --branch 7.x-1.x https://github.com/GetDKAN/dkan.git
$ cd dkan
$ drush make --prepare-install drupal-org-core.make webroot --yes
$ rsync -av . webroot/profiles/dkan --exclude webroot
$ drush -y make --no-core --contrib-destination=./ drupal-org.make webroot/profiles/
↪dkan --no-recursion
$ cd webroot
```

You can also build from a specific release of DKAN by checkout out the correct tag after cloning. For instance:

```
$ git clone --branch 7.x-1.x https://github.com/GetDKAN/dkan.git
$ git checkout tags/7.x-1.17
...
```

The automated software builder will download and configure the latest version of DKAN and prepare it for installation. When complete, proceed to "Installing the DKAN Software" section below.

Note before proceeding: Recline previews require clean URLs

2.1.3 Installation

Once you've downloaded the DKAN software, it's time to install it. If you've previously installed Drupal, this process will be very similar.

With drush

```
$ drush site-install dkan --db-url="mysql://DBUSER:DBPASS@localhost/DBNAME"
```

You can add the `--verbose` switch if you want to see every step. The installation should end with `drush` creating an admin account with a random password, which will be output in a message to the terminal.

With the web installer

- Open a web browser and visit **http://YOURDKANSITE/install.php**:

Fig. 2.1: Installation Screen

- The first installation screen is a language selection menu. Although DKAN does provide limited multi-language support, installation must currently be performed in English. Continue.
- At this point, your server resources and capabilities are checked to ensure they meet DKAN installation requirements. All errors must be corrected before installation can proceed. Instructions for correcting each error condition are provided.

Fig. 2.2: Installation Screen

- Once your server meets all installation requirements, you'll be presented with the database configuration screen. Enter your MySQL database name, database username, and database password, then click "Save to Continue" to proceed.

Fig. 2.3: Progress Bar

- The installation will proceed, displaying a progress bar on the screen. Depending on your server resources, this may take several minutes.

Fig. 2.4: Configuration

- When installation is complete, the site configuration screen will be displayed. Follow the prompts to set your administrative username, email address, site name, time zone, and other default settings as shown. If the final configuration completes without error, you'll see a short congratulatory message and you'll be prompted to access your new site.

2.2 Local Development Environment

For testing out DKAN locally and doing feature work directly on the software (as opposed to working on a particular, customized website), using a standardized, **docker**-based local environment is recommended. This will ensure you have the same setup as DKAN's core developers, and that your environment is very close to that of our continuous integration tools.

These instructions are geared toward people who want to contribute improvements or fixes to DKAN core. Once you have a working local copy, please make contributions using the standard fork and pull-request workflow in Gitihub.

2.2.1 DKAN Tools

This CLI application provides tools for implementing and developing DKAN, the Drupal-based open data portal.

Requirements

DKAN Tools was designed with a **Docker**-based local development environment in mind. Current requirements are simply:

- Bash-like shell that can execute .sh files (Linux or OS X terminals should all work)
- Docker
- Docker Compose

That's it! All other dependencies are included in the Docker containers that dkan-tools will create.

It is also possible to run most DKAN Tools commands using a local webserver, database and PHP setup, but this practice is less supported.

2.2.2 Installation

1. Download or clone this repository into any location on your development machine.
2. Add *bin/dktl* to your *$PATH* somehow. This is often accomplished by adding a symbolic link to a folder already in your path, like *~/bin*. For instance, if DKAN Tools is located in */myworkspace*:

```
ln -s  /myworkspace/dkan-tools/bin/dktl ~/bin/dktl
```

Alternatively, you could add */myworkspace/dkan-tools/bin* directly to your *$PATH*. Enter this in your terminal or add it to your session permanently by adding a line in *.bashrc* or *.bash_profile*:

```
export PATH=$PATH:/myworkspace/dkan-tools/bin
```

Once you are working in a valid project folder (see next section) you can type *dktl* at any time to see a list of available commands.

2.2.3 Starting a project

To start a project with *dktl*, create a project directory.

```
mkdir my_project && cd my_project
```

Inside the project directory, initialize your project.

```
dktl init
```

This will automatically start up the Docker containers, which can also be started manually with *dktl docker:compose up -d*. Any other docker-compose commands can be run via *dktl docker:compose <args>* or simply *dktl dc <args>*.

After initialization, we want to get DKAN ready. We can use *git clone* (recommended if you are working directly on DKAN core and will want to commit and push changes to the DKAN project) or download a tarball of the DKAN source from GitHub, but the easiest method is using this command:

```
dktl dkan:get <version_number>
```

Versions of DKAN look like this: `7.x-1.15.3`. We can see all of DKAN's releases in Github.

Now run the "make" command:

```
dktl make
```

The *make* command will get all of DKAN's dependencies *including* Drupal core. It will also create all the symlinks necesarry to create a working Drupal site under */docroot*.

Finally, let's install DKAN.

```
dktl install
```

You can find the local site URL by typing `dktl docker:surl`.

2.2.4 Structure of a DKAN-Tools-based project

One of the many reasons for using DKTL is to create a clear separation between the code specific to a particular DKAN site (i.e. "custom code") and the dependencies we pull in from other sources (primarily, DKAN core and Drupal core).

To accomplish this, DKAN Tools projects will have the following basic directory structure, created when we run `dktl init`.

```
- dkan              # The upstream DKAN core codebase
- docroot           # Drupal core, and contrib modules not from DKAN
- src               # Site-specific configuration, code and files
|   - make          # Overrides for DKAN and Drupal makefiles
|   - modules       # Symlinked to docroot/sites/all/modules/custom
|   - script        # Deployment script and other misc utilities
|   - site          # Symlinked to docroot/sites/default
|   |   - files     # The main site files
```

```
|   - test          # Custom tests
- dktl.yml          # DKAN Tools configuration
```

We may wish to create two additional folders in the root of your project later on: */src/patches*, where we can store local patches to be applied via the make files in */src/make*; and */backups*, where database dumps can be stored. The first time we run *dktl install* the */backups* folder will be created if it does not already exist.

The /src/make folder

DKAN uses Drush Make to define its dependencies. DKAN Tools also uses Drush Make to apply overrides patches to DKAN in a managed way, without having to hack either the Drupal or DKAN core.

DKAN defines its Drupal Core dependency in */dkan/drupal-org-core.make*. Additional DKAN dependencies and patches are defined in */dkan/drupal-org.make*. These two files should not be changed directly within the *dkan* folder, but they can be *overridden* via two files in your project: */src/make/drupal.make* and */src/make/dkan.make*.

If we want to override the version of Drupal being used (for instance, if we need a security update just released in Drupal core but aren't ready to move to the newest DKAN version), we add the right version to */src/make/drupal.make*:

```
api: 2
core: 7.x
projects:
  drupal:
    type: core
    version: '7.50'
```

In */src/make/drupal.make* we can also define the contributed modules, themes, and libraries that our site uses. For example if our site uses the Deploy module we can add this to */src/make/drupal.make* under the `projects` section:

```
projects:
  deploy:
    version: '3.1'
```

If our site requires a custom patch to the deploy module, we add it to */src/patches*. For remote patches (usually from Drupal.org) we just need the url to the patch:

```
projects:
  deploy:
    version: '3.1'
    patch:
      1: '../patches/custom_patch.patch'
      3005415: 'https://www.drupal.org/files/issues/2018-10-09/use_plain_text_format-
↪3005415.patch'
```

The src/site folder

Most configuration in Drupal sites is placed in the */sites/default* directory.

The */src/site* folder will replace */docroot/sites/default* once Drupal is installed. */src/site* should then contain all of the configuration that will be in */docroot/sites/default*.

DKTL should have already provided some things in */src/site*: *settings.php* contains some generalized code that is meant to load any other setting files present, as long as they follow the *settings.*<something>*.php* pattern. All of the special settings that you previously had in *settings.php* or other drupal configuration files should live in *settings.custom.php* or a similarly-named file in */src/site*.

The src/test folder (custom tests)

DKAN Tools supports custom PHPUnit and Behat tests found in the *src/test* directory.

If your site does not have tests set up yet, running *dktl init:custom-tests* will set up a *src/test* directory in your project with sample phpunit and behat tests to start from.

To run custom tests:

```
dktl test:phpunit-custom
```

and

```
dktl test:behat-custom
```

To manually configure custom phpunit tests (without using `dktl init:custom-tests`):

1. Create *src/test/phpunit*
2. Place a *phpunit.xml* configuration file in *src/test/phpunit*. You can use the *phpunit.xml* file in dkan/test/phpunit as an example, replacing the <*testsuite*> elements to reflect your own custom tests. See the PHPUnit documentation for more information.
3. Add a copy of dkan/test/phpunit/boot.php in the same directory.
4. Store your tests in *src/test/phpunit*. Again, use *dkan/test/phpunit* as a guide.

JUnit style test results will be written to *src/test/assets/junit*.

To manually configure Behat tests:

1. Create *src/test/features* directory.
2. Place Behat configuration files *behat.yml* and *behat.docker.yml* in *src/test*. You can use the corresponding files in dkan/test as references, or even just create symbolic links to them.
3. Store you tests in *src/test/features*.

JUnit style test results will be written to *src/test/assets/junit*.

2.2.5 Restoring a database dump or site files

DKAN Tools' `restore` commands can restore from a local or remote dump of the database, as well as restore a files archive. This simplest way to do this is:

```
dktl dkan:restore --db_url=<path_to_db> --files_url=<path_to_files>
```

As described below, these options can be stored in a configuration file so that you can type simply `dktl restore`.

You may also restore from a local database backup, as long as it is placed in a folder under the project root called */backups*. Type `dktl db:restore` with no argument, and the backup in */backups* will be restored if there is only one, or you will be allowed to select from a list if there are several.

2.2.6 Create and grab a database dump excluding tables

You can create a database dump excluding tables related to cache, devel, webform submissions and DKAN datastore. Running the command `dktl site:grab-database @alias` will create the database backup for the drush alias passed as argument.

This command needs to be run with DKTL_MODE set to "HOST". So you'll need to run `export DKTL_MODE="HOST"` and after the command finishes, you should set it back to its old value or just unset the variable by running `unset DKTL_MODE`.

If you want to import this dump into your local development site, then you can move the file *excluded_tables.sql* into the directory *backups* in the root of your project, then you'll be able to import it by running `dktl restore:db excluded_tables.sql`.

2.2.7 Configuring DKTL commands

You will probably want to set up some default arguments for certain commands, especially the urls for the `restore` command. This is what the dkan.yml file is for. You can provide options for any DKTL command in dkan.yml. For instance:

```
command:
  restore:
    options:
      db_url: "s3://my-backups-bucket/my-db.sql.gz"
      files_url: "s3://my-backups-bucket/my-files.tar.gz"
```

If you include this in your dktl.yml file, typing `dktl restore` without any arguments will load these two options.

2.2.8 Custom Commands

Projects to can define their own commands. To create a custom command, create a new class inside of this project with a similar structure to the this one:

```
<?php
namespace DkanTools\Custom;

/**
* This is project's console commands configuration for Robo task runner.
*
* @see http://robo.li/
*/
class CustomCommands extends \Robo\Tasks
{
    /**
    * Sample.
    */
    public function customSample()
    {
        $this->io()->comment("Hello World!!!");
    }
}
```

The critical parts of the example are: 1. The namespace 2. The extension of RoboTasks 3. The name of the file for the class should match the class name. In this case the file name should be CustomCommands.php

Everything else (class names, function names) is flexible, and each public function inside of the class will show up as an available `dktl` command.

2.2.9 Advanced configuration

Disabling `chown`

DKTL, by default, performs most of its tasks inside of a docker container. The result is that any files created by scripts running inside the container will appear to be owned by "root" on the host machine, which often leads to permission issues when trying to use these files. To avoid this DKTL attempts to give ownership of all project files to the user running DKTL when it detects that files have changed, using the *chown* command via *sudo*. In some circumstances, such as environments where `sudo` is not available, you may not want this behavior. This can be controlled by setting a true/false environment variable, `DKTL_CHOWN`.

To disable the `chown` behavior, create the environment variable with this command:

```
export DKTL_CHOWN="FALSE"
```

Running without Docker

If for some reason you would like to use some of DKTL without docker, there is a mechanism to accomplish this.

First of all, make sure that you have all of the software DKTL needs:

1. PHP
2. Composer
3. Drush

The mode in which DKTL runs is controlled by an environment variable: `DKTL_MODE`. To run DKLT without docker set the environment variable to `HOST`:

```
export DKTL_MODE="HOST"
```

To go back to running in docker mode, set the variable to *DOCKER* (or just delete it).

2.2.10 Using Xdebug

When using the standard docker-compose environment, Xdebug can be enabled on both the web and CLI containers as needed. Running it creates a significant performance hit, so it is disabled by default. To enable, simply run *dktl xdebug:start*. A new file will be added to */src/docker/etc/php*, and the corresponding containers will restart. In most situations, this file should be excluded from version control with .gitignore.

2.2.11 A note to users of DKAN Starter

Users of DKAN Starter will recognize some concepts here. The release of DKAN Tools eliminates the need for a separate DKAN Starter project, as it provides a workflow to build sites directly from DKAN releases. Support for DKAN Starter and its accompanying Ahoy commands is ending, and detailed instructions for migrating DKAN Starter projects to the DKAN Tools workflow is coming soon.

2.2.12 Troubleshooting

Issue	Solution
PHP Warning: is_file(): Unable to find the wrapper "s3"	Delete the vendor directory in your local dkan-tools and run `dktl` in your project directory
Changing ownership of new files to host user ... chown: ...: illegal group name	Disable the chown behavior `export DKTL_CHOWN="FALSE"`

CHAPTER 3

Major Components

This section contains the documentation for each of the major modules and other components that make up DKAN.

With the exception of the modules described in the last two items in this table of contents (Open Data Schema Map and Visualization Entity), and of the Recline module which is described inside the Datasets section, all this functionality is provided by the modules that ship with the DKAN profile.

3.1 DKAN Dataset Module and Sub-Modules

DKAN's core functionality around datasets, their metadata and resources, is defined in "DKAN Dataset" (*dkan_dataset*) and its submodules (in the *modules* folder):

- **DKAN Dataset Content Types** contains the actual Features exports for the Dataset and Resource content types and fields.
- **DKAN Dataset Rest API** defines a REST endpoint via the Services module, exposing full CRUD operations to authenticated 3rd-party apps and services. See the *Dataset REST API documentation* for more information.
- **DKAN Dataset Groups** provides Organic Groups functionality in DKAN, which groups both dataset content and site users into discreet groups with separate branding and granular access permissions. Usually used to allow for multiple data publishers (for instance, sub-agencies sharing a single data portal).

3.1.1 Usage

Creating Datasets and Resources

DKAN's data publishing model is based on the concept of *datasets* and *resources*. A dataset is a container for one or more resources; a resource is the actual "data" being published, such as a CSV table, a GeoJSON data file, or a TIFF aerial image. The dataset and resource content types in DKAN are provided by the DKAN Dataset module.

In our example, we'll be adding a dataset with Wisconsin polling places to a DKAN site. The data may look familiar; it's one of the sample datasets provided with DKAN upon installation.

Step 1: Create the Dataset

By default, only authenticated ("logged-in") users can add new Datasets and Resources to a DKAN website.The default DKAN user permissions allows Site managers, Editors, and Content Creators access to the administration menu. From here a user may navigate to the Content » Add Content » Dataset link to access the "Create Dataset" form.

The Dataset is the container for the actual data resource files and contains basic information about the data, such as title, description, category tags, and license. Once we've entered information about the data, we can click the "Next: Add data" button to begin adding data.

Step 2: Add one or more Resources to the Dataset

After creating a dataset, we're prompted to add one or more data resources to it. There are three types of Resources that can be added to a Dataset, depending on the type and location of the Resource:

Upload This option allows publishers to upload data files to the DKAN site. As in the "link to a file" option, the data within the file will be imported into your DKAN site's Datastore for preview and analysis by your users. See The DKAN Datastore for more information.

API or Website URL Some data resources aren't standalone files but queryable online databases; the interface to these databases is known as an API. Adding links to these types of online database interfaces to your DKAN data catalog can be very useful for developers interested in working with your data.

Remote file This option allows publishers to create a link to a data file published on another Internet website. Although the file itself will remain on the other site, the data within the file can be imported into your DKAN site's Datastore for preview and analysis by your users. See *Datastore* for more information.

Note: To provide *previews* for your resources, they must contain either a local or remote file (*Link to a file* or *Upload a file*). If you use *API or Website URL* your link will be displayed in an iFrame but not further previewing will be possible.

To continue with our Wisconsin Polling Places example, we'll add one resource file to the Dataset we created in Step 1. Our resource file is a CSV, that is, comma-separated values format; this is a popular file format for exchanging tabular data. Let's explore the example resource shown here and the various fields within:

Resource / Choose File upload a file from your local hard drive.

Resource / Recline Views DKAN's "Data Preview" feature allows visitors to preview published data in three views:

- **Map** - data with latitude and longitude columns or GeoJSON files can be previewed in a map interface
- **Graph** - tabular (spreadsheet) data can be graphed by users, letting them create their own meaningful visualizations
- **Grid** - by default, tabular data is presented in a spreadsheet view, with filter, sort, and search capabilities

Title this is the title of the individual data file, not the parent dataset container.

Description a rich-text editor field is provided so publishers can offer detailed and useful descriptions

Format entering the file format here will allow users the ability to search for data by specific format

Dataset this is the parent dataset container; this field should already be populated if you're adding a Resource subsequent to adding a Dataset

At the bottom of the *Add Resource* page, we can choose:

Save Save progress on this resource and immediately return to it for further editing

Save and add another Save this resource and add another resource to the same dataset

Next: Additional Info Save this resource and move to the third stage in adding a complete dataset, entering optional metadata about the dataset

In our example, we're only adding a single resource, so we'll click "Next: Additional Info" to move onto Step 3. If we had more than one resource to add to this dataset, we would choose the "Save and add another" option. Simply clicking "Save" would end the Dataset creation process and save the dataset, for now, with no additional metadata.

Step 3: Adding Metadata to a Dataset

We now come to a third form which allows us to add additional metadata to the dataset. All these fields are optional, but provide valuable information about your dataset to both human visitors to the website and machines discovering your dataset through one of *DKAN's public APIs*.

Let's take a closer look at some of the metadata fields available on this form:

Author The Dataset's author, in plain text.

Spatial / Geographical Coverage Area Lets us define what region the data applies to. In this case, the US State of Wisconsin. You can use the map widget to draw an outline around the state borders, or, click the "Add data manually" button if you already have a GeoJSON string you can paste in.

Spatial / Geographical Coverage Location The region the data applies to, written in plain text. This can be used instead of or in addition to the **Coverage Area** field.

Frequency How often is this dataset updated? We might expect our list of polling places to be updated every year, so we could select "annually." However, often we don't expect the data to be updated (even in this case, perhaps we plan to post the next version of the data as a *separate* dataset), in which case we can leave this blank.

Temporal Coverage Like Geographic Coverage, this field lets us give some context to the data, but now for the relevant time period. Here we could enter the year or years for which our polling places data is accurate.

Granularity This is a somewhat open-ended metadata field that lets you describe the granularity or accuracy of your data. For instance: "Year". Note, this field is depreciated in DCAT and Project Open Data, and may be removed from DKAN.

Data Dictionary This should be a URL to a resource that provides some sort of description that helps understanding the data. See Project Open Data data dictionary for more info.

Additional Info Lets us arbitrarily define other metadata fields. See *Additional Info field* for more information.

Resources This field is a reference to the resources you have already added.

After you click "Save", the metadata we enter will appear on the page for this Dataset:

Configuration

Learn how to *add additional file types to your resources here*.

3.1.2 Advanced Metadata Features

A Dataset is a container for storing files, APIs, or other resources as well as the metadata about those resources. The metadata in a DKAN Dataset is structured specifically for describing Open Data.

The metadata in a DKAN Dataset is culled from the DCAT standard as well as Project Open Data. For more information on the default Dataset fields view the *Open Data Field Comparison Tables*.

The Dataset form allows users to create Datasets and add appropriate metadata:

The DKAN Dataset API exposes Dataset metadata for individual datasets as well an entire catalog.

Custom metadata

It is easy to add new fields to DKAN which will show up on the Dataset form, make available as search facets, and be available to output in one of the Dataset APIs.

If there is information that only pertains to one or more datasets then it is possible to use the "Additional Info" field. This allows content editors to add unique field / value entries that exist only on a single dataset:

Globally-available custom fields can also be added through Drupal's Fields UI and added to public APIs using the *Open Data Schema Mapper*.

Data Extent

The "Data Extent" block is a visual representation of the "Spatial / Geographical Coverage Area".

The "Spatial / Geographical Coverage Area" field is a geojson representation of the area a Dataset covers. This can be a point, box, or other representation.

DKAN provides a widget so that a spatial area can be drawn if desired:

Note: By default the "Data Extent" block utilizes OpenMap's default tiles. DKAN makes the tiles configurable through the Drupal variable **dkan_map_tile_url**.

For example, to set the tiles to use Stamen tiles, run `drush vset dkan_map_tile_url https://stamen-tiles-{s}.a.ssl.fastly.net/terrain/{z}/{x}/{y}.png`

Revision History

DKAN Datasets and Resources track revisions in order to log and display changes, using Drupal's built-in revision system.

User Interface

Revision log entries can be added through the user interface by clicking "Revision information" in the dataset or resource edit form and can be viewed by clicking "Revisions" on a Dataset or Resource page:

Loading Revision information Programmatically

Revision comments generated in code can be viewed by loading a Dataset or Resource and viewing the log: *$node = node_load('dataset node id'); echo $node->log*

Revision List API

A list of recent revisions are available through the revision_list API at "/api/3/action/revision_list"

File Revisions

Copies are kept of files from previous revisions that can be compared manually by a user. Diffs of individual files are not available by default, but could be implemented with some custom code using Apache Solr and the Diff module, or a similar strategy.

3.1.3 Data Preview Features

Resources include powerful preview functionality via the Recline module. *See Visualizations/Data Previews*

3.1.4 Groups in DKAN

Groups allow you to group together datasets under an organization (i.e. Parks and Recreation Department, Department of Education) or category (e.g. Transport Data, Health Data) in order to make it easier for users to browse datasets by theme.

As a best practice, datasets and resources that are added to a Group should share a common publisher.

Essentially, Groups are both a way to collect common Datasets and enable an additional workflow on DKAN. On the outward-facing side, site visitors are able to browse and search Datasets published by a specific Group, which is the common publisher of a number of Datasets.

Behind the scenes, Groups add an additional set of roles and permissions that ensure quality and security when publishing data. Group roles and permissions ensure that Content Creators can add new data but only to their assigned Group. This is especially important for large sites that may have several working groups publishing data to the site. By adding users to the Group's membership roster, you can create and manage a community based around the data within the group.

How to use Groups

Adding a new Group

When adding a new Group, the form has fields for basic information about the Group itself that should tell site visitors what to expect from the Datasets in the Group.

Title Name your Group to reflect the agency or whoever the common data publisher is for the datasets that will belong to the Group.

Image The image here acts like the logo for your Group. It appears on the overview Groups page as well as the individual page of the Group itself. It's best to choose a square image to fit the dimensions of the thumbnail. Whether you choose an image, a logo, or an icon you can use any image that meets the size and file type requirements. As a Site Manager, you may want to add generic icons to the Groups you add if a current logo is unavailable.

Description This text is the full description for your Group similar to an "About" page. The description includes details about the agency, its goals, and information about the data it publishes. While you want to include all the relevant information of the Group, the best descriptions are 1-2 paragraphs long and include a link to the agency's main web page for more details.

Summary text You can use the Summary to create unique text for your Group. This text appears as a snippet under the Group image on the Group overview page. If left blank the first portion of the body text will be used (about 100 words). Including a summary can be useful in adding more key search terms or using a different tone to intrigue site visitors to learn more.

Managing Groups and Members

Once you've added a new Group, you can assign Datasets (and their Resources) to that Group. You can also manage the members of a Group, adding new members and giving certain members different roles. Members of a Group are bound by the permissions of their role and restricted to the content in their Group. As a Site Manager you have access to all Groups and are not limited by the permissions of the Group.

Roles and Permissions

With large sites there is often a need to have special permissions for a group of users to handle a specific set of content. Think of a large agency or department with sub-departments or programs that produce content. On the one hand these users shouldn't have the ability to manage or edit content for the entire site or other Groups. On the other hand it would be impractical for Editors or Site Managers to handle content for a large number of users. To keep content organized and in the hands of its owners without introducing the risk of inadvertent (and sometimes irreversible) actions, Group-level permissions give users the ability to do things they couldn't necessarily do on the site outside of the Group.

Within Groups there are different levels of access a user can have, which determines another level of permissions. Any user who belongs to a group falls into one of two types: Member or Administrator Member. Users not in the group are considered non-members.

Non-Member A non-member is any user on the site who does not belong to the Group. This role can request membership in the Group and view Group content.

Member A Member is a basic user within the Group who is mostly adding and editing their own content for the Group.

Administrator Member An Administrator Member is able to add and remove Group members and manage (create/edit/delete) all content within the Group. It's good practice to have only 1 or 2 users in this role for any given Group.

Managing users

Read more about *adding and managing group members here*.

3.2 Datastore

When you create a dataset with resources, DKAN is reading the data directly from the resource file or API.

The Datastore component provides an option for you to parse **CSV** or **TSV** files and save the data into database tables. This allows users to query the data through a public API.

In turn, by adding your CSV resources to the datastore, you are getting the fullest functionality possible out of your datasets.

Note: CSVs must be in UTF-8 format. If you're having trouble uploading a CSV file to the datastore, please make sure that you've exported it in UTF-8 format.

3.2.1 Importing Resources

When viewing a resource you will see the "Manage Datastore" button among the local task options.

Click the "Manage Datastore" button. The top of this screen will give you useful information about the current status of the resource and the datastore.

Importer The manager chosen.

Records Imported The number of rows from the CSV that have been imported to the datastore.

Data Importing The current state of the importing process. The following states are possible: ready, in progress, done, or error.

Step 1: Configure the Manager

DKAN provides two ways to manage data imports.

Simple Import this is the default manager and will be the only option unless you enable *Fast Import*.

Fast Import this is disabled by default as it is still experimental, you can enable it from the modules UI or with `drush en dkan_datastore_fast_import`. See *how to set up fast import here*

Step 2: Configure the import option settings for proper parsing

Adjust the defaults if necessary.

delimiter the character that separates the values in your file.

quote the character that encloses the fields in your file.

escape the character used to escape other characters in your file.

Step 3: Import

Click the "Import" button at the bottom of the page. Most files will complete the import process right away. Larger files will be processed in 'chunks' during cron runs. Be sure you have [cron](https://www.drupal.org/docs/7/setting-up-cron/overview) running at regular intervals.

Once the import is complete the Data Importing state will display 'done'. The Records Imported should match the number of rows in your file.

3.2.2 Datastore API

Your data is now available via the Datastore API! For more information, see the *Datastore API page*.

Click the "Data API" button at the top of the resource screen for specific instructions.

3.2.3 Dropping the Datastore

To remove all records from the datastore:

1. Visit the resource page.
2. Click the "Manage Datastore" button.
3. Click the "Drop" button.
4. Confirm by clicking the "Drop" button.

3.2.4 DKAN Fast Import Manager

> **Warning:** The *FastImport* Manager is disabled by default and only works with files hosted in the web server and with a properly configured mysql client and server.

DKAN provides a second manager: *FastImport*.

This manager allows the importing of huge CSV files into the datastore at a fraction of the time it would take using the regular import.

Requirements

- A MySQL / MariaDB database
- MySQL database should support *PDO::MYSQL_ATTR_LOCAL_INFILE* and *PDO::MYSQL_ATTR_USE_BUFFERED_QUERY* flags.
- Cronjob or similar to execute periodic imports.

Note: Because of the above requirements, which may not be available on all hosting environments, this module is *disabled* by default in DKAN.

Installation

- Inside your settings.php add a *pdo* element to your database configuration. For example:

```
<?php
$databases['default']['default'] = array (
  'database' => 'drupal',
  'username' => 'drupal',
  'password' => '123',
  'host' => '172.17.0.11',
  'port' => '',
  'driver' => 'mysql',
  'prefix' => '',
  'pdo' => array(
     PDO::MYSQL_ATTR_LOCAL_INFILE => 1,
     PDO::MYSQL_ATTR_USE_BUFFERED_QUERY => 1,
   )
);
```

- Go to **/admin/modules**, turn on DKAN Datastore Fast Import and press **Save configuration**. Alternatively you can use drush to enable this module: `drush en dkan_datastore_fast_import`.
- Make sure you **do not** see this message at the top of the page:

```
Required PDO flags for dkan_datastore_fast_import were not found. This module␣
↪requires PDO::MYSQL_ATTR_LOCAL_INFILE and PDO::MYSQL_ATTR_USE_BUFFERED_QUERY
```

Note: If you are using the docker-based development environment utilized by DKAN Tools, you will need to execute the following commands (take note that admin123 is the password of the admin user in that mysql environment):

```
dktl dc exec db bash
mysql -u root -padmin123
GRANT FILE ON *.* TO 'drupal';
```

Usage

To import a resource using Fast Import, follow the instructions previously given in *"Importing Resources"*.

Troubleshoot

- If you get an error like

```
SQLSTATE[28000]: invalid authorization specification: 1045 access denied for user
'drupal'@'%' (using password: yes)
```

you will need to grant FILE permissions to your MYSQL user. To do so use this command: `GRANT FILE ON *.* TO 'user-name'`

3.2.5 Important notes when upgrading DKAN to 7.x-1.16 from previous versions

Warning: Be sure to read the following breaking changes if you have code that depends on the datastore API.

Field names

Prior to the DKAN 7.x-1.16 release, the datastore field names matched the column headers exactly as they were in the CSV file used to create the datastore. This has changed in the 7.x-1.16 version of the Datastore, now the field names are transformed into lower case and spaces are replaced with underscores. For example, if your CSV file has a column name titled `School Year`, after importing the file to the datastore you will need to access it using `school_year` as the field name.

Empty values

On previous versions of DKAN, when there was a cell with an empty value in a CSV file, that value would be imported as NULL into the datastore. After importing a file to the datastore with DKAN 7.x-1.16 or later, those empty values are kept as empty strings and not as NULL values.

Use of feeds

Use of the feeds module as a method for importing data into the datastore has been deprecated in DKAN 7.x-1.16. It is important that if you have any implementations that still rely on feeds, you will need to refactor that code to use the new 'Simple Import', or add the patched version of feeds from 7.x-1.15.5 to your */src/make/drupal.make* file.

Also, it is important to note that when you upgrade from a previous version of DKAN, DKAN Datastore Fast Import will not be enabled by default. If you were previously using that option you will need to re-enable the module.

DKAN Datastore tables

On previous versions of DKAN, the import process was done via the feeds module, which produced tables with the field *feeds_flatstore_entry_id* as the primary key. Starting in 7.x-1.16, the feeds module has been replaced by **DKAN Datastore Simple Import**, and the new primary key for the DKAN datastore tables is a serial field called *entry_id*.

NOTE: If you are upgrading, existing datastore tables will NOT be updated automatically.

You have two options:

1. You can drop the datastore for each resource and re-import: this will delete the corresponding table and importing will create the table using the new schema.
2. Create a hook update in a custom module and add the following code:

```
$tables = db_query("show tables like 'dkan_datastore_%'")->fetchAll();
foreach($tables as $key => $value) {
  $table_name = reset($value);
  db_drop_primary_key($table_name);
  db_drop_field($table_name, 'feeds_flatstore_entry_id');
  db_add_field($table_name, 'entry_id',
    [
      'type' => 'serial',
      'not null' => TRUE,
      'unsigned' => TRUE,
    ],
    ['primary key' => ['entry_id']]
  );
}
```

This code will look for all of the dkan_datastore tables in your database, drop the primary key, drop the field *feeds_flatstore_entry_id* and add the new field *entry_id* as the primary key for each table.

3.3 DKAN Harvest

DKAN Harvest is a module that provides a common harvesting framework for DKAN. It supports custom extensions and adds drush commands and a web UI to manage harvesting sources and jobs. To "harvest" data is to use the public feed or API of another data portal to import items from that portal's catalog into your own. For example, Data.gov harvests all of its datasets from the data.json files of hundreds of U.S. federal, state and local data portals.

DKAN Harvest is built on top of the widely-used Migrate framework for Drupal. It follows a two-step process to import datasets:

1. Process a source URI and save resulting data locally to disk as JSON
2. Perform migrations into DKAN with the locally cached JSON files, using mappings provided by the DKAN Migrate Base module .

3.3.1 Harvest Sources

Harvest Sources are nodes that store the source's URI and some additional configuration. Users with the administrator or site manager role will be able to create and manage harvest sources.

Create a new harvest source

Go to `node/add/harvest-source` or Content > Add content > Harvest Source, and fill out the form.

Title Administrative title for the source.

Description Administrative description or notes about the source.

Source URI The full Uniform Resource Identifier, such as *http://demo.getdkan.com/data.json*.

Type data.json or data.xml - If the harvest source type you are looking for is not available, please refer to the *Define a new Harvest Source Type* section below.

The form includes four multi-value fields to help you fine tune the results of your harvest.

Filters Filters restrict the datasets imported by a particular field. For instance, if you are harvesting a data.json source and want only to harvest health-related datasets, you might add a filter with "keyword" in the first text box, and "heatlh" in the second.

Excludes Excludes are the inverse of filters. For example, if you know there is one publisher listed on the source whose datasets you do **not** want to bring into your data portal, you might add "publisher" with value "Governor's Office of Terrible Data"

Overrides Overrides will replace values from the source when you harvest. For instance, if you want to take responsibility for the datasets once harvested and add your agency's name as the publisher, you might add "publisher" with your agency's name as the value.

Defaults Defaults work the same as overrides, but will only be used if the relevant field is empty in the source

Project Open Data (as well as most metadata APIs) includes many fields that are not simple key-value pairs. If you need to access or modify nested array values you can use this dot syntax to specify the path: *key.nested_key.0.other_nested_key*. For example, the Publisher field in Project Open Data is expressed like this:

```
"publisher": {
  "@type": "org:Organization",
  "name": "demo.getdkan.com"
},
```

To access the name property for filtering or overriding, you can set *publisher.name* in the first text box and the value you want to use in the second one.

Harvest sources also include a *Topics* field. Chose one or more topics to apply them to *every* dataset harvested from that source.

Now click **Save**. The datasets from the source will be cached, and you will see a preview of what will be imported. This preview page shows a list of dataset titles and identifiers now in the harvest cache, allowing you to perform a basic check on your source configuration. If it does not look right, click the **Edit** tab to make adjustments.

Click **Harvest Now**. The datasets that were cached will now be imported into your site.

Harvest Source nodes are viewable by the public and provide some basic metadata to the user.

Warning: Some behaviors of the **Topics** field on harvest sources to be aware of:

- Changing the Topic on the source and re-harvesting will not update the Topic on harvested datasets if nothing else has changed. The Harvester will only re-import a dataset if it detects changes from the source.
- If you manually add additional topics to a harvested dataset, and there *is* a change at the source, the next time the dataset is harvested your topics will be overwritten.

Warning: Some behaviors of harvesting **resources** to be aware of:

- If only an **accessURL** value is given, the url will be saved to the **API or Website URL** field.
- If a **downloadURL** value is given, the url will be saved to the **Remote file** field.
- The maximum size of a managed file field is 255 characters. It is not possible to increase the size of the field as this would force MySQL to auto-convert the VARCHAR(255) to a SMALLTEXT datatype, which subsequently fails with error 1170 on key length if the column is used as primary key or unique or non-unique index. Therefore the Harvester will check the length of the url and if it exceeds 255 characters, will fall back to using the *effective url* which is the last url in a redirect chain.

Managing Harvest Sources

From the admin menu, navigate to DKAN > DKAN Harvest Dashboard to view *harvest sources*. The DKAN Harvest Dashboard provides site managers a quick overview of *harvest sources*, when they were last updated, number of *datasets*, and status of the source. The dashboard also allows site managers to perform manual harvest operations:

Harvest (cache and migrate) This will cache the source data locally and migrate that source data into your site content.

Cache source(s) This will fetch the source data, apply the source configuration (filters, excludes, etc.) and cache the data locally without migrating. You may wish to do this to check for errors, or to refresh the preview.

Migrate source(s) This will migrate the current cache for the selected sources, no matter how old it is.

Click on the title of a harvest source from the dashboard to see the details of that source. Administrative tasks can be accomplished from the tabs across the top.

View View the *harvest source* node.

Edit Click to make changes to the configuration of the *harvest source*.

Preview Click to pull the latest data from the source endpoint into the cache.

Manage Datasets An administrative view that lets you sort and filter the datasets from this source. The most powerful function on this page is to filter by **orphan** status. When a dataset that was harvested into your system previously is no longer provided in the source, it is considered "orphaned" on your site and unpublished. From the Manage Datasets screen, you can either permanently delete or re-publish orphan datasets.

Events Event Log that provides historical data on all harvests run on this source. The information is managed by the core `dkan_harvest` via a per-harvest source `migrate_log` table that tracks the number of *datasets* created, updated, failed, orphaned, and unchanged and status. If the value for the field Status is Error then you can click on the text to see the log error and identify the problem.

Errors Error log that shows a list of all errors recorded during harvesting on the source.

Harvested Resources

When datasets are harvested, the resources are added as remote files, which means they are links to the original files on the remote server. If you modify the *resource* node in your DKAN site, your changes will be overwritten the next time a harvest is performed. If you add a harvested resource to the *datastore* be sure to set up periodic importing so that the resource stays in sync with the source. For these reasons, we do not recommend that you create visualizations based on harvested resources as the visualizations could break when changes are made to the files upstream.

3.3.2 Harvest Drush Commands

DKAN Harvest provides multiple drush commands to manage *harvest sources* and control harvest jobs. In fact, once your sources are properly configured, running harvests from Drush on a cron job or other scheduling system like Jenkins is highly reccomended.

It is recommanded to pass the `--user=1` option to harvest drush operations (especially harvest migration jobs) to make sure that the entities created have a proper user as author.

List Harvest sources available

```
# List all available Harvest Sources
$ drush --user=1 dkan-harvest-status
# Alias
$ drush --user=1 dkan-hs
```

Run a full harvest (Cache & Migration)

```
# Harvest data and run migration on all the harvest sources available.
$ drush --user=1 dkan-harvest
# Alias
$ drush --user=1 dkan-h

# Harvest specific  harvest source.
$ drush --user=1 dkan-harvest test_harvest_source
# Alias
$ drush --user=1 dkan-h test_harvest_source
```

Run a harvest cache

```
# Run a harvest cache operation on all the harvest sources available.
$ drush --user=1 dkan-harvest-cache
# Alias
$ drush --user=1 dkan-hc

# Harvest cache specific harvest source.
$ drush --user=1 dkan-harvest-cache test_harvest_source
# Alias
$ drush --user=1 dkan-hc test_harvest_source
```

Run a harvest migration job

```
# Run a harvest migrate operation on all the harvest sources available.
$ drush --user=1 dkan-harvest-migrate
# Alias
$ drush --user=1 dkan-hm

# Harvest migrate specific harvest source.
$ drush --user=1 dkan-harvest-migrate test_harvest_source
# Alias
$ drush --user=1 dkan-hm test_harvest_source
```

3.3.3 Extending DKAN Harvest

DKAN developers can use the api provided by DKAN Harvest to add support for additioanl harvest source types. The `dkan_harvest_datajson` module encapsulate the reference implementation providing support for POD type sources.

If you need to harvest from an end point type other then POD. You can extend the DKAN Harvest APIs to implement said support by following a simple checklist:

- Define a new Harvest Source Type via `hook_harvest_source_types`.
- Implement the Harvest Source Type cache callback.
- Implement the Harvest Source Type Migration Class.
- (Optional) Write tests for your source type implementation.

Define a new Harvest Source Type

DKAN Harvest leverages Drupal's hook system to provide a way to extend the source types that DKAN Harvest supports. To add a new harvest source type the we return their definitions as array items via the `hook_harvest_source_types()` hook.

```
/**
 * Implements hook_harvest_source_types().
 */
function dkan_harvest_test_harvest_source_types() {
  return array(
    'harvest_test_type' => array(
      'machine_name' => 'harvest_test_type',
      'label' => 'Dkan Harvest Test Type',
      'cache callback' => 'dkan_harvest_cache_default',
      'migration class' => 'HarvestMigration',
    ),

    // Define another harvest source type.
    'harvest_another_test_type' => array(
      'machine_name' => 'harvest_another_test_type',
      'label' => 'Dkan Harvest Another Test Type',
      'cache callback' => 'dkan_harvest_cache_default',
      'migration class' => 'HarvestMigration',
    ),
  );
}
```

Each array item defines a single harvest source type. Each harvest source item consists of an array with 4 keyed values:

machine_name Unique string identifying the harvest source type.

label This label wil be used on the harvest add node form.

cache callback Cache function to perform; takes HarvestSource object and timestamp as arguments) and returns a HarvestCache object

migration class A registered Migrate class to use for this source type

Cache callbacks

```
/**
 * @param HarvestSource $source
 * @param $harvest_updatetime
 *
 * @return HarvestCache
 */
function dkan_harvest_datajson_cache(HarvestSource $source, $harvest_updatetime)
```

This callback takes care of downloading/filtering/altering the data from the source end-point to the local file directory provided by the `HarvestSource::getCacheDir()` method. The recommended folder structure for cached data is to have one dataset per uniqely named file. The actual migration is then performed on the cached data, not on the remote source itself.

```
sh
$ tree
.
- 5251bc60-02e2-4023-a3fb-03760551ab4a
- 80756f84-894f-4796-bb52-33dd0a54164e
- 846158bd-1821-48d8-80c8-bb23a98294a9
- 84cada83-2382-4ba2-b9be-97634b422a07

0 directories, 4 files

$ cat 84cada83-2382-4ba2-b9be-97634b422a07
/* JSON content of the cached dataset data */
```

The harvest cache function needs to support the modifications to the source available from the *harvest source* via the Filter, Excludes, Overrides and Default fields. Each of these configurations is available from the `HarvestSource` object via the `HarvestSource::filters`, `HarvestSource::excludes`, `HarvestSource::overrides`, `HarvestSource::defaults` methods.

Migration Classes

The common harvest migration logic is encapsulated in the HarvestMigration class, (which extends the MigrateDKAN class provided via the DKAN Migrate Base module. DKAN Harvest will support only migration classes extended from `HarvestMigration`. This class is responsible for consuming the downloaded data during the harvest cache step to create the DKAN *dataset* and associated nodes.

Implementing a Harvest Source Type migration class is the matter of checking couple of boxes:

- Wire the cached files on the `HarvestMigration::__construct()` method.
- Override the fields mapping on the `HarvestMigration::setFieldMappings()` method.

- Add alternate logic for existing default DKAN fields or extra logic for custom fields on the `HarvestMigration::prepareRow()` and the `HarvestMigration::prepare()`.

Working on the migration class for Harvest Source Type should be straightforward, but a good knowladge of how Migrate module works is a big help.

HarvestMigration::__construct()

Setting the `HarvestMigrateSourceList is the only logic required during the construction of the extended *HarvestMigration*. During the harvest migration we can't reliably determin and parse the type of cache file (JSON, XML, etc..) so we still need to provide this information to the Migration class via the `MigrateItem` variable. the Migrate module provide different helpful class for different input file parsing (`MigrateItemFile`, `MigrateItemJSON`, `MigrateItemXML`). For the the POD `dkan_harvest_datajson` reference implementation we use the `MigrateItemJSON` class to read the JSON files downloaded from data.json end-points.

```
public function __construct($arguments) {
  parent::__construct($arguments);
  $this->itemUrl = drupal_realpath($this->dkanHarvestSource->getCacheDir()) .
    '/:id';

  $this->source = new HarvestMigrateSourceList(
    new HarvestList($this->dkanHarvestSource->getCacheDir()),
    new MigrateItemJSON($this->itemUrl),
    array(),
    $this->sourceListOptions
  );
}
```

HarvestMigration::setFieldMappings()

The default Mapping for all the default DKAN fields and properties is done on the `HarvestMigration::setFieldMapping()` method. Overriding one or many field mapping is done by overrrding the `setFieldMapping()` in the child class and add/update the new/changed fields.

For example to override the mapping for the `og_group_ref` field.

```
public function setFieldMappings() {
  parent::setFieldMappings();
  $this->addFieldMapping('og_group_ref', 'group_id');
```

Resources import

The base `HarvestMigration` class will (by default) look for a `$row->resources` objects array that should contain all the data needed for constructing the resource node(s) associated with the dataset. the helper method `HarvestMigration::prepareResourceHelper()` should make creating the `resources` array items more streamlined.

Example code snippet:

```
/**
 * Implements prepareRow.
 */
public function prepareRow($row) {
  // Redacted code
```

```
  $row->resources = $this->prepareRowResources($row->xml);

  // Redacted code
}
```

Harvest and DKAN Workflow support

By default, DKAN Harvest will make sure that the harvested *dataset* node will be set to the `published` moderation state if the DKAN Workflow module is enabled on the DKAN site. This can be changed at the fields mapping level by overriding the `workbench_moderation_state_new` field.

3.4 DKAN Workflow

3.4.1 Introduction

For large organizations, it can be difficult to moderate vast amounts of content submitted by a wide range of publishers and agencies.

DKAN Workflow is an optional module for DKAN based on the Workbench family of modules.

With Workflow, site managers and administrators can determine which users are able to add, edit and delete content, as well as which users can view and approve of content under review.

Workflow creates a moderation queue so that content is published to the live site only after a designated supervisor or group moderator has reviewed and approved it.

When using DKAN Workflow, content exists in three states:

- **Draft** - A saved work in progress.
- **Needs Review** - The author feels the content is ready to go on public on the live site, and would like the supervisor to review it.
- **Published** - The content is public and visible on the live site.

The above image displays what you see on My Workbench after login. The Workbench navigation bar contains your content, drafts, and more. The "Create Content" menu features a list of content types you can create.

There are also three different Workflow roles, each with their own moderation permissions. These roles are Workflow Contributor, Workflow Moderator and Workflow Supervisor. For more on Workflow Roles and Permissions, please skip ahead to the "Workflow Roles and Permissions" section of this document.

Installing Workflow

DKAN Workflow is included on all out-of-the-box DKAN sites; however, it is not enabled by default. It can be enabled either from the Modules management page or by using drush.

```
drush en dkan_workflow -y
```

Enabling DKAN workflow will automatically enable all other required modules and add the Workflow Supervisor role to all users already assigned the site manager role. (More information available in the *Workflow Roles* section).

You may also see a message instructing you to rebuild permissions. If so, click the "Rebuild permissions" link to update the node access settings.

3.4.2 Requirements for DKAN Workflow

The DKAN Workflow component as a whole is comprised of three modules:

- DKAN Workflow
- DKAN Workflow Permissions
- Views Workflow List

In addition to these core modules, DKAN Workflow depends on multiple Drupal contrib modules:

- Workbench
- Workbench Moderation
- Workbench Email
- Drafty

Finally, the following Drupal contrib modules provide extra functionality (Menu and link badges, amongst other features):

- Link Badges
- Menu Badges
- Better Exposed Filters

All of the aforementioned dependencies are declared in the drupal-org.make file.

Workflow Roles and Permissions

The three Workflow roles correspond with the three core DKAN roles and permissions If a user is given a Workflow role, they must also be granted the corresponding core DKAN role.

- **Workflow Contributor = Content Creator**

Workflow Contributor is the most basic role; users with this role can add content, save as Draft or move it to Needs Review, but cannot publish content directly to the live site. They can only view content that they've created, and cannot modify the content of others.

- **Workflow Moderator = Editor**

Workflow Moderator is the middle role, mostly pertaining to moderating specific groups. This role reviews and publishes (or unpublishes) content for their group(s), rather than for the entire site. Workflow Moderators can also view and approve of content that has not yet been assigned to a group.

- **Workflow Supervisor = Site Manager**

Workflow Supervisor is the most powerful role and should only be assigned to highly trusted users. Users with the role of Workbench Supervisor can add, edit, modify, publish, unpublish, moderate or delete _all_ site content. This role is the only role that have access to the "Stale Drafts" and "Stale Review" tabs (more information below).

Here is how core roles in DKAN are automatically correlated to Workbench roles and permissions:

What a user will see	"My Drafts"	"Needs Review"
Workflow Contributor	Only content that they submitted.	Can see only content they have submitted.
Workflow Moderator	The content submitted to their organic group. Their own content.	The content submitted to their organic group. Their own content.
Workflow Supervisor	Only content that they submitted.	All the "Needs review" content.

My Workbench

When logged in as a user that has been assigned a Workbench role, the "My Workbench" button will be displayed on the site's main navigation toolbar.

"My Workbench" is also accessible directly via *admin/workbench*.

3.4.3 The "My Workbench" Moderation Toolbar

My content This tab provides a list of all of the content you've created.

My drafts Drafts you've written and drafts you have permission to view.

Needs review Content that Needs Review and can either be published to the live site or sent back to Drafts.

Stale drafts This tab contains Drafts that have sat for over 72 hours without a change in moderation state. (The "stale drafts" moderation state, as well as "stale reviews," are only visible to Site Managers.)

Stale reviews This tab provides content filed under Needs Review that has sat for over 72 hours without a change in moderation state.

3.4.4 Additional features:

Content Filters: Users can filter through content by *Title*, *Type* (Dataset, Resource, Data Story, etc), and *Groups*.

Bulk updates:. Certain operations such as changing content from Needs Review back to Draft can be applied to multiple items at once.

3.4.5 Editing Content

If you'd like to change the moderation state of an individual node (such as a dataset or resource), you can do so while editing the node itself.

Scroll to the bottom of the node's "Edit" page, and look under under the **Publishing options** sidebar; there, you'll see DKAN Workflow moderation state options.

Authors and reviewers can change the node's moderation state and add a note about the change via the **Moderation notes** text area.

3.4.6 Changing Notification Email Settings

For each DKAN Workflow moderation state transition (for example from *Draft* to *Needs Review*, from *Needs Review* to *Draft*, etc) the users with corresponding Workflow roles will receive a notification via email.

There are three scenarios in which one will receive email pertaining to DKAN Workflow:

1. They are the original content author.
2. They are a Workflow Moderator of a Group that the content has been assigned to.
3. They are a Workflow Supervisor, in general.

Emails will display the context that had triggered the notification as well as links to the updated content.

Advanced Options

3.4.7 Tweaking the Email template

To change DKAN Workflow moderation email templates, go to the *admin/config/workbench/email* configuration page. For more in-depth documentation, please review the *Workbench Modules Docs*.

3.4.8 Workbench Modules Docs

For more information, please refer to the following documentation:

- Workbench documentation on Drupal.org.
- Workbench Moderation documentation on Drupal.org.
- Workbench email documentation on Drupal.org.

3.5 DKAN Topics

While DKAN includes a free-tagging tags/keywords field for datasets, many data portals organize datasets into more predefined categories by subject matter. These are usually a small collection of subjects with logos that are incorporated into the site navigation. Neither DKAN's tags or "groups" (which are designed for grouping user permissions and usually represent organizational divisions) are exactly appropriate for this task.

The DKAN Topics module adds a "topics" vocabulary to DKAN, and corresponding functionality throughout the site. It adds a facet to the search/datasets page, and a pane to the default homepage. Topics can be administered through the standard Drupal taxonomy interface.

The included DKAN Default Topics module will add, on enable, a set of default civic topics using the Taxonomy Fixtures module.

DKAN Topics is enabled by default on new DKAN installations, with default terms loaded into the vocabulary. The module can be disabled and uninstalled, and all existing topics will be removed.

3.5.1 Permissions

- Users with the Site Manager or Editor role can add and edit topic terms.
- Users with the Administrator role can add new icons.

3.5.2 Adding a new topic term

From the Administration menu, navigate to `Site Configuration > Taxonomy > Topics > Add term`

Name Enter the term for your new topic.

Description This field is not currently displayed publicly.

Icon Type DKAN Topics comes with a default set of font icons that can be used with your terms, you can upload your own font icons if desired. See Adding new icons. Or you may select to use image icons, when you toggle the image option, an image upload input field will appear.

Icon If using font icons, select the icon you want to associate with your topic term.

Icon Color Icons will display the same color as text on datasets unless a specific color is selected here.

3.5.3 Editing topic terms

1. From the Administration menu, navigate to `Site Configuration > Taxonomy > Topics`
2. You will see a list of current topic terms, click the 'edit' link under Operations that corresponds to the term you would like to edit.
3. Make changes and click "Save".

3.5.4 Removing Topics from the main menu

1. Navigate to `Site Configuration > Menus`
2. On the Menus screen, click "list links"
3. Uncheck the box under "Enabled" for Topics
4. Click "Save configuration"

3.5.5 Adding new icons

The font used for Topics can only be changed if there are **NO** default icon values in use, only one icon font can be used at a time.

1. Navigate to `Configuration > Content Authoring > Font Icon Select Options`
2. Click "Upload New Library"
3. Enter a title for your new font option
4. Upload the four files for your font
5. Click "Save"
6. Navigate to `Structure > Taxonomy > Topics > Manage Fields > Icon`
7. Select your font from the font dropdown in the Icon field settings section.
8. Click "Save settings"

3.5.6 Known Issues

- This module adds a main menu link for "Topics". If you want a different word in place of "Topics", you can change the name in the main menu configuration but the icons in the dropdown will stop working. If you use String Overrrides you can change the Menu link title and the icons will continue to work, however the facet block title and the dataset form field title will still display as 'Topics'.
- Adding a new icon font for use with topics **needs work** to keep the icon functionality in facets and menus from breaking.

3.6 Fixtures and Default Content

Fixtures are a programming concept for default data that is included with application code for testing or other purposes. The data is provided in a structured format like XML or JSON, and imported into the database as part of a build process.

In DKAN, fixtures are used to provide datasets and other supporting content out of the box. The most visible use case for this will be DKAN's default content, which showcases DKAN's various capabilities. The fixtures themselves for default content are packaged in a separate sub-module, DKAN Default Content.

The DKAN Fixtures module provides tools to easily export all the content that lives inside a DKAN site into JSON fixture files, following a defined schema. Currently the content supported by the module are Groups, Resources, Datasets, Data Dashboards, Data Stories and Pages. Visualization Entites are also supported.

The module also provides basic Migrate classes that can be used to import content easily on a DKAN site.

3.6.1 Default Content Module

DKAN Default Content is the module that holds all the default content delivered with DKAN. All content is imported through the *fixtures* that can be found inside the /data directory. DKAN Fixtures was used to generate the default content fixtures and to migrate all the data using the migration clases that are provided.

Updating the fixtures

The default content fixtures can be updated easily through the DKAN Default Content module using the following steps:

1. All the content present on the DB is going to be exported, so be sure to clean all the content first.
2. Run `drush dkan-save-data`
3. All default content fixtures should be updated and saved inside the 'data' directory in the dkan_default_content module.

Please note that some rules should be followed when preparing the default content:

- Only published datasets are going to be exported.
- Only resources associated with datasets are going to be exported.
- Only groups that have associated datasets are going to be exported.
- The size of of the dkan_default_content module should be kept as small as possible, so small files and images should be used.
- When using internal visualizations as visualization embeds be sure to use the 'Local' option on the visualization embed settings, so that embeds are not pointing to a specific domain.

Importing default content

All the default content is imported automatically as soon as the DKAN Default Content module is enabled. Enable the DKAN Default Content module in the browser via admin/modules, or on the command line via drush:

```
drush en dkan_default_content
```

Removing the default content

All the default content is automatically removed as soon as the DKAN Default Content module is disabled with the exception of pages (Homepage, About page, etc). Disable the DKAN Default Content module in the browser via admin/modules, or on the command line via drush:

```
drush dis dkan_default_content
```

Upgrading pages from ctools panel pages to page nodes

Starting with DKAN 7.x-1.13, the default homepage has been converted from a Panel Page into a common page node. Page nodes now have panelizer enabled, so the full layout of a panel page can be reproduced in a simple page node. DKAN now provides a function that can be used to perform this conversion automatically. The provided function will:

- Generate an exact copy of the specified panel page automatically.
- Generate a new panelized page node and disable (not delete) the old ctools panel page homepage
- The new node page will be set as the site homepage if the $is_homepage parameter is set as 'true'.

Please note that some CSS adjustments might be needed in order for the node page to look exactly like the panel page, as CSS IDs and classes might be different.

The function can be found in the dkan_sitewide module and can be used as follows:

```
drush php-eval "dkan_sitewide_convert_panel_page(<page-name>);"
```

About paths

Pathauto is disabled for content created using dkan_fixtures because performance reasons. Instead, paths should be added to the fixtures using the path key.

3.6.2 Taxonomy Fixtures

A similar system exists for importing and exporting taxonomy terms as default content. The Taxonomy Fixtures module ships with DKAN but is available for us in other Drupal projects.

3.7 Federal Extras

The Open Data Federal Extras Module module extends DKAN Dataset to include selected Project Open Data and other federal requirements. See the Project Open Data Schema for more information (this module essentially adds the fields marked "USG"). It includes a list of U.S. federal bureau and program codes, with a script to keep program codes up-to-date.

3.7.1 Additional Fields

- Bureau Code
- Program Code
- Data Quality
- primary IT Investment UII
- System of Records

Enabling the module will add these fields to your Dataset content type. Note that *disabling* the module will *not* remove them. To remove the fields completely (which will permanently delete all data in those fields), *uninstall* the module from the module administration screen or via `drush pm-uninstall`.

Updating the Program Code list

1. Go to https://project-open-data.github.io/schema/#programCode
2. Download "Federal Program Inventory"
3. Export in csv to `fed_program_code_list`
4. cd 'fed_program_code_list'
5. Make sure filenames at beginning of `list_to_array.php` are correct.
6. Run `php -r " require 'list_to_array.php'; opfe_list_to_array_process();`

The Bureau Code List

The Bureau Code list is built from a CSV downloaded from seanherron/OMB-Agency-Bureau-and-Treasury-Codes. This repository has not been updated recently and new sources for this data are currently being considered.

3.8 Search

DKAN offers a faceted search similar to CKAN. This functionality is provided by the Search API and Search API DB modules. DKAN can easily be updated to use Apache Solr to power the search using the Search API Solr module.

3.8.1 Search API

The Search API module provides a framework for easily creating searches on any entity known to Drupal, using any kind of search engine. It incorporates facet support and the ability to use the Views module for displaying search results.

3.8.2 Apache Solr

Search API Solr Search provides a Solr backend for the Search API module, and delivers enterprise class, high performance search functionality. Apache Solr runs as a separate service from the web server and requires extra resources to integrate into your website. This can increase the price for hosting. Recommended for high traffic sites.

Requirements:

- Search API module

- Search API Solr Search module
- An Apache Solr server

For further details see the Search API Solr's handbook documentation.

DB vs Solr Search

Solr:

- PRO: Increased performance and scalability for complex queries.
- PRO: Reduce database size.
- CONS: Requires external service.
- CONS: Increases hosting cost.

DB:

- PRO: All information is in the same database.
- PRO: Easy to set up.
- CONS: Poor performance for sites with high traffic.
- CONS: Increased database size if you have a lot data to index.

Switching to Solr

To switch from the native database to Solr simply:

- Create or purchase a Solr instance
- Install search_api_solr
- Go to Configuration -> "Search API" then "Add server"
- Enter a server name and under "Service class" select "Solr service"
- After clicking "Create server" you should see a success message
- Update the Dataset index to use the Solr server.
  - Go to `admin/config/search/search_api`
  - Select **Edit** on the datasets index
  - Select the solr server you just added
  - Click **Save Settings**
- Re-index your site `admin/config/search/search_api/index/datasets`.

3.8.3 Extending the Search Functionality

If you have the ability to add custom or contrib modules to your DKAN codebase, you can extend the search functionality in several ways. These modules may be added to DKAN core in the future but for now are not officially supported.

"Did You Mean?" Spellchecking

To add spellcheck, simply install the Search API Spellcheck

Searching within Resource files

PDFs, CSVs and other files attached to Resources can be searched by using the Tika library. This functionality is made possible with the Search API Attachments module

Tip: See the Search API Handbook for more recipes and information.

3.9 Theme

DKAN takes advantage of Drupal's well-developed and flexible theming system, allowing administrators complete graphical control over the entire DKAN site. Features such as responsive page templates, accessible design elements, and built-in media management provide the latest in design and presentation technologies.

3.9.1 Default DKAN Theme

In Drupal and DKAN, the collection of page templates, fonts, colors, images, and other "look and feel" elements are known as a "theme." NuBoot Radix is the default theme for DKAN and is a subtheme of Radix which uses bootstrap styles and is compatible with panelized sites. This theme has some basic customization features built in, for many client sites, these configurations will be all that is necessary to meet the client expectations. The configurable items are:

- Logo (svg logos will display better on retina displays)
- Front page Hero image OR you can set a solid color background for the hero region.
- Favicon.
- Copyright text.
- Color options via the Colorizer screen
- Fonts: use font-your-face to use alternate fonts

DKAN administrators have the choice of customizing the existing DKAN Theme through theme settings, implementing an entirely new theme, or creating a *subtheme of nuboot_radix*.

By default, the DKAN Theme is located in: *[SITEROOT]/themes/nuboot_radix*

3.9.2 Theme and Appearance Settings

The DKAN Theme does provide a few simple customizations that administrators can use to change the default appearance of the site from the theme settings screen. If logged in as an administrative user, navigate to Appearance >> Settings

Site name and slogan

From the settings screen, you can toggle on/off the **site name** and **slogan**, simply check the box next to the elements you want to use.

Logo

Uncheck the 'Use the default' checkbox, and upload a new logo file in the logo image settings section.

Shortcut icon

If you would like to use a different favicon image, uncheck the 'Use the default' checkbox, and upload your own.

Copyright info

To change the **copyright** information that displays at the bottom of every page, edit the text in the copyright field and save.

Hero background image/color

The **Hero Unit** is the background image that displays on the front page. To use a different photo than the one supplied, click the 'Choose file' button to upload a new image. This image will expand to cover the full width of the browser so be sure to upload a horizontal/landscape image and not a vertical/portrait image.

Color scheme

To use the **colorizer** option, you must use the default theme as the admin theme. Navigate to *[SITE-ROOT]/admin/appearance* and scroll to the bottom. Confirm that the Admin theme is set to 'Default Theme'.

Now navigate to *[SITEROOT]/admin/appearance/colorizer* by clicking on the 'Colorizer' tab. Here you will see the color scheme options. There are a few default options you can select from the drop down, or you can enter hex values to create a custom color scheme, be sure to click 'Save Configuration' when finished. Your new colors are saved to a css file in your files directory. If you do not see your changes you may need to clear the colorizer cache by clicking the 'Clear Colorizer Cache' button. These colors will override all other styles in the theme.

Fonts

On the Appearance page, you will see a sub-menu item for **@font-your-face**. Navigate to *[SITE-ROOT]/admin/appearance/fontyourface*.

By default, the Google fonts api is enabled. If you click on the **Browse all fonts** tab, you will be able to select which google fonts to add to your site. Once you have made your selections, click on the **Enabled fonts** tab to view the font settings screen. Click on the **By CSS selector**, here you can select the css tag and what font should be applied to it using the table. To add fonts to specific css selectors, add them to the open text field at the bottom of the list, select a font from the dropdown next to it and click the **Save applied fonts** button For more information on how to use @fontyourface visit the project page.

Creating a new subtheme

To create a Nuboot Radix subtheme, run these commands

```
drush en radix

drush vset theme_default radix

drush radix "MyThemeName" --kit=https://github.com/GetDKAN/radix-kit-nuboot/archive/
↪master.zip
```

```
drush vset theme_default MyThemeName

drush dis radix
```

Your new subtheme will be placed in to the /sites/all/themes/ directory, it will contain only the directory structure, add your overrides where appropriate.

3.9.3 Theming Tools

Install Node and npm. You will use gulp for compiling the sass files. To get your local environment set up, follow these steps:

1. Install Node and npm. You can read a guide on how to install node here
2. Install bower: `npm install -g bower`.
3. Go to the root of your theme and run the following commands: `npm run setup`.
4. Update browserSyncProxy in config.json
5. Edit the files under the scss and js directory, these will be compiled into the assets directory. Run the following command to compile Sass and watch for changes: `gulp`.

3.9.4 Icon Fonts

The Nuboot Radix theme ships with two icon fonts:

dkan-flaticon

This font is used for file types (csv, pdf, xls, etc) designed by Freepik

The font files and the css are inside the Nuboot Radix theme *dkan/themes/nuboot_radix*. If you would like to use your own file type icons you can override the dkan-flaticon css by creating a custom theme. OR, if you would like to use the dkan-flaticon icons but NOT use Nuboot Radix as your base theme, you will need to copy the dkan-flaticon fonts and the dkan-flaticon.css into the theme you are using.

```
dkan/themes/nuboot_radix/assets/fonts/dkan-flaticon.eot
dkan/themes/nuboot_radix/assets/fonts/dkan-flaticon.svg
dkan/themes/nuboot_radix/assets/fonts/dkan-flaticon.ttf
dkan/themes/nuboot_radix/assets/fonts/dkan-flaticon.woff
dkan/themes/nuboot_radix/assets/css/dkan-flaticon.css
```

dkan-topics

This font is used for the Content Type and *Topics* icons, see Streamline Icons

If you would like to use your own icon font for Topics, use the *steps outlined here*.

3.10 DKAN Periodic Updates

The DKAN Periodic Updates module provides the user the ability to execute automated updates on existing resources on a daily, weekly or monthly basis. It also provides a UI in which you can check at the status of the updates. The

module requires a CSV file to execute the updates, this file is known as "the manifest", when you upload it and enable the updates, this is what happens:

- The manifest is evaluated on every cron run to determine which resources need to be updated (based on the frequency defined in the manifest and the "last update" date).
- For each resource that needs an update, the system will: - Download the file URL specified in the manifest (or link it, according to which file field is used on the corresponding resource). - Update the resource node to use the new file. - If the file is a CSV or TSV, then the resource is imported into the datastore.

3.10.1 Periodic Updates Settings

You can find the periodic updates settings under DKAN –> Periodic Updates –> Settings (*/admin/dkan/periodic-updates*). The settings for this module include:

- Periodic updates state
- Manifest for periodic updates

1- Periodic updates state

Enable periodic updates by checking the "Enable periodic updates" box on the Periodic Updates page, and click on "Save settings". You'll also need to upload a valid manifest in the section "Manifest for periodic updates", otherwise the updates cannot be executed. If you already have a manifest in the system but need to pause the periodic updates, all you need to do is uncheck the option "Enable periodic updates" and save settings.

2- Manifest for periodic updates

The manifest is a CSV file in which you define all of the resources that need to be updated periodically in the system, this file should contain the following column names:

`resource_id,frequency,file_url`

resource_id is the UUID related to the resource node.

frequency it can be set to *daily*, *weekly* or *monthly*. If you leave it empty or put any other string, then the system asumes it has to be imported daily.

file_url the URL of the remote file that needs to be downloaded or linked and saved to the node.

Once you upload the manifest and click on "Add file", the file will be recognized by the system.

Note: for the periodic updates to be executed you need to have uploaded a valid manifest and enabled the periodic updates. If you only have one of these items, the updates will not be executed.

3.10.2 Periodic Updates Status

The status page can be found under DKAN –> Periodic Updates –> Status (*/admin/dkan/periodic-updates/status*). In this page you'll get information about the resources specified in the manifest. If the "Enable periodic updates" checkbox from Settings is not marked, then you'll get a message saying "Periodic updates are disabled.". If it is marked but no manifest has been uploaded, then you'll get a message saying "No manifest was found.". If the updates are enabled and a valid manifest is uploaded, then you'll see a list of the elements included in the manifest file, that list includes:

- the Destination Resource ID: this shows the resource UUID specified in the manifest but it is also a link to the resource node when applicable,

- the Source File URL,
- the Update frequency,
- the Status: this represents the messages generated on each update, it will tell you wether the resource has been updated, if the process finished correctly or if/what errors were produced,
- the Last update date: this is the date in which the resource was last updated via periodic updates.

3.11 DKAN Roles and Permissions

Roles categorize types of users. Permissions are assigned to these roles and represent functions the user can perform across the site.

Below is a list of the roles and general permissions included in the **DKAN Permissions** module. The descriptions should help show what different user types are able to do on the site. Whan adding new users, site managers will assign the appropriate role(s) to match the tasks the user is expected to perform.

There will also be cases where you need users to have different permissions in the context of a particular **group** (for instance, to be able to modify content belonging to their agency but not other agencies on the same website). Read more about *group permissions here*.

3.11.1 The DKAN Permissions module

The DKAN Permissions module provides default roles and permissions for the DKAN distribution. It uses the export method provided by the Features Roles Permissions module so you can examine the specific roles and permissions provided by reviewing dkan_permissions.features.roles_permissions.inc.

3.11.2 Drupal Core Roles

Anonymous User

- General site visitors that are **not** logged in.
- Can view and search published content.

Authenticated User

These users have an account on the site, can log in, and have profile information that can be verified (and authenticated). This type has the **lowest** level of permissions.

Permissions:

- Can log in.
- Can edit their user profile.
- Can view and search published content.

Administrator

The administrator role holds **every** permission and requires high technical competency. This role has the ability to cause serious damage, so it's generally reserved for a single web professional. Administrators hold the **highest level** of all roles and permissions.

Permissions:

- Enable/disable modules and features.
- Change the appearance of the site with alternate themes.
- Create and edit user roles and permissions.
- Create views, blocks, features, content types.
- Access any UI configuration.

3.11.3 DKAN Core Roles

Content Creator

Content Creators can add content to the site. This will be someone working in your organization who helps by adding to the data catalogue but is not responsible for anything more. This level of access takes users into the production side of the site, but gives little freedom to move outside of creating and adding certain content types. Limiting this role is critical for avoiding inadvertent damage to site content.

Permissions:

- Create dataset, resource, data story, and data dashboard content.
- Create chart visualizations.
- Edit **own** content (can not edit content added by another user).
- View **own** unpublished content and revision history of all published content.
- Add and view files to site library.

Editor

This will typically be a person handling the content on a frequent basis. Someone in your organization with expertise on the subject-matter that is expansive as well as in-depth. An editor role is similar to a content creator role because the focus is on content, however, an editor will have the ability to manage and edit content added by others. The editor role does not go further into administrative functions.

Permissions:

- Create dataset, resource, data story, and data dashboard content.
- Create chart visualizations.
- Edit, delete, and manage versions of content added by other users.
- Add and view files to site library.

Site Manager

The highest level for *non-technical* users. A site manager is mostly concerned with the admin functions of the site. Typically this will fall to someone in a supervisory role. The site manager takes a high-level view of the site, its content, and the users on the site. This person is able to make general configurations to the site and assigns roles to new users but does not deal with the technical configuration of the backend.

Permissions:

- Create, edit, delete **all** content types created by **any** user.
- Create, edit and manage **Harvest Source** content.
- Assigns roles to all user levels, but cannot create new roles/perms.
- Create and manage groups.
- Manage site logo, name, slogan, copyright, colors, fonts, main menu, recline config, open data schema mapper, dataset forms and previews.

3.11.4 DKAN Workflow Roles

If your organizaton needs an editorial workflow for managing content creation and editing, DKAN also includes a feature called DKAN Workflow that adds three more roles to establish a content approval process. *Read more about that here*.

3.11.5 Installation/upgrade notes

On new DKAN installs, the DKAN Permissions module is enabled by default. The older ("sidewide"-namespaced) permissions module will still be included in the codebase to support existing sites, but will be disabled by default on new installs. For _existing_ sites, the opposite is true - updating your code will _not_ cause the newer module to be enabled automatically or disable the older permissions module.

If you have been using the older DKAN Sitewide Roles and Permissions module on an existing site and do upgrade, we do recommend you disable it and enable the new DKAN Permissions module. The newer module has an improved set of roles and permissions designed around what we consider the most general use-cases. However, this will likely mean reviewing all of the user accounts on your site and ensuring that they have the roles that they should.

You also of course have the option of disabling both modules, setting your own prefered roles and permissions and exporting those to a custom feature.

3.12 Content and Storytelling Tools

DKAN uses Panels to allow site editors to create customized layouts and manage content through a drag and drop interface. For more on how to use the editing tools *view the IPE section*.

3.12.1 Content Types

Page

Use the **Page** content type for basic informational content that describes the purpose of your website. Examples would include the front page, an about page, an FAQ page, etc.

Create a page

1. From the administration menu, navigate to Content > Add Content > Page.
2. Fill in the title and body fields.
3. Choose the desired layout.
4. Click Save.
5. Now add additional content by clicking the + buttons in the regions where you want the new content to display. Learn more about the In-Place Editor (IPE) *here*.
6. Click Save at the bottom of the page.

Data Stories

Use the **Data Story** content type for narrative content. Narrative content is designed to tell a story and show the impact data has on the everyday lives of citizens. By adding context and synthesis, you can bring out the personal elements of data, and help highlight the investment and value of open data that might not otherwise be apparent.

Create a data story

1. From the administration menu, navigate to Content > Add Content > Data Story.
2. Fill in the title, add a cover photo, add tags, topics, and a description.
3. Choose the desired layout.
4. Click Save.
5. Now add additional content by clicking the + buttons in the regions where you want the new content to display. Learn more about the In-Place Editor (IPE) *here*.
6. Click Save at the bottom of the page.

Data Dashboard

Mix videos, images, slide shows, visualizations, text, tables, and maps to most effectively deliver your content. With more than 20 responsive layouts to choose from and the easy to use drag and drop interface, any user can create compelling data-powered content within minutes.

Create a data dashboard

1. From the administration menu, navigate to Content > Add Content > Data Dashboard.
2. Fill in the data dashboard title.
3. Choose the desired layout.
4. Click Save.
5. Now add content by clicking the + buttons in the regions where you want the new content to display. Learn more about the In-Place Editor (IPE) *here*.
6. Click Save at the bottom of the page.

3.12.2 Using the in-place editor (IPE)

The In-Place Editor (IPE) interface at its core it is a drag and drop content manager that lets you visually design a layout and place content within that layout.

DKAN's customized IPE interface

DKAN has simplified the list of IPE options to show only the most used elements, this makes adding and editing content a faster and less daunting process.

Editing existing panels-based content

1. Go to a panelized page (in this case the DKAN homepage) and click "Customize this page":

2. Use the drag and drop controls to rearrange a given pane, click and hold the directional arrows button on the top right corner of the panel, drag it below the region name you want it to display in, wait until you see a yellow space light up before you let go :

3. Save changes by clicking the save button at the bottom:

Adding new panes to Panels-based pages

1. Go to a page managed by a panel layout and click the **Customize this page** button at the bottom:
2. Click the "plus" sign ("+") to add a new pane to any given panel region.
3. Select the type of content you would like to add:

Note:

- If you are adding a chart, use the **Visualization** pane.
- If you are adding a video, use the **Video** pane.
- If you want to embed an iframe that is not a video or visualization, wrap the iframe with the .iframe-container class to make it responsive on smaller screens.

```
<p class="iframe-container"><iframe src="..."></iframe></p>
```

---

4. Fill in the form that is presented, click 'Finish'.
5. Remember to click 'Save' at the bottom when you are done adding content to the page.

Altering the layout of an existing page

1. Go to the page you want to change the layout for and click **Change Layout**:

2. Select new layout:

3. Choose where existing panes belong in the new layout:

4. Click Save (or Save as Custom) and enjoy the new page:

3.13 Open Data Schema Map

This module provides a flexible way to expose your Drupal content via APIs following specific Open Data schemas. Currently, the CKAN, Project Open Data and DCAT-AP schemas are provided, but new schemas can be easily added through your own modules. A user interface is in place to create endpoints and map fields from the chosen schema to Drupal content using tokens.

This module was developed as part of the DKAN project, but will work on an Drupal 7 site. A separate module exists for DKAN-specific implementation.

Note that serious performance issues can result if you do not follow recommendations in the *ODSM File Cache section*.

3.13.1 Basic concepts

Schema

A schema is a list of field definitions, usually representing a community specification for presenting machine-readable data. The core Open Data Schema Map module does not include any schemas; they are provided by additional modules. A schema module includes:

- a standard Drupal .module file – with an implementation of `hook_open_data_schema()` to expose the schema to the core Open Data Schema Map module, plus _alter functions for any needed modifications of the UI form or the data output itself.
- the schema itself, expressed as a .json file. For instance, see the Project Open Data schema file to see how these schema are defined in JSON

API

An API in this module is a configuration set that exposes a specific set of machine-readable data at a specific URL (known as the API's endpoint). This module allows you to create multiple APIs that you save as database records and/or export using Features. An API record will contain:

- an endpoint URL
- a schema (chosen from the available schemas provided by the additional modules as described above)
- a mapping of fields defined in that schema to Drupal tokens (usually referencing fields from a node)
- optionally, one or more arguments passed through the URL to filter the result set

3.13.2 Usage

Installation

Enable the main *Open Data Schema Map* module as usual, and additionally enable any schema modules you will need to create your API.

Creating APIs

Navigate to admin/config/services/odsm and click "Add API."

Give the API a title, machine name, choose which entity type (usually *node*) and bundle (in DKAN, this is usually *Dataset*).

You will need to create the API record before adding arguments and mappings.

Arguments

The results of the API call can be filtered by a particular field via arguments in the URL. To add an argument, first choose the schema field then, if you are filtering by a custom field API field (ie, a field whose machine name begins with "field_"), identify the database column that would contain the actual argument value. Leave off the field name prefix; for instance, if filtering by a DKAN tag (a term reference field), the correct column is field_tags_tid, so you would enter "tid". Which Drupal field to use will be extrapolated from the token you map to that schema field.

Field Mapping

The API form presents you with a field for each field in your schema. Map the fields using Drupal's token system. Note: using more than one token in a single field may produce unexpected results and is not recommended.

Multi-value fields

For Drupal multi-value entity reference fields, the schema can use an array to instruct the API to iterate over each value and map the referenced data to multiple schema fields. For instance, in the CKAN schema, tags are described like this in schema_ckan.json:

```
      "tags": {
      "title":"Tags",
      "description":"",
      "anyOf": [
        {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "id": {
                "title": "UUID",
                "type": "string"
              },
              "vocabulary_id": {
                "title": "Vocaulary ID",
                "type": "string"
              },
              "name": {
                "title": "Name",
                "type": "string"
              },
              "revision_timestamp": {
                "title": "Revision Timestamp",
                "type": "string"
              },
              "state": {
                "title": "state",
```

```
                "description": "",
                "type": "string",
                "enum": ["uncomplete", "complete", "active"]
              }
            }
          }
        }
      ]
    },
```

You can choose which of the available multivalue fields on your selected bundle to map to the "tags" array, exposing all of the referenced "tag" entities (taxonomy terms in this example) to use as the context for your token mappings on the schema fields within that array. First, simply choose the multivalue field, leaving the individual field mappings blank, and save the form.

When you return to the tags section of the form after saving, you will now see a special token navigator you can use to find tokens that will work with this iterative approach (using "Nth" in place of the standard delta value in the token):

3.13.3 Customizing

Adding new schemas

You are not limited by the schemas included with this module; any Open Data schema may be defined in a custom module. Use the open_data_schema_ckan module as a model to get started.

Date format

Date formats can be chanaged manually by changing the "Medium" date time format in "admin/config/regional/date-time" or in code by using one of the alter hooks:

3.13.4 A Note on XML Output

Open Data Schema Map provides an XML output format. This is provided via a separate submodule in the `modules/` folder for historical reasons, but should be refactored into the main ODSM module in a future release.

XML endpoints still require a *schema* defined in JSON. Defining your own XML endpoint may be less than intuitive for the time beind, but take a look at the DCAT schema module for a model.

3.13.5 The ODSM File Cache

Open Data Schema Map endpoints that list a large number of entities – Project Open Data (`data.json`), the CKAN Package List (`/api/3/action/package_list`) and DCAT-AP Catalog (`catalog.xml`) – perform a full entity load for each record listed in order to perform the token replacements. This can cause a major performance hit each time any of these URLs is hit on a site with more than a few dozen datasets, and on a site with thousands the response time can be two minutes or more.

Open Data Schema Map includes a file caching function to save a snapshot of any endpoint as a static file to be served up quickly, with very few hits to the database.

File caches can be generated either via a Drush command, or an admin UI. The recommended usage on a production website is to set up a cron job or use a task runner like Jenkins to regenerate the file caches for your performance-intensive endpoints daily (usin the drush command), at whatever time your site experiences the least amount of traffic.

The trade-off of course is that any additions or changes to your site will not be reflected on these endpoints until they are regenerated.

Drush Use

The Drush command supplied by Open Data Schema Map is `odsm-filecache` (also available simply as the alias `odsmfc`). This command takes as its argument the machine name for an ODSM endpoint. For example:

```
drush odsm-filecache data_json_1_1
```

This will render the full `data_json_1_1` endpoint (which is the `data.json` implementation that ships with DKAN) to the filesystem, saving it to:

```
public://odsm_cache_data_json_1_1
```

Now a hit to `/data.json` will be routed to this file, which in most cases will actually live at `/sites/default/files/odsm_cache_data_json_1_1`.

UI Use

An administrative UI to regenerate file caches manually is also included. This interface is useful in cases where manual creation of the cache files is sufficient.

To use, navigate to admin/config/services/odsm where there is a column called "Cache" with links to the individual admin pages for specific enpoint caches. If there is no cache the link is labled "none", otherwise the link is labled with the age of the cache in hours. From the cache admin pages you can create, delete or regenerate the cache.

3.13.6 Schema Validation

Both the Project Open Data and DCAT-AP schemas ship with validation tools you can access from the Drupal admin menu. More documentation on this feature coming soon...

3.13.7 Community

We are accepting issues for Open Data Schema Map in the DKAN issue queue only. Please label your issue as **"Component: ODSM"** after submitting so we can identify problems and feature requests faster.

If submitting a pull request to this project, please try to link your PR to the corresponding issue in the DKAN issue thread.

3.14 Visualizations

DKAN provides visualization functionality in two primary ways: through the automatically-created "data previews" that are displayed with data *resources*, and the more-complex and manually-build visualization "entities":

3.14.1 DKAN Data Preview Features

DKAN allows users to have a preview of their data when uploaded to or linked to a *resource*. Which preview type is displayed for a particular *resource* depends on the data **format** selected. If no data format is provided, DKAN

will attempt to auto-detect the format from the file's metadata; re-editing the *resource* and correcting the format field manually may be necessary if the auto-detection is not successful.

This functionality is provided via the Recline module, which is not part of the core DKAN repository but is a basic dependency of it (and will be included when building the distribution via `drush make`).

Recline.js

DKAN, like CKAN, offers an integration with the Recline Javascript library. Recline allows site visitors to preview tabular data visually. The preview works for CSV and XLS*[0] files that are uploaded to the DKAN site or hosted remotely and linked to, as well as for data stored in DKAN's local SQL-based *datastore*.

Record limit: The previews will display up to 100 records by default. You can use the pager to preview additional results based on the range given. If you want to preview more than 100 records at a time, adjust the max range value to the desired range.

Grid View

All tabular data can be rendered as spreadsheet-style rows and columns:

Map View

Visitors can preview data that contains either coordinates or GeoJSON on a Leaflet.js -based map:

Graph View

If enabled, visitors can chose one column of your data as an X-axis, one or more as Y-axis data, and preview your data as a bar, point or line graph.

File size limits

Files can only be previewed if they are well formatted and small enough to render in the browser.

If files are too large to preview within 1 second you will get the following message *"File was too large or unavailable for preview."*

Files that are too large to preview in the browser can be previewed by *adding them to the datastore*. Once a file is in the datastore the preview is only asking for the first 25 rows of the data. Thus large datasets can be previewed.

Additional Preview Types

DKAN provides preview formats for several additional file types beyond what is supported by Recline.js, these include: JSON, geojson, XML, ArcGIS REST, WMS, images, PDF, and ZIP files. These additional preview formatters are defined in a forked version of Recline

[0] For xls files be sure to fill in the format field to see previews of the data

Zip files

DKAN offers the ability to preview the files and folders locked in ZIP files. DKAN will display a list of contents for ZIP files uploaded as resources on datasets.

Image files

Image files (JPG, PNG or GIF) uploaded as resources will be displayed directly on the resource page.

Web Map Service (WMS)

DKAN can use Leaflet to display a preview of a WMS server, provided endpoint.

WMS support in DKAN is still somewhat experimental and your results may vary.

ArcGIS REST

An ESRI/ArcGIS REST endpoint may also be displayed in a Leaflet preview.

JSON files

GeoJSON files

XML files

PDF files

External Previews

Starting with version 7.x-1.10, DKAN supports previewing/opening resources in external services that offer simple URL-based integrations. For instance, the CartoDB mapping service offers an Open in CartoDB service. Enabling this for CSV files will result in a dataset display like this:

External preview functionality can be enabled and configured in the "DKAN Dataset Previews" administration page (/admin/dkan/dataset_preview).

Configuration

By default previews are available for resources with files below 3MB of size. However you can customize this limit in the recline configuration page (*/admin/dkan/recline*).

3.14.2 Visualization Entity

Visualization Entity is included in the DKAN distribution.

This module **aims** to provide:

- A base drupal entity to extend and create **visualization bundles**.

- A set of visualization bundles that provide functionality out of the box for the module and indicate a good example for extending this entity.
- A set of permissions common to all visualization bundles.
- A common "iframe view" shared between all visualization bundles.
- A common "embed" functionality shared between all visualization bundles.

Visualization Chart

Visualization Entity Charts is enabled by default in DKAN. This modules provides the ability to create embeddable NVD3 charts.

Usage

New chart entities can be created by going to `/admin/structure/entity-type/visualization/ve_chart/add`. A multi-step process will guide you through the creation of a chart based on an uploaded data file.

Step One - Choose a Resource

- Enter a title for the chart.
- Enter a description if needed.
- Then start typing the title of a resource that you would like to use as the data source. A list will appear, select the resource from the list.
- OR, if the data you want to use is not on your site, click the Upload Data tab to upload a CSV data file.
- Click the Next button.

Step Two - Define Variables

- **Series:** Add all the columns you would like to plot along the y-axis, the **value** axis. A collection of related values is what makes up a 'series'.
- **Y-Field Data Type:** The data type will be auto-detected but if you see issues you can manually select the data type here.
- **X-Field:** Choose a single column for the x-axis, the **category** axis.
- **X-Field Data Type:** The data type will be auto-detected but if you see issues you can manually select the data type here.

Step Three - Choose Chart Type

Select the chart type that will best represent your data. **NOTE**: X and Y Axis Fields are not supported by the *Pie Chart* type.

Step Four - Preview and Adjust

You can adjust colors, margins, include a goal, labels, tick values, and more. Click the question mark icons if you need help understanding the configuration options.

By default the chart will use the first 100 records of your data source. To use all records, click the Dataset tab to reveal the data pager, edit the max range value from 100 to the total number of records present.

Query Editor

Click the '+' on the query editor to see the query input field. Enter text to query the data. Returned rows will contain data matching your text (including partial text matches). Click on the Dataset tab to better see how the data is modified by your query.

Filter Editor

Click the '+' on the filter editor to add one or more filters to limit the data used for the chart. Multiple filters will be applied with the AND operator (all criteria must be met for the data to be included in the chart).

1. Create a filter
   - Select the field you would like to filter by.
   - Select filter type: Select Value to filter by strings (labels), select Range to filter by numerical values, and select Geo distance to filter by geographical data.
   - Click **Add**
   - Value filters check for exact matches (no partial text matches; use the Query Editor instead if you need to search for partial text matches)
2. Configure the filter
   - Fill in the fields to complete the filter.
   - Click **Update** to reload the chart.

To remove a filter, click the trash can icon next to the filter name.

Chart Configuration

X Axis

- **Format** Select an appropriate format for the X Axis labels.
- **Axis Label** will provide a custom label for the x axis.
- **Note:** Axis labels do not display for Pie Charts.
- **Label rotation** will change angle of label values.
- **Tick Values** Enter a numerical range to set the start and end values to display.
- **Step**: Use the Step field to define the value between each tick within the range. **NOTE:** If the range set for tick values is smaller than the range of complete data represented, the chart will be abbreviated.

Y Axis

- **Axis Label** Provides a custom label for the y axis.
- **Note:** Axis labels do not display for Pie Charts.
- Adjust the *distance* field if your axis label overlaps the y-axis data labels. You can move the label left with positive values, and right with negative values. You may need to adjust the left margin of the chart as well.
- **Tick Values** Enter a numerical range to set the start and end values to display.
- **Step**: Use the Step field to define the value between each tick within the range. **NOTE:** If the range set for tick values is smaller than the range of complete data represented, the chart will be abbreviated.

General

Color Set the color the chart is drawn in. Use either a HEX color code or a valid css color name Separate multiple colors with commas.

Goal Overlay a goal or target line on the chart.

Margin Enter value of margin in the order: *top, right, bottom, left*

Show Title Display the title you entered on step 1.

Show Controls Whether to show extra controls or not. Extra controls include things like making multi-Bar charts stacked or side by side.

Show Legend Display a legend for the chart.

Show Tooltips Shows data and label on hover.

Group By X Field If there are two or more rows that have the same value in the column assigned to the x-axis field, those rows will be combined and display as a single data point. This is only relevant for combining numerical data.

Fewer X-axis Labels Reduces the number of labels displayed along the x-axis.

Save the chart

Remember to click **Finish** to save your configuration changes.

Recline

The bundle also includes an integration with the Recline module. If you have a content type with a recline file field, you can add a Recline Field Reference field to your chart bundle. This field type is defined in a module that comes bundled with Visualization Entity. The included DKAN integration module adds a Recline Field Reference pointing specifically at DKAN's Resource content type. In this case, entering an existing Resource node in the reference field will automatically populate the resource file into the chart entity's file field.

GeoJSON

Warning: Under Development. Do not use on production.

Enable the geojson bundle:

```
$ drush -y visualization_entity_geojson_bundle
$ drush cc all
```

Create Visualization

- Look for **Content -> Add Content -> Resource** in the admin menu and click on it.
- Upload a geojson data file for the resource
- Fill the required fields, enter 'geojson' in the format field, and **save** the resource
- Look for **Structure -> Entity Types -> Visualization -> Geojson Visualization -> Add Geojson Visualization** in the admin menu and click on it.
- Set a **title**
- Select the **resource** containing the **geojson** data file you uploaded
- Click **Save** & Enjoy!

Choropleth

> **Warning:** Under Development. Do not use on production.

Enable the choropleth bundle:

```
$ drush en -y visualization_entity_choropleth_bundle
$ drush cc all
```

Examples Files

Two example files are provided in the **examples** folder:

```
africa.geojson
africa-data.csv
```

Create Visualization

- Look for **Content -> Add Content -> Resource** in the admin menu and click on it.
- Upload a **africa-data.csv** file from the examples folder for the resource.
- Fill the required fields, enter 'geojson' in the format field, and **save** the resource
- Look for **Structure -> Entity Types -> Geo File -> geojson -> Add geojson** in the admin menu and click on it.
- Set **Title**
- Upload a **geojson** file
- Fill **name attribute** with the **column name** in the data (csv resource) that will match the **name** property for the features in the **geojson** file.

- Click **Save**.
- You'll get a preview for the geojson file you just uploaded.
- Look for **Structure -> Entity Types -> Visualization -> Choropleth Visualization -> Add Choropleth Visualization** in the admin menu and click on it.
- Fill Title
- Select the **geojson** file we created for the **geojson** field.
- Select the **resource** file we created for the **resource** field.
- Select the **colors** you like to use for the choropleth map.
- Fill **data column** with the column in the csv data you'll like to pick as the source of numerical data for the polygon coloring. If you leave this field blank, you'll get a list of radio buttons to pick up the column when the visualization gets rendered.
- Fill the **data breakpoints** with comma separated numbers. If you leave this field blank, breakpoints will be calculated for you based on the data.
- Click **Save** & Enjoy!

This module is **still on early development**.

Additional visualization functionality can be found in the following projects, which are not included in the core DKAN project and are still in a relatively expiremental state:

- DKAN Datastore CartoDB Integration
- React Dash library

3.15 DKAN Link Checker

DKAN Link Checker adds configuration and additional reporting to the link checker module.

> The Link checker module extracts links from your content when saved and periodically tries to detect broken hypertext links by checking the remote sites and evaluating the HTTP response codes.

3.15.1 Installation

DKAN Link checker will check links in datasets, resources, and harvest sources. It is not enabled by default, to enable it, run these commands:

- `drush en dkan_linkchecker -y`
- `drush cc all`
- `drush linkchecker-analyze`
- `drush cron`

Links will be processed in batches and it may take a while to go through all of the links of your site.

3.15.2 Permissions

Users with the **site manager** role will be able to

- View the broken links report at `admin/reports/dkan-linkchecker-report`

- Access the link checker configuration screen at `admin/config/content/linkchecker`

For more information on link checker click here

Note: The three modules mentioned above that are not distributed with DKAN continue to be maintained in separate repositories because they work independently of DKAN and could be installed in non-DKAN Drupal sites.

CHAPTER 4

DKAN Community

4.1 Welcome!

We're excited you're here. There are several ways you can get involved with DKAN, but be sure to review our *Code of Conduct*.

4.1.1 Slack

Join the DKAN Slack community!

This community resource is for DKAN developers and users to share how they're using the platform, help answer each other's questions, discuss improvements they'd like to see, and find ways to contribute to DKAN. You'll meet developers, open source and open data enthusiasts, city, state, and government officials, scientists, and interesting people all working toward a more transparent, smarter, data-driven world.

Join the community

- Join the DKAN Slack community
- Complete your profile (bio/photo).
- Join as many or as few channels as you'd like (you'll see descriptions for each channel detailing its purpose).

4.1.2 Connect

- Follow the DKAN Blog on Medium
- Follow @getDKAN on Twitter
- Attend DKAN Events

4.1.3 GitHub

For developers and contributors:

- Pull requests welcome! Add a link to the issue it fixes and a detailed description about your fix.
- Submit an issue at the official DKAN repo Please add detailed steps on how to reproduce the error, what version of DKAN you are using, and any environment details that could help expedite work on the problem.

4.2 Open Data Standards & Tidy Data

A basic overview of open data standards and best practices.

4.2.1 The open data publishing process from start-to-finish

Fig. 4.1: This flowchart displays the steps of the open data publishing process.

4.2.2 Datasets, Resources and Groups

How DKAN organizes data

All open data catalogs built on DKAN are organized by Datasets, Resources and Groups.

These three content types serve as a means of organizing data for maximum site functionality, efficiency, and user-friendliness.

You can picture a DKAN-based open data portal as a massive file cabinet, as in the below diagram:

Fig. 4.2: This image displays a DKAN Open Data site as a file cabinet with Datasets, Resources and Groups represented as folders, files and drawers.

Datasets Datasets are used to group related pieces of data and their associated metadata. These pieces of data - known as Resources - can then be found under a single url along with descriptions and licensing information.

Datasets serve as the folders for the Resources within. In real life, a typical folder could contain a printed spreadsheet, a PDF of a map, and a CD-ROM within it. In DKAN, you can similarly add a wide number of Resources in various formats to a single Dataset.

Resources In DKAN, Resources are the actual files, APIs or links that are being shared within a Dataset. See the "Dataset" definition for an image of a Dataset containing two Resources.

Resource types include csv, html, xls, json, xlsx, doc, docx, rdf, txt, jpg, png, gif, tiff, pdf, odf, ods, odt, tsv, geojson and xml files.

If the Resource is an API, it can be used as a live source of information for building a site or application.

Resources can be viewed within your browser using DKAN Data Previews or used to create visualizations.

Groups Each Group is home to Datasets and Resources that share a common publisher.

Typically, the publisher is an organization or governmental agency (i.e. Dept. of Education, Health Care Administration, Transit Authority, etc.).

Groups also enable you to assign roles and permissions to members of the group. By adding users to the Group's membership roster, you can create and manage a community based around the data within the group.

4.2.3 Metadata

The "Who, What, When, Where, Why" of data

What is metadata?

As mentioned above, metadata is the "Who, What, When, Where, Why" of data.

To return to the file cabinet metaphor, metadata is the information written on the outside of each folder about the folder itself and the contents within.

Metadata contains the attributes that describe each dataset. Examples include the name of the dataset's author, the title of the dataset, the date that it was last updated, any relevant contact information associated with the dataset, and more.

Metadata can be qualitative or quantitative in nature. Quantitative data is data that can be evaluated numerically; if the variable can be counted or measured, it's quantitative. Qualitative data is data that is descriptive, and cannot be measured.

An example of qualitative metadata:

"Author: Leslie Knope, Agency: Dept. of Parks and Recreation."

An example of quantitative metadata:

"Downloaded: 311 times, Last Accessed: 03-16-2016, 5:51 p.m."

Where is metadata displayed in DKAN?

When viewing a dataset, you can scroll down the page to the "Dataset Info" section to view its metadata.

The screencap below is an example of metadata for the "Safety Net Clinics" dataset on HealthData.gov:

Fig. 4.3: Metadata in DKAN is displayed as a table on the Datasets page

An example of how metadata is presented within DKAN.

Source: * https://www.healthdata.gov/dataset/safety-net-clinics

4.2.4 Open Data Formats

An open data format is a file format for storing digital data, defined by a published specification usually maintained by a standards organization, and which can be used and implemented by anyone.

Common open file formats include csv, html, xls, json, xlsx, doc, docx, rdf, txt, jpg, png, gif, tiff, pdf, odf, ods, odt, tsv, geojson and xml files.

All of the above file formats are supported as data resources in DKAN.

CSV Files

Introduction

A CSV (comma-separated values) file stores tabular data (numbers and text) in plain text, with each line of the file serving as a data record. Each record is made up of one or more fields, separated by commas - this usage of the comma as a field separator explains the name of the format.

In plain English, CSV files are a handy way to store data that could otherwise be displayed as a table. The structure of CSV files makes them easy to parse by nearly every database, spreadsheet or statistical analysis application in existence - including, of course, DKAN.

Here is an example of how a CSV file stores a table containing the quiz grades of four students, sorted by date:

“Date”,”Student”,”Grade”

“March 16th”,”Smith, Sally”,”A”

“March 25th”,”Williams, Jane”,”B”

“April 1st”,”Smith, Sally”,”C”

“April 8th”,”Doe, Jonathan “”Johnny””,”A”

Exported as a table, it looks very similar:

Date	Student	Grade
March 16th	Smith, Sally	A
March 25th	Williams, Jane	B
April 1st	Smith, Sally	C
April 8th	Doe, Jonathan “Johnny”	A

Each field is separated by a comma, and each new line represents a separate data record.

There are no official standards for the CSV file format itself, given that it has been in use since 1972 before such file format standards existed. That’s right - this widely-accessible format that can store millions of data records in a single file has existed and been in continuous use since the days of punch-card computing.

This is because CSV files have always been easy to type, are less prone to producing incorrect results if a record is entered incorrectly, and as shown above, is easily human-readable. These are still the major advantages of the format when it comes to viewing, editing, accessing and storing open data.

Possible issues with storing data resources in CSV format

One issue users may encounter with delimiter-separated files is the usage of an alternative delimiter such as a semicolon or tab instead of a comma. These files may be mistakenly saved as .csv files, causing problems when a program built to parse comma-delimited files attempts to read them. **

CSV Files vs. Microsoft Excel (.XLS) files in DKAN

CSV files can also only contain one “page” of data; therefore, unlike in Microsoft Excel, you cannot save a “workbook” with multiple pages representing multiple tables.

For example, if you had multiple pages of student grade data in Excel - one for the students of the Fall 2015 class, one for the Spring 2016 class and another for Summer 2016 - these would have to be saved as three separate CSV files.

On the other hand, CSV files can contain millions upon millions of rows. There is literally *no limit* to how long they can be, making them more practical than XLS files, which can only contain 65536 rows and 256 columns.

When would you need millions of rows? Examples include: All of the traffic tickets issued by a major metropolitan city within a calendar year, or open transit data containing a data record for every single bus arrival at every bus stop

for every single route in the city. For NYC, these real-life open data files are truly massive, beyond what Excel can handle.

Issues you may encounter when importing data via Microsoft Excel

Excel will open .csv files, but as mentioned, depending on the system's regional settings, it may expect a semicolon as a separator as opposed to a comma. This is because, in some countries and languages, the comma is used as the decimal separator. (i.e. displaying the number 3.14 as 3,14.)

Excel may also auto-reformat what looks like numbers, eliminating leading + or 0 characters. This can break data records starting with those characters, such as phone numbers. Many regional versions of Excel also cannot deal with Unicode in CSV files.

One solution when encountering these issues while attempting to open a CSV file in Excel is to change the filename extension from .csv to .txt; then opening the file manually with the Excel "Open" command. This will allow you to manually specify the delimiters, encoding, format of columns, etc. A preview is displayed so that you can be sure the file looks the way you want it to.

When saving a CSV file in Excel, a prompt will appear warning you that Excel's formatting cannot be saved with the file. This is referring to formatting you may have applied to the file such as bold text, special colors or fonts, added images or anything else that goes beyond the limitations of the format.Each field is separated by a comma, and each new line represents a separate data record.

4.2.5 Data Standards 101

Data is more useful when more people can use it.

What are data standards?

Data standards are the rules that help keep the publishing and organization of open data orderly and efficient.

If you're a writer, you may be familiar with the various stylebooks used to standardize the writing and design of documents in terms of grammar, punctuation and sources cited.

Examples include the AP Stylebook for news reports, and the Chicago and MLA manuals of style for academic papers. You can think of the various guidelines in the stylebooks as data standards applied to written content.

A listing of common standards for open data has been provided within this document.

Why do data standards matter?

Standards for data and metadata formatting and organization matter because of interoperability and functionality between datasets.

One of the most important parts of data analysis is figuring out the relationships between data resources - and if multiple resources have been prepared in a single standardized format, it makes comparing them between one another far more efficient.

One example that pertaining to data standardization is how dates are formatted:

April 2, 1974

04-02-74

04/02/1974

4/2/74

19740402

04021974 - is this April 2 or February 4?

2 April 1974

If you were trying to compare or join together datasets from different sources, each of which used a different format for their date variable, it would be a much more difficult task than if a common date format had been decided upon ahead of time.

For a listing of open data standards resources, please proceed to the bottom of this document.

What makes open data truly 'open'?

The determination of whether data is truly open comes down to three categories:

Availability and Access

- The data must be available in a convenient and modifiable form.
- It can be linked to and easily shared with others.
- It has been provided in a standard, structured format so that it is machine-readable and can be easily manipulated.
- Guaranteed availability and consistency over time regarding the data itself as well as its accompanying metadata.
- The data can be traced back to where it originates.

Reuse and Redistribution

- Open data is data that is free to access, use and share.
- **Universal Participation:** The general public must be able to use, re-use and redistribute the data.

4.2.6 Best Practices for "Tidy Data"

Data cleaning

Data cleaning - also known as data wrangling - is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database.

Cleaning and reorganizing data before it is published on an open data portal is time-consuming but necessary. Not only is it one of the first steps that must be performed when publishing data, but it may end up being repeated many times as new problems come to light or new data is collected. Tidying data makes the sharing and publishing of data more efficient.

Tidying data makes the publishing, sharing and interoperability of data more efficient, and allows it to be read by more software platforms and operating systems than it otherwise would be.

By adhering to tidy data standards, data publishers can ensure that their data can be as accessible as possible to the widest number of users.

What makes tidy data 'tidy'?

Like families, tidy datasets are all alike but every messy dataset is messy in its own way.

Tidy datasets are:

- Easy to manipulate
- Easy to model

- Easy to visualize

Tidy datasets also have a *specific structure* that comes down to three simple rules:

- Each variable is a column.
- Each observation is a row.
- Each type of observational unit is a table.

The following example of student grades is a 'tidy' data table: "Date," "Student," and "Grade," are all variables, and have been given columns. The observations - corresponding to the dates of each test - have each been given a row.

Test Date	Student	Grade
March 16th	Smith, Sally	A
March 25th	Williams, Jane	B
April 1st	Smith, Sally	C
April 8th	Doe, Jonathan "Johnny"	A

What makes data *untidy*?

Anything that makes accessibility or visualization difficult can be considered untidy.

A table stored within a Word file is a good example of untidy data because the formatting-within-formatting keeps the data from being machine-readable.

Releasing personally identifying data about individuals or data violating local, national or international privacy laws is not only unethical, but also very much untidy.

4.2.7 How to create quality metadata

Checklist for quality metadata

To provide quality metadata to users, it is essential to include all of the descriptive information necessary to locate, understand, and use Dataset and its associated Resources.

Metadata should be written in a standard format according to the following best practices:

- **Write simply, completely, and consistently.** General users should be able to understand your metadata and, therefore, understand the Dataset.
- **Use taxonomies** Taxonomies limit terms to those that can be auto-filled and therefore ensure standardization. Within DKAN, Tags and Topics are two examples of taxonomies, also known as "controlled vocabularies."
- **Use specific and descriptive tags.** Tags are essential for users browsing though Datasets and Resources.
- **Provide an appropriate and descriptive title for your Dataset:** For example, the title "Geographic Distribution of City Parks Department Expenditures FY2016" gives users a pretty good idea of what the data is about.
- **Clearly state data limitations, if applicable:** One example would be noting usage constraints such as metadata for a map shapefile that advises "Not to be used for navigational purposes."
- **Avoid using special characters:** The following characters are examples that may affect your metadata's machine-readability: !, @, <, >, (,)
- **Review your metadata's accuracy:** Take a second look at your metadata, and perhaps even ask a colleague if they can look it over.

Most importantly, you may ask yourself:

- Could someone use an automatic search to locate this data set?

- Could they assess its usefulness?
- Do your metadata include enough specific information to uniquely identify and locate any geospatial data based solely on your documentation?
- Can a novice understand what you wrote?
- Does the documentation adequately present all the information needed to use or reuse the data represented?
- Are your key words descriptive enough to help other people find your data set?
- Have you used enough broad terms? Have you used enough narrow terms?

Guidelines For Releasing Data or Statistics In Spreadsheets

Source: www.cleansheet.org

Follow these simple guidelines to make your data or statistical releases as tidy and useful as possible.

1. Don't merge cells. Sorting and other manipulations people may want to apply to your data assume that each cell belongs to one row and column.
2. Don't mix data and metadata (e.g. date of release, name of author) in the same sheet.
3. The first row of a data sheet should contain column headers. None of these headers should be duplicates or blank. The column header should clearly indicate which units are used in that column, where this makes sense.
4. The remaining rows should contain data, one datum per row. Don't include aggregate statistics such as TOTAL or AVERAGE. You can put aggregate statistics in a separate sheet, if they are important.
5. Numbers in cells should just be numbers. Don't put commas in them, or stars after them, or anything else. If you need to add an annotation to some rows, use a separate column.
6. Use standard identifiers: e.g. identify countries using [ISO 3166](https://en.wikipedia.org/wiki/ISO_3166) codes rather than names.
7. Don't use only color or other stylistic cues to encode information. If you want to color cells according to their value, use conditional formatting.
8. Leave the cell blank if a value is not available.
9. If you provide pivot tables, make sure the underlying data is available separately too.
10. If you also want to create a human-friendly presentation of the data, do so by creating another sheet in the same workbook and referencing the appropriate cells in the data sheet.

More Data Cleaning Resources

https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html

https://vita.had.co.nz/papers/tidy-data.html

https://en.wikipedia.org/wiki/Data_cleansing

http://www.ats.ucla.edu/stat/sas/library/nesug99/ss123.pdf

https://betterevaluation.org/sites/default/files/data_cleaning.pdf

https://support.office.com/en-us/article/Top-ten-ways-to-clean-your-data-2844b620-677c-47a7-ac3e-c2e157d1db19

4.2.8 Open Data Standards Resources

Official governing bodies and non-profit organizations:

Project Open Data

https://project-open-data.cio.gov/

https://project-open-data.cio.gov/open-standards/

https://www.data.gov

Project Open Data is the United States' official policy regarding open data publishing and metadata standards. The homepage explains the mission and goals of Project Open Data and its participants, and the Open Standards page lists standards, specifications, and formats supporting open data objectives.

Links to best practices for data standards are also provided: International Standards, US National Standards, and US Federal Government Standards.

Gov.UK - Working With Open Standards

https://www.gov.uk/service-manual/technology/working-with-open-standards

The United Kingdom's official guide to working with open data standards, approved by the Open Standards Board at the Government Digital Service.

Code for America: Guide to Making Open Data Useful

http://archive.codeforamerica.org/our-work/data-formats/

Code for America builds open source civic-focused technology and has organized "a network of people dedicated to making government services simple, effective, and easy to use." This guide explains the necessity of open data formats in plain English - very useful for the less technically-inclined.

4.2.9 List of Civic Data Standards:

Johns Hopkins University's GovEx Labs

https://labs.centerforgov.org/open-data/civic-data-standards/

Johns Hopkins University's GovEx labs' defines a civic data standard as "an an open, collaboratively developed set of data schematics or semantics which facilitates interoperability between multiple providers and consumers for the public good."

This site provides a listing of civic data standards that GovEx labs have identified so far, with the disclaimer that they vary greatly in terms of active communities, current use, and approaches used for development.

The Open Contracting Partnership

Open Contracting Data Standard

https://www.open-contracting.org/data-standard/

http://standard.open-contracting.org/latest/en/

The Open Contracting Partnership is an organization composed of stakeholders and subject matter experts regarding government contracting. They "work across sectors and along the whole process of government contracting to use the power of open data to save governments money and time, deliver better goods and services for citizens, prevent corruption, and to create a better business environment for all."

The Open Contracting Data Standard (OCDS) "enables disclosure of data and documents at all stages of the contracting process by defining a common data model. It was created to support organisations to increase contracting transparency, and allow deeper analysis of contracting data by a wide range of users."

The Open Data Foundation

Academic Resources

http://www.opendatafoundation.org/?lvl1=resources&lvl2=papers

The Open Data Foundation, a U.S. based nonprofit, has provided an annotated bibliography of academic papers and other published works regarding open data publishing and the open data community.

The Open Data Institute

Open Data Maturity Model

https://theodi.org/guides/maturity-model (other guides: https://theodi.org/guides)

The Open Data Institute is a London-based independent non-profit dedicated to using open data to address today's global challenges. The Open Data Maturity Model is "a way to assess how well an organisation publishes and consumes open data, and identifies actions for improvement.

The model is based around five themes and five progress levels. Each theme represents a broad area of operations within an organisation. Each theme is broken into areas of activity, which can then be used to assess progress."

Open Knowledge International

The Open Data Handbook

https://opendatahandbook.org/guide/en/

Open Knowledge International is "a worldwide non-profit network of people passionate about openness, using advocacy, technology and training to unlock information and enable people to work with it to create and share knowledge."

This handbook discusses the legal, social and technical aspects of open data, as well as the why, what and how of open data – why to go open, what opening up your data means, and the how to 'open' data.

The Sunlight Foundation

Open Data Policy Guidelines

https://sunlightfoundation.com/opendataguidelines/

The Sunlight Foundation is "a nonpartisan nonprofit that advocates for open government globally and uses technology to make government more accountable to all." Their list of open data guidelines addresses what data should be public, how to make data public, and how to implement policy.

4.2.10 Additional Open Data Community Resources

5- Star Data Scoring

An independent project regarding the costs and benefits of open data:

http://5stardata.info/en/

"Tim Berners-Lee, the inventor of the Web and Linked Data initiator, suggested a 5-star deployment scheme for Open Data. Here, we give examples for each step of the stars and explain costs and benefits that come along with it."

Open Data Services

CoVE - Convert, Validate and Explore

http://cove.opendataservices.coop/ https://github.com/OpenDataServices/cove/

Open Data Services are a United Kingdom based cooperative that provides "technologies, support, and services relating to the production, standardization and use of [socially impactful] open data."

Their CoVE (Convert, Validate and Explore) web application currently supports two open data standards: The 360Giving Data Standard, and the Open Contracting Data Standard.

360Giving Data Standard

http://cove.opendataservices.coop/360/

"360Giving provides support for grantmakers to publish their grants data openly, to understand their data, and to use the data to create online tools that make grant-making more effective."

The Open Contracting Data Standard

http://standard.open-contracting.org/validator/

"The Open Contracting Data Standard promotes the effective use of contracting data, helping users to 'follow the money', and it provides a clear template for governments wishing to disclose their data."

The Open Data Substrate

GitHub project to catalog open data standardization efforts

https://open-data-standards.github.io/

"The goal of this industry organization is to provide a forum to quickly establish standards so that Open Data implementors can ensure they are developing Open Data solutions that interoperate. This community-driven initiative to promote vendor interoperability and data portability is derived from a fundamental belief that 'data is more useful' when 'more people can use it'."

4.2.11 The 8 Principles of Open Government Data

Democratic guidelines regarding open data best practices

For more information, see: https://opengovdata.org/

Background

"On December 7-8, 2007, thirty open government advocates gathered in Sebastopol, California and wrote a set of eight principles of open government data.

This page annotates the original 8 principles and links to additional principles found around the web."

The Original 8 Principles of Open Government Data [Source]:

> **"Government data shall be considered open if it is made public in a way that complies with the principles below:"**

1. **Complete** All public data is made available. **Public data** is data that is not subject to valid privacy, security or privilege limitations.
2. **Primary** Data is as collected at the source, with the highest possible level of granularity, not in aggregate or modified forms.
3. **Timely** Data is made available as quickly as necessary to preserve the value of the data.
4. **Accessible** Data is available to the widest range of users for the widest range of purposes.
5. **Machine processable** Data is reasonably structured to allow automated processing.
6. **Non-discriminatory** Data is available to anyone, with no requirement of registration.
7. **Non-proprietary** Data is available in a format over which no entity has exclusive control.
8. **License-free** Data is not subject to any copyright, patent, trademark or trade secret regulation. Reasonable privacy, security and privilege restrictions may be allowed.

4.2.12 Relevant Reading Materials

"Enabling Civic Data Standards" - by Andrew Nicklin, Trainer of Chief Data Officers at the Johns Hopkins University's Center for Government Excellence GovEx Labs. https://medium.com/@technickle/enabling-civic-data-standards-e38b0c40e3a#.hm37j1c5r

"Open Data Standards" W3C blog post: https://www.linkedin.com/pulse/open-data-standards-steven-adler

4.3 Professional Help

For professional DKAN development services and support:

- CivicActions
- Angry Cactus
- National Democratic Institute
- Annai
- Solucionex

4.3.1 Add your company

If you provide DKAN development and/or support services, here's how to add your company to business directory:

- Create a /dkan page on your website (ex: https://civicactions.com/dkan/)
- Create a [new issue](https://github.com/GetDKAN/dkan/issues/new) and include logo, company name, link to /dkan page, brief description of services with respect to DKAN

Someone on the DKAN team will review your request and add your company if it's appropriate for the DKAN community.

4.4 Code of conduct

4.4.1 Purpose

A primary goal of the Project DKAN is to be inclusive to the largest number of participants, with the most varied and diverse backgrounds possible. As such, we are committed to providing a friendly, safe and welcoming environment for all, regardless of gender, sexual orientation, ability, ethnicity, socioeconomic status, and religion (or lack thereof).

This code of conduct outlines our expectations for all those who participate in our community, as well as the consequences for unacceptable behavior.

We invite all those who participate in Project DKAN activities to help us create safe and positive experiences for everyone.

4.4.2 Open source & culture citizenship

A supplemental goal of this Code of Conduct is to increase open source and culture citizenship by encouraging participants to recognize and strengthen the relationships between our actions and their effects on our community.

Communities mirror the societies in which they exist and positive action is essential to counteract the many forms of inequality and abuses of power that exist in society.

If you see someone who is making an extra effort to ensure our community is welcoming, friendly, and encourages all participants to contribute to the fullest extent, please recognize their efforts.

4.4.3 Expected Behavior

The following behaviors are expected and requested of all community members:

- Participate in an authentic and active way. In doing so, you contribute to the health and longevity of this community.
- Exercise consideration and respect in your speech and actions.
- Attempt collaboration before conflict.
- Refrain from demeaning, discriminatory, or harassing behavior and speech.
- Be mindful of your surroundings and of your fellow participants. Alert community leaders if you notice a dangerous situation, someone in distress, or violations of this Code of Conduct, even if they seem inconsequential.
- Keep conversations in appropriate channels to help people find the information they need.

4.4.4 Unacceptable behavior

The following behaviors are considered harassment and are unacceptable within our community:

- Violence, threats of violence or violent language directed against another person.
- Sexist, racist, homophobic, transphobic, ableist or otherwise discriminatory jokes and language.
- Posting or displaying sexually explicit or violent material.
- Posting or threatening to post other people's personally identifying information ("doxing").
- Personal insults, particularly those related to gender, sexual orientation, race, religion, or disability.
- Inappropriate photography or recording.
- Unwelcome sexual attention. This includes, sexualized comments or jokes and unwelcomed sexual advances.
- Deliberate intimidation, stalking or following (online or in person).
- Advocating for, or encouraging, any of the above behavior.
- Promoting your particular product or service (though we of course welcome relevant recommendations).

4.4.5 Consequences of unacceptable behavior

Unacceptable behavior from any community member, including sponsors and those with decision-making authority, will not be tolerated.

Anyone asked to stop unacceptable behavior is expected to comply immediately. If a community member engages in unacceptable behavior, we may take any action deemed appropriate, up to and including a temporary ban or permanent expulsion from the community without warning.

4.4.6 Reporting guidelines

If you are subject to or witness unacceptable behavior, or have any other concerns, please notify us as soon as possible by emailing getdkan@gmail.com.

4.4.7 Addressing grievances

Only permanent resolutions (such as bans) may be appealed. To appeal a decision of the working group, contact the Project DKAN team at getdkan@gmail.com with your appeal and the team will review the case.

4.4.8 Scope

We expect all community participants (contributors, paid or otherwise; sponsors; and other guests) to abide by this Code of Conduct in all community venues–online and in-person–as well as in all one-on-one communications pertaining to community business.

This code of conduct and its related procedures also applies to unacceptable behavior occurring outside the scope of community activities when such behavior has the potential to adversely affect the safety and well-being of community members.

4.4.9 Contact info

Project DKAN, getdkan@gmail.com

4.4.10 License and attribution

This Code of Conduct is distributed under a Creative Commons Attribution-ShareAlike license. Portions of text derived from the Django Code of Conduct and the Geek Feminism Anti-Harassment Policy.

CHAPTER 5

User Guide

Warning: We are in the process of migrating our legacy, user oriented documentation to this site. There are some new changes to DKAN that have not yet been updated in this documentation, so there may be some inconsistencies with how DKAN works in the latest version. There may also be some formatting issues that still need to be corrected as we transfer the documentation from the previous system. Thanks for your patience!

The Site Manager role is the highest non-technical role available on DKAN sites. Users assigned to this role need a good understanding of how DKAN works, and what administrative actions may be necessary to support the site and its users.

I'm a Site Manager, what do I need to know?

The role of *Site Manager* is broad. You'll have access many parts of the site and will need to know what's happening at a high level, but typically a Site Manager isn't dealing with the finer details of content management.

Other roles, like *Editors* and *Content Creators*, focus on maintaining high quality data and narrative content that both follow standards as well as engage site visitors. Get more information about different user roles in the *People section* of this playbook.

As a Site Manager, you build the framework that other roles operate within and build on. With that structure in place, you can focus on the overall experience site visitors have as they navigate your Open Data site. A Site Manager's objective is to make open data meaningful by reaching citizens and connecting them to the right data at the right time.

What does a Site Manager handle?

Data and Content One of the biggest pieces of a DKAN site is of course the data itself. A Site Manager can decide how users can add data and create content, and use special visual tools to showcase the stories and insights gained from datasets.

People Data and content management can't be done alone. Site Managers handle all the users on the site and decide who has access to what. They can also create Groups for organizations, which have *even more possibilities for roles and permissions*. Build your team to build your site.

Structure Creating new Pages, DKAN Topics, creating Groups, etc. are all forms of structure on a DKAN site. Use these structure types to create a framework for other users to operate within and build upon.

Appearance You can make your DKAN site look and feel completely aligned with your organization to give site visitors the best possible experience while going through the site. Add your logo, change page layouts, customize the Home page, or pick out the right fonts to match your brand.

5.1 Account Access and Setup

5.1.1 Getting to the site

Your DKAN Open Data site will have a login page located at the site's URL with the ending /user. Once you've accessed the login page, you can log in with your account information. We suggest bookmarking the page for quick access.

5.1.2 Getting your login information

For your account details, you may receive an automated email with a link and further instructions for login or simply be contacted directly by the person managing your site. Once you receive your account information, you can access the site, log in, and begin setting up your profile.

Access the site and log in/log out

Get to your DKAN site and log in by either:

- Finding your DKAN site and typing /user in the web address bar after the site name.
- Following the link sent to you with your login information.

If you don't remember or can't find your password you can get a new password by clicking the Request new password button.

Use the login information provided to access the site and your profile. You'll see your most recent content, your profile details, and your profile image.

When you're done adding and editing content on the site, you'll want to make sure you log out to avoid security risks. You can log out from anywhere on the site by using either of the buttons shown in the picture below. The blue navigation bar is not on every page of the site, but the black administration bar (the Admin Menu) always appears.

Once you log out, you'll be taken back to the Home page of your DKAN site as if you were a general site visitor.

5.1.3 Setting up your profile

Once logged in, you'll see your profile page. By clicking the Edit button above your profile image, you can change the details of your account including:

- Your Username.
- The email address linked to your account and how you will be contacted.
- Your password (you can also request a new password if you've lost or forgotten it).
- Additional information about you in the About section and your timezone.
- Your profile image.
- You can also see which Groups you belong to and delete your account by clicking the Cancel account button.

5.2 Adding New Content

Your DKAN site is an access point for users to connect with open data, but data isn't the only content type that you can add. There are a variety of content types at your disposal, and it's up to you to choose which best suits your needs.

5.2.1 Content Types in DKAN:

Dataset

Datasets are used to group related pieces of data and their associated metadata. These pieces of data - known as Resources - can then be found under a single url along with descriptions and licensing information.

You can think of Datasets as 'folders' for the Resources within. In real life, a typical folder could contain a printed spreadsheet, a PDF of a map, and a CD-ROM within it. In DKAN, you can similarly add a wide number of Resources in various formats to a single Dataset.

Datasets are assigned to Groups, which helps sort Datasets according to a common publisher. For more on Groups, see below.

Admins, Site Managers, Editors and Content Creators can all add Datasets.

Resource

In DKAN, Resources are the files, APIs or links that are being shared within a Dataset.

Resource types include csv, html, xls, json, xlsx, doc, docx, rdf, txt, jpg, png, gif, tiff, pdf, odf, ods, odt, tsv, geojson and xml files.

If the Resource is an API, it can be used as a live source of information for building a site or application.

Resources can be previewed within your browser in the form of a grid, graph or map using DKAN Data Previews or used to create visualizations.

Admins, Site Managers, Editors and Content Creators can all add Resources.

Group

Groups allow you to group together Datasets under a organization (i.e. Department of Transportation, Department of Education) in order to make it easier for users to browse Datasets by theme.

As a best practice, Datasets and Resources that are added to a Group should share a common publisher.

Groups also enable you to assign roles and permissions to members of the group. By adding users to the Group's membership roster, you can create and manage a community based around the data within the group.

Groups may be added by Admins and Site Managers, but not by Editors or Content Creators.

Page

Pages are used for static content, such as an "About" or "FAQ" page. They may be added by Admins, Site Managers and Editors, but not by Content Creators.

Data Story

Data Stories are used for creating blog-like content pertaining to data over time. A real life example would be monthly Data Stories created by the Parks and Recreation Department to use charts and maps to update site visitors on a local tree-planting campaign.

Much like Data Dashboards, Data Stories offer a wide range of possible layouts and can be curated with data visualizations like charts, maps and tables.

Admins, Site Managers, Editors and Content Creators can all add Data Stories.

Data Dashboard

The Data Dashboards feature allows you to create pages that bring together various forms of content about a specific subject all in one place.

A real-life example of a data dashboard can be found on any major weather site when looking up the charts, maps and tables from a specific zip code.

Admins, Site Managers, Editors and Content Creators can all add Data Dashboards.

Harvest Source

A Harvest Source is a content type that allows administrators or Site Managers to "harvest" data from outside sources. To "harvest" data is to use the public feed or API of another data portal to import items from that portal's catalog into your own.

Harvested datasets are fully imported from an external source onto your DKAN site. This means that the datasets exist both on the external source and your site - independently from one another, and with all of the same information (including title, metadata, tags, and so on.)

By importing datasets from external sources, you can provide more open data on your site without manually managing the dataset content itself. Site visitors will see that a dataset was harvested and its original source, promoting visibility across agencies and sectors.

Admins and Site Managers can add Harvest Sources, but not Editors or Content Creators.

Visualization

Visualizations provide intuitive, understandable displays of data. Visualizing information fulfills a basic need; the need to tell a story.

Visualizations allow users to understand the data from a new perspective, and can be viewed within the site or embedded elsewhere without having to download the data itself. This not only saves time, but also saves memory, bandwidth and hard drive storage space.

Currently, Charts are the only Visualization type included within out-of-the-box DKAN, but Maps are underway in the future.

Admins, Site Managers, and Editors can create Visualizations, but not Content Creators.

5.2.2 Where do I add new content?

The Add Content menu

The most direct path for creating content is through the **Add content** tab on the Admin Menu toolbar. Select the content type you'd like to add, then complete the fields shown in the following form.

The Content Page

In some cases, you may want to see what content exists before creating something new from scratch. As your site grows, it's helpful to check what already exists before starting something new.

You can click **Content** from the admin menu to access the Content page and view all existing content, publish or unpublish content, and add new content from scratch.

In the example below, the user is adding a new Dataset with Tags, Topics, and other metadata associated with Datasets.

Though not all fields in the Dataset creation form are required, it's best to include as much information as possible. This way, users and the general public will have the proper metadata.

The final step in creating a piece of content is using the submenu at the bottom of the form to add the administrative information to the content.

Use this menu to add information about:

URL path settings: By default, the "generate automatic URL alias" option is selected because DKAN is designed to provide the most intuitive and SEO-optimized URL path. To create your own path, uncheck the box and type in a new alias.

Revision information: This information is important for content that has been added and then edited. By checking the "create new revision" checkbox, DKAN will create a revision of the content.

Revisions can be helpful if a change is made and published and later you want revert to an old revision. Include notes about the changes in the Revision log message text box to give other users context.

Authoring information: In DKAN, content must be assigned to an author and given an authoring date so that it can be managed later even if the author is the Anonymous user. An author must be a user on your DKAN site, and this is helpful in knowing which users have added specific pieces of content. By default, DKAN assigns the author to the user who is logged in when the content is added.

If you leave the "authored on" field blank, DKAN will use the date and time of when the content was originally saved.

Publishing options: Authenticated users who have created or have the right to modify site content can publish and unpublish content. Published content is saved and visible on your DKAN site. Unpublished content is saved, but it is not visible on your DKAN site.

Why unpublish content? The "unpublish" option is helpful for when content should be saved as a Draft, or archived but not deleted. Deleting content permanently erases it, so the "unpublish" option allows you to retain content that is no longer published live.

Groups: While Datasets are the "folders" for data resources, Groups are the container for those folders. Resources cannot be added themselves to a Group, they must be stored within a Dataset to be associated with a Group.

For example, the Dataset "City Bike Lanes" would be associated with the group "Department of Transportation"

Datasets can be added to multiple Groups. All users must belong to at least one Group to have the Groups menu item available to them.

When a Dataset is added to a Group, it will be included on the Group's home page and may be edited by the Administrator members of a Group. As a best practice, users should only add Datasets to Groups they are a member of.

Viewing content you've already uploaded

To see a list of content that you've already uploaded, you can navigate to your user profile page by clicking on your username on the right-hand side of the screen.

Once you're on your user profile page, you'll see a list of datasets, resources and other content types that were uploaded by your account.

Viewing a listing of the site's files

From the admin menu, click "Content." Once you're on the Content page, the "Files" button will take you to the files listing.

5.2.3 Adding a Resource

There are three ways to import a resource to your DKAN open data catalog:

1. **Upload a file:** Select the file from your hard drive or local network, then click "Upload."
2. **API or website URL:** Provide the link to an outside API or website URL to add it to your DKAN data catalog as a resource.
3. **Remote file:** You can import a file hosted elsewhere on the Internet, provided it is in a format supported by DKAN. (File formats are listed on the "Add Resource" screen.)

CSV files, once added as a Resource, can then be imported to the DKAN Datastore for efficient storage and access. *(recommended)*.

Steps to upload a new resource

1. From the **Admin Menu**, hover over the **Add Content** menu link.
2. On the drop-down menu, select the **Resource** option.
3. Choose a method for uploading the Resource (upload, API/website, or remote).
4. Enable Data Previews such as Grid, Graph or Map (optional). Also, select the appropriate delimiter, and choose whether your Data Previews should have an option to be embedded elsewhere.
5. Give the Resource a title (required), a description (optional), and add to a Dataset (optional).
6. Click the Save or Save and add another button at the end of the form to add the Resource.

It's possible to add a Resource as a stand alone piece of content, but it isn't recommended. As a best practice, and for optimal site organization, all Resources should be added to Datasets.

Otherwise, Resources on their own do not contain metadata – which is why it's important to assign them to Datasets, which do have metadata. Resources also inherit the group association of the dataset they belong to.

When adding a new Resource, DKAN provides a list of existing Datasets and you can select the appropriate one. If none of the existing Datasets are a good fit, then consider first adding a new Dataset. You can also add a Resource and then assign it to a Dataset at a later time.

Enabling Data Previews

As site visitors browse through a Dataset and its Resources, they can click the Explore Data button to preview a specific Resource.

DKAN's built in Data Preview tools give site visitors the option to see a basic visual of a Resource as a map, chart, or grid.

Users uploading Resources can enable DKAN Data Previews on JSON, geoJSON, XML, ArcGIS REST, WMS, image, PDF, and ZIP files. The options are provided as three checkboxes during the process of uploading a Resource.

Data Previews only display if the contents of the Resource match the data format for a preview. For example, if the map preview is enabled but the Resource doesn't have latitudinal/longitudinal or GeoJSON data, then the Preview page will be blank.

Fig. 5.1: This image displays where you can enable grid, graph and/or map previews for a Resource.

Data Preview Types:

Grids and Graphs: This type of Data Preview works well with tabular data like CSV or XLS files. Grids appear most similar to a spreadsheet.

Graph previews allow site visitors to select the values from the data that may be plotted as a column, bar, line or point graph.

Maps: If the resource contains a Latitude column and a Longitude column (in decimal values - see http://www.earthpoint.us/Convert.aspx), then each record from the data will be displayed as a point on the map. Site visitors can click each point to learn about the data.

External Previews:

Need to use advanced data visualizations from an outside source? Use DKAN External Previews.

ArcGIS, CartoDB, Infogram and Tableau, as well as other outside data viz tools can all be embedded within DKAN Data Stories and Data Dashboards, as well as on Pages.

Additionally, External Previews can be used for ArcGIS and Carto previews of data resources - directly on the resource's page.

Once External Previews are enabled, site visitors can view a Resource and click the Open With button to visualize the contents of the Resource with an External Preview.

Choose which visualization tool is best for previewing a Resource based on its data format.

For Admins: How to enable External Previews:

1. On the Admin Menu, hover over the **DKAN** menu link until the drop-down menu appears.
2. Select **Data Previews.**
3. From the Data Previews page, scroll down to the section titled External Preview Settings.
4. In the External Preview Settings section, check the box for the External Previews you want to make available for viewing a Resource.
5. If a data format is not listed in this section, you can add the data format to the list of available formats for Resources.

There are two types of External Previews that may be enabled by Site Managers: **CartoDB** and **ArcGIS**.

Carto Preview: CartoDB is an open source platform with a powerful datastore that allows users to create their own maps using Carto Builder.

Supported formats: CSV, Excel, GeoJSON, KML, OpenXML, XLS

ArcGIS Preview: ESRI ArcGIS can be used to create multi-dimensional maps (such as the topography of a mountain range, or the flow of a watershed) and does not limit the amount of layers you can add to your map.

ArcGIS Previews require a URL in the resource API field and will not work with Resource files.

Supported formats: ArcGIS endpoints, SHP files (shapefiles)

5.2.4 Adding a Dataset

Remember to include as much metadata (the who, what, when, where and why of data) as possible in order to give site visitors context.

Add a Dataset:

1. From the **Admin Menu**, hover over the **Add Content** menu link and select **Dataset**.
2. Add a title, description, Tags, contact information and public access level (required). Optionally, Datasets may be added to Groups and assigned Topics.
3. Add a license to clarify reuse limitations.
4. Click the **Next: Add data** button to add at least one Resource.
5. Follow the steps for adding a Resource.
6. Click the Save button to finalize the addition.

Fig. 5.2: This animated screencap displays the process of editing and saving a dataset.

Directly under the title of the Dataset, you may change the URL path for your dataset in the dataset/ field. Note that the title and URL path are not linked. That means that you can change the title without affecting the URL path and vice versa.

Adding Metadata

Metadata is the "Who, What, When, Where, Why" of data. Metadata contains the attributes that describe each Dataset. Examples include the name of the Dataset's author, the title of the Dataset, the date that it was last updated, any relevant contact information associated with the Dataset, and more.

When viewing a Dataset, you can scroll down the page to the "Dataset Info" section to view its metadata.

In addition to providing important context, metadata makes the data published machine-readable. That means that programmers, analysts and other technical users can use the information for their own purposes.

Though most metadata fields are not required, adding more detail to your metadata will make for more usable datasets. In some cases, extra metadata fields are required to be compliant with certain standards and initiatives.

The fields included in the Additional Info screen are the metadata for the Dataset and are compatible with DCAT, an RDF vocabulary designed to facilitate interoperability between data catalogs published on the web. These fields are also compatible with the Common Core metadata schema from Project Open Data.

Site Managers can select to make Project Open Data and DCAT fields required for publishing a Dataset by enabling POD and/or DCAT validation.

When viewing a Dataset, scroll down the page to the Dataset Info section to view its metadata.

Adding more relevant information: In the image below, you can see a section titled Resources and below that Related Content. In the Resources section you can choose from existing Resources to pull into the Dataset. You can even choose the order Resources appear in by dragging the individual rows up and down. Click the Add another item to add as many Resources as you want to the Dataset.

Scroll to the Related Content section to add links to other content that site visitors should see. This is a great way to link to your Data Stories, Charts, and Dashboards (or external links) that showcase the impact that data can have on the daily lives of citizens.

Below is a Dataset that has been filled out completely with a description, metadata, assigned to a Group and includes related content.

5.2.5 Visualizations

Visualizations take Resources on your DKAN site and generate visual representations to make data understandable and accessible. DKAN offers several built-in tools for making data visualizations easy. These were designed with ease of use and flexibility in mind.

A Chart is the means, but the end must be defined by the citizen need. What is important for the site visitor to know about the data? What can we learn by comparing the different information contained in a single Resource? Once a Chart is added you can feature it to support the narrative of a Data Story or complete a Data Dashboard.

While this tool is incredibly powerful, it also includes more variables that depend on one another. As a Site Manager, you have access to create Visualizations on DKAN. This type of content is unique to Site Managers and Editors, and as a Site Manager you have access to manage all content regardless of the author.

Adding Charts

In general, you'll add DKAN Charts for your visualizations. Charts are a powerful tool for taking data and making it meaningful to the average site visitor who may have little to no experience with data and analysis. Charts offer power and flexibility to represent exactly what you're looking for with minimal effort and no specific technical training required. Data that power charts can come directly from your DKAN data catalog or alternatively any URL, public Google spreadsheet, or data proxy/API.

Charts are ideal for showing comparative and/or historical information. Site visitors can look at a Chart and quickly discern the relationship between several data points. Charts easily adapt to represent a number of combinations between many values. Visualizations may range from a simple 2-dimensional comparison to more complex, multi-faceted relationships.

Supported data and file types:

- **Using internal CSV files:** Charts visualize data that has its contents organized into rows and columns (tabular data). DKAN Charts support CSV files when selecting an internal Resource hosted on DKAN. Select the CSV option for the back-end when loading the data source.
- **External CSV and XLS files:** You can create a Chart from files hosted elsewhere on the Web as long as a link is provided. Linked files can be a CSV or XLS. When files are externally linked select the DataProxy option for the backend when loading the data source.
- **Using Google spreadsheets:** Public Google spreadsheets are files created with Google sheets that have been published to the web. You can create your Chart with the public link and by selecting the Google spreadsheet option for the back-end when loading the data source.

Choosing your data: The first step in adding a Chart is choosing which data you want to visualize. Choose a title and add a description, then select the data source. You have a 3 options for selecting the data source:

- **Upload a new file:** This is a file stored locally (ie a file on your computer's hard drive) and not already on your DKAN site. Uploading a file to power your Chart does not automatically add the file as a Resource on your DKAN site. Use the Upload button in the File field to choose a file from your computer. Note file size and type limits apply.
- **Choose an existing Resource:** Select a Resource that has been added to your DKAN site. Start typing in the Existing Resource field and DKAN will autocomplete with matching Resources.
- **Link to an external file:** Use the Source field to link to a file hosted elsewhere on the web.

Choose a data processor: Once you select the data source, it's important to choose the right data back-end to process the data. The processor reads the contents of a file and makes it possible to define which variables should be visualized. This works in the background, but you should know which data sources match which data back-ends. There are 3 data back-end to choose from:

- **CSV:** CSV is the default selection, and it is used for Charts powered by internal data sources. If you upload a new file or select an existing Resource as your data source then your data back-end is CSV.
- **DataProxy:** If you use an external link for the data source, you may use a CSV or XLS file type. An external link is the only way to power a Chart with an XLS file. If you select a data source by using an external link then your data back-end is DataProxy.
- **Google Spreadsheet:** You can power a Chart with a Google spreadsheet if the document has been published to the Web and made public. If you select the public link to a Google spreadsheet then Google Spreadsheet is your data back-end.

Defining your Chart variables: In essence, Chart variables are the two axes of your Chart that you set. The x-axis and the y-axis each have their own set of values that run along each respective axis. Because Resources often contain more than two columns (all with their own set of values), you can choose which columns you want as the x- and y-axis as well as add Series. Series can be selected from the different columns within your Resource to compare multiple columns along the Chart axes. This provides flexibility when using large files to create Charts.

You can choose which contents within the data source to display on your Chart. Some data sources may be fairly simple with only a couple columns while others may contain dozens. Options for the variables are based on the contents of the data source selected to power your DKAN Chart, so you'll choose from columns and their values. There are 3 variables to select for when adding your Chart:

- **Series.** Series show the values within a column as the y-axis values mapped along the X-Field values. Once you choose a column to provide the values for the X-Field, Series provide the corresponding y-values. You can choose multiple columns from your Resource to be Series, which can be helpful for showing multiple data points next to one another.
- **X-Field.** The X-Field provides the x-axis values for your Chart. Choose a column from your Resource to populate the X-Field with values.
- **Data Format:** Selecting the correct data format helps Charts to display correctly. Choose the format that matches the format of the values in your X-Field. If you're not sure, you can leave the selection on Auto and DKAN will make the best selection. If the values are text/non-numeric, select the String format.

Choose a Chart type: Different types of data work better with certain Chart types more than others. DKAN offers a number of different Chart types like line graphs, bar charts, and pie charts and different types of data will work better as a line graph rather than a bar chart.

For continuous data (like time) use a line Chart to show the movement of the data. For categorical data (like a discrete totals within a category) use bar charts, and for data that totals a sum use a pie chart.

There are a number of Chart types to best display your data depending on what you want the Chart to show and the contents of your Resource. You can choose a Chart type and then move to the Preview and Adjust screen to make the final modifications to your Chart. You can always change the Chart type by using the Back button, so that you can test and see which Chart type works best with your data.

In the example below, the Site Manager is adding a Chart that uses an existing Resource. By typing, DKAN suggests an autocomplete option and the Site Manager selects the Resource. Once the Resource is selected, the Site Manager can define the variables of the Chart. In this example, the Resource is very basic with only two columns that be chosen from, but more robust Resources could have several columns to choose from.

Adjusting your Chart settings: After the data is loaded and the variables selected, you can see how your Chart will appear and make adjustments so that your visualization best depicts the meaning of the data. On the Preview and Adjust screen, you make any final modifications to your Chart through a number of options on the Chart Configuration menu. The Chart preview will adjust in real-time to show you what the Chart will look like on your site. Use the preview to test out different adjustments for your Chart settings.

In the example below, a Site Manager is adjusting the Chart settings for a Chart they're adding. Though there are a number of options, the data here is fairly basic. The Site Manager rotates the labels by putting in a degree of rotation in the X Label Rotation field, changes the color of bars by adding a hex value in the Color field, and adds a label to the x-axis by putting a name in the X Axis Label field.

As the example continues below, the Site Manager decides to show the title of the Chart and selects the Show Legend option. Show Tooltips and Reduce Ticks are selected by default. Click on the Finish button at the bottom of the page to finalize your selections and see the final results of how the Chart will appear on your DKAN site.

Unlike other content types, Charts don't automatically collect on a page on your DKAN site. You can make Charts visible by including them in Dashboards and Data Stories.

Key information when adjusting your Chart settings:

- **Query Editor:** The Query Editor field lets you search the contents of the Resource powering your Chart and visualize the most relevant pieces. This function is useful for especially large Datasets. Use this setting to perform a complex search on the data in your Resource and narrow the focus to display on your Chart. It's good for highlighting key insights in the data. Use the same format conventions as in the Resource (ie $0.00, x/y/z) when performing the search.
- **Filter Editor:** Terms add a broad filter to highlight characteristics shared by multiple data points in your Resource. This is adds more focus than visualizing all the contents of a Resource, but is not very overly complex. Use this to draw specific comparisons in your visualization. Add multiple filters to give a specific cross-section within the data.
  - **Field:** Create a term to filter the data by first choosing a Field from a column within the Resource. All the columns will appear in a drop-down menu to choose from. Use terms to narrow the view of the data.
  - **Filter Type:** Choose from the drop-down list to further specify conditions for the data you're looking for within the Field you've already selected.
- **X-axis Chart Settings:** These settings are specific to the x-axis:

X-Format: Choosing the X-Format lets you specify how the x-axis values are represented rather than as the basic numbers. For example, the value 5.2 will show as $5.20 if the X-Format is $0.00.

- **X Label Rotation:** Use this to rotate the values of the x-axis of your Chart. With 0 degrees rotation, the labels appear side by side. Enter a number to add a degree of rotation and the labels will appear at an angle.
- **Step:** Set the number of increments that will appear on the x-axis. The total distance on the x-axis from the 0 value to the final value will be divided into the number of increments set. By default, the Step is not set.
- **Tick Values:** Set a range of values from your Resource to narrow which values appear on your Chart. By default, every value in the Resource is displayed.
- **X Axis Label:** This is the name that describes the x-axis and appears on your DKAN Chart below the x-axis. Create a label to provide more context for the data being visualized.
- **Y-axis Chart Settings:** These settings are specific to the y-axis.
  - **Format:** Choosing the Format lets you specify how the y-axis values are represented rather than as basic numbers. For example, the value 5.2 will show as $5.20 if the Format is $0.00.
  - **Y Axis Label:** This is the name that describes the y-axis and appears on your DKAN Chart below the y-axis. Create a label to provide more context for the data being visualized.
  - **Distance:** The distance of the Y Axis Label from the left edge of the page. The larger the number, the closer the label appears to the y-axis of your Chart.
- **General Chart Settings:**
  - **Margin:** Margins add padding (extra white space) around your Chart, measured in pixels. Padding is added to the top, right, bottom and left respectively. Adjust the padding to accommodate long labels, Chart values, label rotations, etc.
  - **Transition Time:** Change the time it takes to animate the data in a Chart. Longer transition time will make the sections of a Chart appear more slowly. Note: this does not affect pie charts.
  - **Color:** Change the color of the segments of your Chart by adding color names (blue, green, etc.) or the hexadecimal numbers of specific hues (#FFD9AA , #FFFFFF). You can also use the color selection tool to visually select a color rather than by typing it in. You can add any number of different colors for the Chart segments by adding commas in between colors.
  - **Sort:** Choose which criteria the Chart sorts data by and displays on the graph, like A-Z or highest to lowest. Criteria could be values from the Chart variables or left to the default sort setting.
  - **Goal:** This setting creates a line at the value you designate on the Chart. It signifies a baseline, an average, or a goal among the values to compare the rest of the data. Enter a value in the Goal field to select the value to appear parallel to the x-axis. You can also choose the color of the line, whether you want to show the label (the label is "Target" and cannot be changed), and if the label should appear directly on the chart or outside of it.
- **Checkboxes:**
  - **Show title:** A Chart must be titled when it is created. By checking this box, you can display that title as a header on the Chart.
  - **Show controls:** Select the Show controls option to make your Chart interactive. On bar charts, you can include buttons for site visitors to choose how data is displayed on the Chart either as Grouped and Stacked. Check this box to show buttons that show data either as a single stack composed of all the Series (Stacked) or the data are grouped together but have discrete bars (Grouped).
  - **Show legend:** When selected, this shows site visitors the names of the Series included in your Chart. Site visitors can show and hide Series on the Chart when Show Legend is checked.
  - **Group by X-Field:** With non-numerical discrete data (usually text), you may have repeated x-values on your x-axis. Check this box to add the outputs together and display as a single x-value on your Chart.

- **Show Tooltips:** Check this box so that site visitors can mouse over the individual sections of your Chart and see exact values. If this box is checked, you won't also need Show Values, which creates a fixed label for each value.
- **Reduce Ticks:** In a value range, you may not need display every value (for example, 1-1000). Check this box to group values by increments to reduce the number of x-axis values shown on the x-axis.
- **Stagger Labels:** Staggering places labels slightly above and below each other rather than on the same line, so that they don't overlap. Check this box if your labels don't appear correctly.
- **Show Values:** Show exact values on your Chart with a fixed label. If this box is checked, you won't also need Show Tooltips (which creates hover text with values).
- **Show Data Points:** This option only applies to the line chart type. Check the Show Data Points option to add a dot on the line Chart for every unique data point in the Resource.
- **Donut:** This option only applies to the pie chart type. Select the Donut checkbox to change the aesthetic of your pie chart to look like a donut shape. This adds some variety and visual flexibility to the standard pie chart type.

Going back to change Chart selections: To make changes on any of the previous screens, use the Back button rather than the key on your keyboard or back tab in your browser. By moving back without using the Back button, you may lose all your work or encounter other errors.

5.2.6 Adding a Data Story

Similar to a blog post, Data Stories provide a narrative that adds the depth of impact. Stories focus on how data changes real lives every day. While the form might look familiar, it's helpful to know how the content will appear on DKAN.

1. Log in to your DKAN site.
2. From the Admin Menu, hover over the **Add Content** link
3. Select the **Data Story** menu item from the drop-down menu.
4. Title the Data Story and provide a banner image
5. Add Tags and Topics to make the content easy to find.
6. Choose a layout for the Data Story. By default, the most basic layout is selected.
7. Click the Save button to create the content.

Once the Data Story is added, the content may be altered, rearranged or new content added using the In-place Editor. Learn more about how to use the In-place Editor.

Key information when adding a Data Story:

- **Image:** Choose a large, high quality image for your Data Story. This image appears in a large format across the top of the Data Story. Because of the size, you'll need a large image (minimum 900x1200 pixels) with high resolution so that it appears as expected. In Data Stories, these images can only be uploaded; there isn't an option to link directly to an image from the web. First select the image by clicking on the Choose file button and then add the image by clicking the Upload button.
- **Edit summary:** Click the Edit summary link to open another text box. In the Summary text box, you can add unique details about your Data Story. This text appears as teaser text as site visitors browse through the Stories page. If you don't want to write additional summary text, DKAN will simply pull the first portion of your Data

Story in the Body text (about 100 words). Including a summary can be useful in adding more key search terms or using a different tone to intrigue site visitors to learn more.

- **Body:** This is the section where the contents of your Data Story appear. Because DKAN doesn't automatically save content and publishes directly to the site once you save, we recommend drafting in a separate text editor so that you can write at your own pace and use your own review process before pasting into the Body section of your Data Story.
- **Text editor options:** Use the Body text box for the contents of your Data Story. Use the tools in the text editor to format and style the body of your text. With these tools you can add images, links, quotes, and line breaks directly in the text box.

Adding Tags and Topics: You can add Tags and Topics to your Data Story so that it's easy to find in a search and as site visitors browse the content on your DKAN site. Tags are free-form, so they can be newly added in the field and can contain any words.

Think of Tags as keywords either within or related to the content. So if you have a Data Story about chickenpox vaccines in the state of Mississippi you might include a Tag for "chickenpox", "vaccines", "Mississippi" and additionally "public health" and "viruses". By including Tags on your Data Story, the Data Story associated with those terms will appear when the terms are included in a search.

Topics are similar but distinct from Tags. Topics are preset and they act more as a category that content is collected under on your DKAN site. Topics aren't limited to a common data publisher or common metadata; they represent a conceptual relationship between pieces of content. As a Site Manager, you can preset which Topics may be assigned to content.

Choosing a layout: Layouts are like templates for the design of a page. In most cases, you would need to have technical experience with code to change the way that content appears on a page and what content is allowed. With DKAN layouts you can choose from a set of layouts pre-made to beautifully combine different content in the same place without needing to touch any code.

Choose the layout for your Data Story and add data, media, text, etc. in the different panels. By default the most basic layout (Boxton) is selected, but choose the layout best fits the types of content you want to include for your Data Story.

Layouts are composed of different regions. Each rectangle and square shown in the different layouts is a region, and each region can contain one or more (or zero) pieces of content. Choosing the right layout is often a matter of trial and error depending on how the content is oriented and how you want it arranged. The regions in a layout are suited better for some content than others; as you add your content you can easily change the layout to meet your needs without losing any of the content.

5.2.7 Adding a Data Dashboard

DKAN Dashboards provide the ultimate flexibility in bringing content together. Layouts are like templates for the design of a page. In most cases, you would need to have technical experience with code to change the way that content appears on a page and what content is allowed. With DKAN layouts you can choose from a set of layouts pre-made to beautifully combine different content in the same place without needing to touch any code.

1. From the Admin Menu, hover over the **Add Content** menu link until a drop-down list appears.
2. From the list, select the **Data Dashboard** link.
3. Give the Dashboard a title that is short so that it's easy for site visitors to search and find.
4. Optionally, choose one or more Topics to associate with the Dashboard.
5. Give a brief summary of the dashboard in the description field explaining what kind of information it contains.
6. Choose a layout that best fits the expected arrangement of the content. Content will automatically be resized to fit the dimensions of the layout. Once a Dashboard is added, the layout may be changed at any time without losing its contents.

7. Click the **Save** button at the bottom of the page to add the Data Dashboard.

Once the Dashboard itself is added, content is added to the layout of the Dashboard in panes. Add visualizations, media, text, etc. to the Dashboard.

Example Data Dashboards can be found on the Dashboards page of demo.getdkan.com.

5.2.8 Layouts for Dashboards and Data Stories

Layouts are composed of different regions. Each rectangle and square shown in the different layouts is a region, and each region can contain one or more (or zero) pieces of content. Choosing the right layout is often a matter of trial and error depending on how the content is oriented and how you want it arranged. The regions in a layout are suited better for some content than others; as you add your content you can easily change the layout to meet your needs without losing any of the content.

Using the In-place Editor: Once you've selected the layout and save, you can begin adding content to the regions in the layout using the In-place Editor. The In-place Editor is a drag-and-drop tool that lets you visually place content within your selected layout and see a real-time preview of what it will look like once saved.

- **Add (+) button:** The button to add content is represented on the In-place Editor by a + icon. Click on the + button to add a new piece of content to the region. You can add as many pieces of content to a region as you want. The content will fit to the region of the layout regardless of how many pieces of content are added.
- **Style button:** The button to add styling to a region is represented by the paintbrush icon in the top-right corner of the region. Use this button to change the style of the region as a whole. That might affect the appearance (like adding rounded corners to the region) or the user experience (like making a region and its content collapsed or exposed).
- **Edit button:** You might think the Edit button is how you edit the content contained on your Dashboard. This button actually lets you edit the administrative details of the Dashboard. That includes information like the Title of the Dashboard, assigned Topics, authoring information, published status, etc.

Customize display: Site Managers can change the layout even after adding content to your Dashboard or reset if you want to remove all content. You can also use the content menu to see another view of the content on your Dashboard. This is useful for rearranging content after changing layouts or shifting several pieces of content on a Dashboard. Click on the content link to open another set of options.

- **Title type.** The Title type refers to how the title is set. Leave the selection at Manually set for your Dashboard to keep the original title. You won't change the title of your Dashboard here; this title is added and changed in the Edit menu with other administrative information.
- **Substitutions:** You won't need to manage Substitutions, so you can leave this option hidden.

Gear button: On the Customize display screen, you can use the gear icon on the region sections to add and manage content for the whole region as well as change the appearance settings. You can also edit each piece of content within a region using the individual gear icons in the content boxes.

5.2.9 Adding a Page

One of the most basic content types on DKAN is a Page. Though the content type is straightforward it has implications for the structure, appearance, and experience of your DKAN site.

Key Information when adding a Page:

Choosing a layout: Layouts are like templates for the design of a page. In most cases, you would need to have technical experience with code to change the way that content appears on a page and what content is allowed.

With layouts you can choose from a set of layouts pre-made to beautifully combine different content in the same place without needing to touch any code. Choose the layout for your Page and add data, media, text, etc. in the different panels.

By default the most basic layout (Boxton) is selected, but choose the layout best fits the types of content you want to include for your Page. Keep in mind, you can change your layout anytime.

Creating a menu link: The most important piece of creating a page is adding the navigation for it. In order for site visitors to find your page and benefit from its content, add a menu link and decide the parent menu item. For high-priority content, like a Contact page, put the link on the main menu bar. Otherwise, decide which parent page the new page belongs to.

Special note: We recommend that you do not add menu links to the Datasets, Groups, Stories, Dashboards, or Topics pages.

5.2.10 Adding a Group

Groups are both a way to collect common Datasets and enable an additional workflow on DKAN. On the outward-facing side, site visitors are able to browse and search Datasets specifically published by a Group, which is the common publisher of a number of Datasets.

Behind the scenes, Groups add an additional set of roles and permissions that ensure quality and security when publishing your data. Group roles and permissions ensure that Content Creators can add new data but only to their assigned Group. This is especially important for large sites that may have several working groups publishing data to the site. Read more about Group roles and permissions.

When first adding a new Group, the form has only a few fields. This is the basic information about the Group itself that should tell site visitors what to expect from the Datasets in the Group.

Key information when adding a Group:

- **Title:** Name your Group to reflect the agency or whoever the common data publisher is for the datasets that will belong to the Group.
- **Image:** The image here acts like the logo for your Group. It appears on the overview Groups page as well as the individual page of the Group itself. It's best to choose a square image to fit the dimensions of the thumbnail. Whether you choose an image, a logo, or an icon you can use any image that meets the size and file type requirements. As a Site Manager, you may want to add generic icons to the Groups you add if a current logo is unavailable.
- **Body text:** This text is the full description for your Group similar to an About page. The description includes details about the agency, its goals, and information about the data it publishes. While you want to include all the relevant information of the Group, the best descriptions are 1-2 paragraphs long and include a link to the agency's main web page for more details.
- **Summary text:** You can use the Summary to create unique text for your Group. This text appears as a snippet under the Group image on the Group overview page. If left blank the first portion of the body text will be used (about 100 words). Including a summary can be useful in adding more key search terms or using a different tone to intrigue site visitors to learn more.

Adding Datasets to a Group

Once you've added a new Group, you can assign Datasets (and their Resources) to that Group. Adding a Dataset to a Group is part of the content creation process when adding a new Dataset. The final step in creating any piece of content is using the submenu at the bottom of the form to add the final administrative data to the content. In the case of Datasets that includes adding Datasets to Groups.

When adding a Dataset to a Group, users can add a Dataset to as many Groups as there are on the site. Your groups are Groups that the user authoring the content belongs to, and Other groups are all the Groups of which a user is not a member. **All users must belong to at least one Group to have the Groups menu item available to them.**

When a Dataset is added to a Group, it will be included on the Group's home page and may be edited by the Administrator members of a Group. As a best practice, **users should only add Datasets to Groups that they are a member.** Certain users won't be able to access their own content if they assign it to a Group that they do not belong to.

Adding members to a Group

Groups have members, who must be first approved, and members have different roles in the Group. A user's membership status affects how they can interact with the Group. As a Site Manager, you can add members to a Group and give members different roles.

5.2.11 I added my content, where did it go?

You added new content, filled out the fields, included all the details, and then hit the Save button. Now what?

Regardless of the type, once you click on the Save button you'll next see a preview of how your content looks. Keep in mind that once content is saved (and if it has a published status) it is live on your DKAN site. That means the content is visible to the public. Most users can only save their content and have it directly published. Only Site Managers can add content in an unpublished state. The Preview screen shows you how the content will look to site visitors, so that you can make any final quick edits before moving on.

In the image below, you can see that the content is on the View screen and the content has just been created. This is how the Data Story will appear to a general site visitor (without the ability to edit, of course). At this point, you can get a sense of the appearance and use the In-place Editor to make any final changes.

Manage existing content: Once content is saved it is published and can be managed as existing content.

5.3 Admin Menu

The Admin Menu on your site is the main tool for navigating data and content management. Get familiar with the Admin Menu as a first step to mastering your DKAN site.

When you're logged onto the site, you'll see a black navigation bar at the top of the page. This is the Admin Menu. As a Site Manager, this menu looks different for you than other roles on the site. Editors and Content Creators have fewer options because they have fewer permissions on the site.

Use the Admin Menu as your anchor on your DKAN site. Everything you can do on the site is accessed through the Admin Menu.

5.3.1 Admin Menu Items

5.3.2 Add Content

Add Content is shortcuts menu item for creating new content. Click on Add Content to go to a page of available content types to create or simply choose from the drop-down menu. As a Site Manager, you have permissions to create all the content types possible.

5.3.3 Content

Click on the **Content** menu item to access all the content that exists on the site. As the Site Manager, you have access to create all the content types possible as well as edit, unpublish, and delete all existing content regardless of who the author is. You can create new content from this page as well as manage all the existing content and files on your DKAN site from here. Files include things like images, videos, font files for icons, other graphics, etc.

In smaller organizations, Site Managers may both be writing and editing their own content to then directly publish the content to the live site. Larger organizations may have people in other roles like Editors and Content Creators to help with handling a large mass of content on the site. Depending on the scale of your organization and volume of content, you may spend more or less time directly handling content.

In any scenario, as a Site Manager you can use this page to look at the content on the site to see who created a particular piece of content, when it was last updated, its status (published or unpublished) and take action on existing content. You can handle individual pieces of content, but you can also perform "bulk actions". This is a particularly useful function especially when dealing with a high volume of content. From this page you can take the same action on several different pieces of content by simply checking the boxes on the left, selecting an update option and clicking the Update button.

In the example below, the Site Manager is selecting a few pieces of content to perform a single action on. The Site Manager is able to unpublish 3 pieces of content at the same time by using bulk actions. While the example is simple, this function becomes helpful on site that contain thousands of pieces of content.

5.3.4 Visualizations

As a Site Manager, you have access to create visualizations on DKAN. This type of content is unique to Site Managers and Editors, and as a Site Manager you have access to manage all content regardless of the author. Typically, you'll use DKAN Charts.

Charts are a powerful tool for taking data and making it meaningful to the average site visitor who may have little to no experience with data and analysis. Click on the **Visualizations** menu item to access all the existing visualization content in your DKAN site, make edits, delete or create new visual content.

5.3.5 People

From the **People** menu item, you can create new users, manage existing accounts, and get a comprehensive view of all the users on the site.

The People main page gives you all the information about the users on your DKAN site including the username and role as well as the length of the account and when the user last accessed their account. This high-level view is especially helpful for managing a large number of accounts and performing "bulk actions".

Sort the order of info columns by clicking the title link. In other words you can sort A-Z by clicking the Username title link You can use filters and combinations of filters to search for users with specific roles, permissions, and status.

Make bulk actions to users by selecting the checkboxes on the the far left of the page, choosing an action from the Update options menu and clicking the **Update** button.

5.3.6 Site Configuration

These options let you manage general settings of certain features on your DKAN site like how user accounts are set up, search options, which fonts are applied to the text on the site and more. These settings determine how lower-access users interact with the site and give you flexibility to change the default behavior.

Click the **Site Configuration** menu item to see all the configuration options.

5.3.7 DKAN

From the DKAN menu, Site Managers can access some of the more technical operations of DKAN. DKAN is the technical engine that powers DKAN, and the available options in this menu are customized to DKAN and open data publishing.

Use the **DKAN** menu item to add APIs, enable External Previews, access the Harvest dashboard, manage the Recline configuration, and more.

5.3.8 Recline configuration

DKAN Internal Previews provide site visitors a visual snapshot of the contents of a Resource. Previews are powered by a tool called Recline that works in the background.

In cases where the file is relatively small (under 3MB) the Previews tool, Recline, will display the file contents without issue. To preview contents of a file larger than 3MB there are two options: import the file into the DKANDatastore or adjust the Recline configuration.

Import file: As a best practice, we recommend importing CSV files into the Datastore whenever possible. In the case of Internal Previews, if the file is imported to the Datastore there are no size limits on what a site visitor can preview.

Adjust Recline Configuration: For files that cannot be imported to the Datastore, the entire file is downloaded to be previewed. File size limits maintain a positive user experience by preventing errors or loading errors, however they can also keep a site visitor from seeing the contents of some file.

Site Managers can adjust the size limitations to be higher or lower with Recline Configuration:

1. From the Admin Menu, mouse over the DKAN menu item.
2. Select the Recline Configuration menu item.
3. On the Recline Configuration page, enter file size limits using standard conventions (MB, GB, etc.)

5.3.9 Caches

From the Admin Menu you can access caches to flush directly from the drop-down menu items.

Fig. 5.3: A sample listing of Charts from the DKAN Demo site.

More actions (home icon): On the Admin Menu, you can click on the Home icon any time to return to the Home page of your DKAN site. Additionally, if you hover over the home icon, you'll see two options.

Flush all caches: Flush all caches is a drop-down menu item that allows you to delete stored information on DKAN. Caches are helpful in storing information on the site that was recently used or likely to be used again in the near future. DKAN has a number of caches that are specialized to capture certain information in different places on the site.

- While caches are useful for keeping information easily accessible, they can significantly slow down computer speed as the caches accumulate more information. By flushing a cache, you delete the stored information and increase computer speed. Click on the main menu item to simply flush all the caches possible, or hover over the arrow to see all the individual options.

Note: By clicking any of the options to flush caches, you will not be taken to a landing page as with other menu items. Clicking on these menu items directly performs the task, and you'll get a confirmation message like in the image below.

*Drupal.org issue queues:** For Drupal-savvy users, the home icon also has Drupal.org menu items. Clicking directly on Drupal.org will take you the main Drupal website that contains extensive documentation on Drupal features, modules, functions and more. - You can also hover over the menu item to see a list of queues. These queues are a place to report and see already-reported issues that you may encounter on your DKAN site. Some of the queues are specific to DKAN and others are for general Drupal features. These queues are a good resource for troubleshooting issues and reporting problems so that they can be fixed if you're familiar with Drupal.

5.4 Appearance

You can make your DKAN site look and feel completely aligned with your organization to give site visitors the best possible experience while going through the site. Add your logo, change page layouts, customize the Home page, or pick out the right fonts to match your brand.

5.4.1 Colorizer

Colorizer is what powers the color scheme on a DKAN site. Color schemes are a large part of a consistent aesthetic, which is especially important when working with many collaborators.

The default color scheme gives your site a consistent look and feel and meets standard accessibility requirements. You can also customize the color scheme to align with your agency's image using Colorizer.

With Colorizer, you can choose a different preset color scheme or create your own custom color scheme for full customization.

Colorizer can be accessed by navigating to the Site Configuration drop-down menu and selecting "Colorizer."

Using Colorizer

Global settings: The global settings are technical details pulled from the CSS template that powers the Colorizer tool. The global settings remain the same, unless you have your own CSS template.

Color scheme settings: The color scheme settings customize a color scheme that will be consistent throughout your site.

Color scheme: Click on the Color scheme drop-down menu to choose from a list of preset themes. Click on the custom theme to choose your own.

Custom color scheme: Create your custom color scheme in one of the following ways:

- **Using the color picker:** With the custom color scheme selected, use the color picker to choose the primary color for your color scheme. The secondary colors will automatically update to complement the primary color selected.
- **Manually:** Hex numbers can be changed individually by manually entering a new hex number. When changed manually, secondary colors do not automatically update.

Save as default: Once the color scheme is selected, whether custom or preset, save the changes by clicking the Save as default button at the bottom of the page.

Clear Colorizer Cache: With custom themes, you can reset to the default theme by clicking the Clear Colorizer Cache at the bottom of the page. Clear the cache to start the customization process over or to simply use the default theme.

5.4.2 Theme Settings

The theme settings are options that personalize your website. Page elements like logo, site name, the hero section, etc. can all be customized to reflect your agency's brand.

Using theme settings:

List: Click the List button to see which site themes are enabled and disabled. From here, enable site themes and choose the default theme. Site themes affect how page elements are displayed on a site. By default, DKAN uses the NuBoot Radix theme. Use the List page to choose your theme.

Global settings: These options control the default display settings for your entire site, across all themes. Unless they have been overridden by a specific theme, these settings will be used.

- **Toggle display:** Turn the display of page elements on and off with the selected checkboxes. With these options, Site Managers can decide to show or not show the logo, site name, option to login, etc. Pay attention to elements like menus, since these affect how site visitors can navigate the site.
- **Logo image and Shortcut icon settings:** Uncheck the default logo box to add a custom logo. Small, but high-resolution photos work best for these elements. Using a custom logo and shortcut icon are additional ways to personalize your DKAN site.

NuBoot Radix: These options control the display settings for the NuBoot Radix theme. When your site is displayed using this theme, these settings will be used. Because NuBoot is the default theme, use these settings to make changes.

- **Footer text:** The footer appears on the bottom of every page on the site. In the Copyright section, edit the footer text for additional branding.
- **Hero Unit:** The front page display includes a large banner at the top of the page, called the Hero Unit. This image is the first contact a site visitor makes with your site, so it can make a strong visual impression. Add a high-resolution image or choose a solid color. In the example below, the solid blue color is the Hero Unit.

5.4.3 Featured Groups sort order

Featured Groups is a type of pane that highlights a set number of Groups on a page. If a DKAN site has a Featured Groups pane, Site Managers can decide which Groups are highlighted by changing the sort order of the Groups list.

1. Choose the order of how Groups appear in a Featured Groups pane with the sort order:
2. From the Admin Menu hover the mouse over the **DKAN** menu item.
3. From the drop-down menu, select **Featured Groups Sort Order**.
4. View the list of Groups; all Groups are listed.
5. Click and drag the cross icon to change the order of the Groups.

The pane reflects the order on the Featured Groups Sort Order page. To remove a Group from this display, move it further below on the list. Depending on the site, the Featured Groups pane could have as few as 1 Group or as many as 12. The pane displays the Groups at the top of the list for as many Groups are included in the pane.

5.4.4 Changing front page content

The landing page of any website is critical. It's the first impression you give a site visitor, and it may be the deciding factor if that site visitor continues to explore the content on your site or if they move on to the next thing. The content on your front page should be eye-catching to get site visitors interested in exploring more. As a Site Manager, you can customize the content of the front page to be engaging and timely.

Customize branding elements

The logo, shortcut icon, and hero unit on a DKAN website can be customized to fit a specific look and feel on an ongoing basis as designs and logos change over time.

Change the logo

1. From the Admin Menu, hover over the **Site Configuration** menu link to display the drop-down menu.
2. From the drop-down menu, select **Theme Settings.**
3. On the Theme Settings page, scroll down the page to the section labeled Logo image settings.
4. Uncheck the box labeled, Use the default logo.
5. Paste the URL to a remotely hosted image or upload an image.
6. Click the Save configuration button at the bottom of the page to finalize the changes.

For the logo, use a high resolution image sized with small to medium dimensions.

Change the shortcut icon

1. From the Admin Menu, hover over the **Site Configuration** menu link to display the drop-down menu.
2. From the drop-down menu, select **Theme Settings.**

3. On the Theme Settings page, scroll down the page to the section labeled Shortcut icon settings.
4. Uncheck the box labeled, Use the default shortcut icon.
5. Paste the URL to a remotely hosted image or upload an image.
6. Click the Save configuration button at the bottom of the page to finalize the changes.

For the favicon, use a high resolution image sized with small dimensions.

Change the hero unit

The hero unit on a website is the large, banner-like image or color on the front page. This section of the website is the backdrop for the welcome to new and returning site visitors. As the first impression, the hero unit can play an important role in retaining site visitors.

1. From the Admin Menu, hover over the **Site Configuration** menu link to display the drop-down menu.
2. From the drop-down menu, select **Theme Settings.**
3. On the Theme Settings page, click the tab labeled NuBoot Radix for specific theme settings.
4. Scroll down the page to the section labeled, Hero Unit.
5. For a hero unit using an image, upload the file by clicking the Choose file button (max size is 2MB).
6. For a hero unit using a solid color (recommended), enter the hex value of the color. Hex values use a hashtag with numbers and letter format such as #000000 or #4c9d9b

5.4.5 Using the In-Place Editor

The In-place Editor is a tool on DKAN designed to make it easy to add, change, and move around content elements directly on the page. You can use the In-place Editor to add new content, drag-and-drop content that already exists, and see changes on the page in real-time.

From the front page, click the Customize page button at the bottom of the page to open the In-Place Editor and access the existing content. You may want to leave some elements, like the search bar, or you may want to start completely new.

Using the In-place Editor gives you flexibility in the way content is displayed and curated. Add simple content types like images, videos, links, and text or you can add more dynamic content like visualizations and slideshows.

Because this feature shows changes in real-time, it's well-suited for testing how content will look before clicking the Save button. Play with the placement of content on the page, the styling of the region, and keep tweaking until the page looks right.

Add new content (+):

1. On the bottom of the Page, click the **Customize this page** button.
2. Choose the region to add new content to.
3. The add content button is represented on the In-place Editor by a + icon. Click on the + button to add a new piece of content to the region.

4. Choose what element to add to the page (exe visualization, text box, image, video, etc.)
5. Fill in the details of the form, specific to the type of content.
6. Click the **Finish** button.
7. Preview the changes on the Page. Make edits or rearrange with the In-place Editor.
8. Click the **Save** button at the bottom of the page to finalize the new content.

Which content, where?: Choosing the right content type will depend on the information and the overall arrangement of the different pieces of content. Visual content, like slideshows (a collection of rotating images), videos, maps, and visualizations grab attention and so they do well near the top of the page. Other content, like text, files, and tables add density to a page so less is more. They're best used for quick bites of information and as complements to visual content.

In the example below, the Site Manager is using the In-place Editor to move content from region to region and delete irrelevant content. the placement of the content is previewed in real-time so you can see how the page will look once the changes are saved and published to the public site.

Edit existing content:

1. Click the **Customize this page** button at the bottom of the page with the content you want to edit.
2. On the pane of the content, click the the gear icon. Each pane can be edited individually by clicking the gear icon for the specific pane with the content to be edited.
3. A menu will appear with the administrative details included when the content was first created.
4. Use the Content settings section to make changes.
5. Click the **Finish** button at the bottom of the screen to save the changes.
6. Review changes on the Page and continue to make changes as needed with the In-place Editor.
7. Click the **Save** button at the bottom of the page to finalize the new content.

Remove content: To remove the entire pane and all its contents click the trash can icon to remove the pane in one click. Once you choose to remove a pane it cannot be undone, so be careful when removing content.

Style a whole region or a single pane: Styling gives you even more control over the experience a site visitor has going through the site. You can style an entire region to affect all the panes within the region to have the same styling.

Alternatively, you can choose a style for single panes within a region. The style button is represented by the paintbrush icon in the top-right corner of the region or pane.

Use this button to change the style of the region as a whole or an individual pane. Styling may affect the appearance (like adding rounded corners to the region) or the user experience (like making a region and its content collapsed or exposed by default).

In the image below, none of the panes have any styling applied.

In the image below each pane has a style different from the others in the same region (from left to right): rounded corners, collapsible (not by default), no style, collapsible (by default).

In the example below the Site Manager is changing the style for the entire region on a page. The Site Manager chose rounded corners for the style. This style groups together a set of information to help guide the site visitor and create

a clear differentiation from the other sets of information. When adding styling, you want to think about how it can improve the experience a site visitor has on your site rather than choosing a style based solely on the aesthetic.

5.4.6 Choose fonts

As a Site Manager, you can control what fonts are used for the text across your DKAN site. By setting fonts, the text is formatted automatically rather than leaving it to individual users to decide.

This maintains a consistent style throughout your site which is important to the site visitor's ease in navigating content. User experience can't be emphasized enough. It can make the difference of a person going exploring through your content or leaving after clicking on just a couple pages.

While DKAN defaults to a certain font for text styles, the Site Manager has control over fonts to best match the overall look and feel of the organization. Mix and match which fonts are used for specific text areas. Choose one font for text in description and summary boxes, choose another for all headers or a specific size of header or add custom font specifications.

Enabling fonts

Before making font selections, you first have to enable fonts.

1. On the Admin Menu, find the **Site Configuration** menu link.
2. From the drop-down, select the **Fonts** menu item to access fonts and set defaults.
3. Displayed is the list of fonts that have been enabled as well as browse fonts to either enable or disable. By default, no fonts are enabled; texts appears in the default font set by the DKAN theme.
4. Click on the **Browse all fonts** link.
5. Select a number of fonts to enable and click on the Enable link for each font selected. Once enabled the font can be used in default settings.
6. Click the Enabled fonts link to view the fonts just enabled.

Enabled fonts are the options that may be used for default font settings. Browse from a list of over 3000 font options provided by Google fonts.

The tiles have the name of the font as well as a preview, and tiles in green mean that the font is currently enabled. Switch to a list view of fonts you've already enabled (if applicable) by selecting the **Enabled fonts** tab.

Once fonts are enabled, they will appear on the fonts main page and you can set defaults with these fonts. Quickly and easily disable fonts from either the **Enabled fonts** menu or while looking for new fonts in the **Browse all fonts** menu. Click the **Disable** link associated with the font.

Selecting default fonts

Once your fonts are enabled, Site Managers can make selections for how text appears on the site by default.

From the **Enabled fonts** menu, make selections based on the font or based on the CSS selector. CSS selector is a technical term that refers to headers, body text, and any other selectors.

Find the font you want to apply and use the drop-down menu in the CSS Selector section to choose the right selector. This is good for quick changes to your font selections. Click the **Save applied CSS selectors** to finalize your choices.

Alternatively, you can apply fonts to CSS selectors by clicking the By CSS Selector tab.

Here, you can make several changes to the different selectors and apply font changes. This is a good option when you have many changes to make to your font selections. You can also add fonts to the Fallbacks column for your primary font selections.

By adding a font in this field under the Fallbacks menu section next to the primary selection the fallback font will appear if the primary font is unable to display.

You can select multiple fallback fonts by using commas; if none are selected then the default DKAN font will display.

Click the **Save applied fonts** button at the bottom of the page to save your selections.

5.5 Data and Content

On DKAN the data you publish for site visitors is handled as content. Treating data like content makes it easier for general site visitors to consume information and accessible for people.

Equally important is making your open data accessible for computers (ie machine-usable). Technologists leverage this access by developing applications to interact with your open data programmatically. This opens an entire realm of possibilities.

DKAN is provides tools for generating content for the technical and non-technical users of your open data site, along with tools for Site Managers to monitor and manage that content. The following sections explain all of those capabilities.

5.5.1 DKAN Content Types

While DKAN handles data as content along with many other types, not all content is the same. You can think of content in three different ways on DKAN: data, narrative, and curation content. Narrative and curation content support the data content published on your DKAN site. They help highlight the investment and value of open data that might not otherwise be apparent.

Data content types

Data content is specifically for publishing and managing open data. That is, the actual files in which your open data is contained. These include Resources and Datasets.

Resources

Resources are the most basic content type on DKAN. They represent the actual file that can be viewed and downloaded by site visitors. DKAN supports a large range of file formats including csv, html, xls, json, xlsx, doc, docx, rdf, txt, jpg, png, gif, tiff, pdf, odf, ods, odt, tsv, geojson and xml. Resources may be uploaded alone, but in order to include metadata you must add the Resource to a Dataset.

Add a Resource when you're trying to:

- Upload files of open data to your DKAN site.
- Link to data from an online, external source.

- Publish many different formats of the same data.

Datasets

Datasets are often described as "containers" because they group related pieces of data (Resources). Grouping Resources together in Datasets gives the data a common summary description, licensing information, and a unique URL to easily share the dataset directly. The image above shows the Resources, which are the actual files that can be previewed and downloaded by site visitors.

In the image below everything surrounding the Resources is the "container", or Dataset. It includes a name for the Dataset, a description of the collection of Resources, tags, license information, the data author, and other metadata about the Resources. Resources may be uploaded alone, but metadata is associated with the dataset and cannot be individually attributed to a Resource.

Add a Dataset when you're trying to:

- Collect Resources with common metadata (such as author, open data license information, tags, etc.)
- Create a unique URL to link site visitors directly to a Dataset.

Narrative content types

Narrative content is designed to tell the story and reveal insights of data. Narrative content allows you to add context and bring the personal elements of data out to the front. These include Data Stories and Visualizations.

Data Stories

Data Stories show the people in data. It's not always obvious how rows and columns can make a difference in the world of citizens. Data Stories help show the impact data can make in our everyday lives by adding context and synthesis through a narrative form. This content type is easy to use and similar to writing a blog post. Use Data Stories to combine narrative, media, and data for a compelling way to connect to data.

Add a Data Story when you're trying to:

- Tell the story within the data on your DKAN site.
- Show the impact of data on citizens' daily lives.
- Demonstrate the value and return on investment of open data.

Visualizations

Visualizations on DKAN create a powerful access point for users. Visual representations of data are simpler to comprehend, regardless of technical expertise, and make large quantities of data consumable rather than overwhelming. DKAN Charts (pictured below) can also be interactive, letting site visitors toggle which data points are shown in the Chart. Use Visualizations to make your data more than just available.

Add a Visualization when you're trying to:

- Make large datasets understandable instead of overwhelming.

- Make your data available and accessible to site visitors regardless of technical expertise.
- Use data-driven decision making for internal use.

Curation content types

Curation content types bring both data and narrative content types into conversation to create context and bring out meaning. These include: Pages, Dashboards, and Groups.

Pages

One of the most basic content types on DKAN is a Page. Though the content type is straightforward it has implications for the structure, appearance, and experience of your DKAN site. You can think about Pages as the anchors of your site.

With a Page, not only do you add content but you create additional space on your DKAN site. With other pieces of content, like Dashboards and Data Stories, the content is collected onto a single page. In contrast, Pages typically won't change themselves, though the content on them could change at any frequency.

Add a Page when you're trying to:

- Add a new space on your DKAN site for key information (like an About page).
- Add a space that itself won't change, though the contents within it might (like a Contact page).

Data Dashboards

Dashboards provide the ultimate flexibility for you to give site visitors the best experience possible while also showcasing the data and content on your DKAN site. Using this curation content type, you can mix and match all kinds of content without ever touching code.

Mix videos, images slideshows, DKAN Visualizations, text, tables, and maps to most effectively deliver your content. With more than 20 responsive layouts to choose from and our easy to use drag and drop interface, any user can create compelling data-powered content within minutes.

Add a Dashboard when you're trying to:

- Combine many different types of content in one place.
- Put data and narrative content in conversation to better connect open data to citizens.
- Use flexible layouts for easy content curation.

Groups

Groups are both a way to collect common Datasets and enable an additional workflow on DKAN. On the outward-facing side, site visitors are able to browse Datasets published by a specific Group, which is the common publisher of a number of Datasets.

Behind the scenes, Groups add an additional set of roles and permissions that ensure quality and security when publishing data. Group roles and permissions ensure that Content Creators can add new data but only to their assigned

Fig. 5.4: Example of a list of resources on a dataset page.

Group. This is especially important for large sites that may have several working groups publishing data to the site. Read more about *Group roles and permissions* .

Add a Group when you're trying to:

- Collect and categorize Datasets by a common publisher.
- Need a workflow for Content Creators to publish data to your DKAN site within a specific Group.

5.5.2 Managing Existing Content

In smaller organizations, Site Managers may both be writing and editing their own content to then directly publish the content to the live site. Larger organizations may have people in other roles like Editors and Content Creators to help with handling a large mass of content on the site. Depending on the scale of your organization and volume of content, you may spend more or less time directly handling content. Learn more about user roles in the *People* section of this guide.

Where do I edit existing content?

Site Managers can edit any content on the site, either from the main Content page or directly on the piece of content itself.

Edit from the Content menu

The main Content page is the most comprehensive place to find and access content to edit an individual piece. Though the menu is comprehensive, editing the details of a piece of content can only happen one at a time. From the Admin Menu, click on the **Content** link to access all the content that exists on the site.

As the Site Manager, you have access to create all content types possible as well as edit, unpublish, and delete all existing content regardless of who the author is. You can create new content from this page as well as manage all the existing content and files on your DKAN site from here. Files include things like images, videos, font files for icons, other graphics, etc.

In any scenario, as a Site Manager you can use this page to look at the content on the site to see who created a particular piece of content, when it was last updated, its status (published or unpublished) and take action on existing content.

Which content, when?

Oversight over the details of content typically falls to the Editor role, but depending on the size of your team the Site Manager may play multiple roles and handle more content review and revision. Ultimately the quality of the content on your DKAN site falls to you as the Site Manager.

When any user creates a new piece of content it defaults to a published state. That means the content automatically appears on the live site, visible to the general public. If you need a publishing review process on DKAN, you can enable DKAN Workflow. This module provides a system for reviewing content before it is published.

Fig. 5.5: Example of a dataset page.

Fig. 5.6: Example of the Stories page.

Fig. 5.7: Example of the pie chart visualizations added to a Data Story page.

Though you might not edit every piece of content, you'll generally review all the content that is visible to the public (in a published state) and make edits whenever necessary. Only Site Managers have the ability to edit or delete any piece of content regardless of the author, so use your best judgment when reviewing and editing content. If you're ever unsure, you can unpublish the content and go back to the author before making changes or removing the content.

Filtering content

Your DKAN site may have hundreds, or even thousands, of pieces of content from Resources to Data Stories; and as a Site Manager you'll have access to every piece of content. You can sift through the content and use filters on the Content main page to find content effectively. Filters help narrow your search by a set of criteria.

Title Type in text to narrow results to content where the title contains the text entered here.

Type Select a content type from the drop-down menu to limit the results to content that matches the type selected.

Published Limit results to published or unpublished content.

Author Type in a username to limit the results to content submitted by the user entered here.

Date Use the date filters to limit the results to content matching the dates entered.

Bulk Operations

To perform a task on multiple pieces of content at one time use the Operations selector. This is good for publishing/unpublishing/deleting several items at once, or assigning content to a different author.

1. Navigate to the Content page *admin/content*
2. (Optional) Filter the content to bring the content you want to work with to the top of the list.
3. Check the box next to each item you want to run the operation on.
4. Select an operation from the **Operations** dropdown list.
5. Click the **Execute** button.

Edit from the page

As you navigate through your DKAN site, you have more options available to you as a Site Manager than other users. You'll see these options on every page including an option to edit while on the page of a piece of content. You can edit directly from the page by clicking the **Edit** button. The options for editing content are the same as when adding new content, and the form appears the same.

In general, it's okay to make small changes to content while it's published. Small changes include adding Tags and Topics, adding or removing the content from a Group, spelling and grammar corrections, styling, and other minor updates.

For major changes to any piece of content, it's best to first unpublish the content and continue to make changes behind the scenes. You can change the status of the content by unchecking the **Published** box in the administrative submenu

Fig. 5.8: Different layouts that can be used to build Dashboards.

Fig. 5.9: Example of a Groups list.

at the bottom of the form. Major changes include changing the title, changing the layout, adding new information that needs review, adding or changing images, and any other changes that aren't ready to be published immediately.

Fig. 5.10: In this example, a Site Manager visits a Dataset page and decides to edit the content. The Site Manager clicks on the **Edit** button to open the form and make changes to the Dataset.

View changes

When editing content, a user is making changes to something that is published and visible to the general public. Before saving the changes, you can see exactly what changes have been made in a summary form to check that all the changes are correct. Use the View changes button at the bottom of the page of any piece of content to get the breakdown.

The changes are organized into two columns and sectioned off further by the different parts of the content for a side-by-side comparison of what has changed in the piece of content. On the right side, there is the column with the original version (called Original). On the left side, new additions are added to the Changes column.

Original The column titled Original shows what information was in the original version before any changes were made. It contains information that has been removed or, alternatively, what remains when new information is added. The column is further organized into the different fields that make up the content form. These fields include the body text, Topics, Tags, metadata, Groups, etc. so you can see exactly where changes were made. When information is removed, the changes are highlighted in yellow and have a minus sign to the right-hand side. If information remains when new information is added, then the information appears in the Original column, but it's not highlighted and there is no minus sign. In that case, there should be new information in the Changes column.

Changes The column titled Changes shows what information will appear in the new version once changes are saved. It contains information that has been added or, alternatively, what remains if information is removed.

Fig. 5.11: In this example, a Topic has been removed and a new Topic added. In the Changes to Tags section, you can see that the "trees" Tag was removed. The Changes column shows the information that will be included in the new version (everything that wasn't removed).

Revisions

Revisions is a powerful capability, especially when working on content that undergoes several changes. Revisions help track and record changes to create a backup of a piece of content, but they're not automatically generated every time a change is made. Create a new revision any time that a significant change is made to create versions of a piece of content. This will create a safety net in case anything major needs to be reversed. It will also document institutional knowledge as there are personnel changes at your organization.

Keep in mind that once you delete a piece of content, you also delete its revision history. We suggest unpublishing content, rather than deleting it, so that it's not visible to the general public but still exists on your site behind the scenes.

Create a new revision

Once a piece of content has been published, you can go back and make changes to the content as needed. When you make a change, you have the option to create a new revision of the content. This creates a new version with the changes incorporated, but also keeps the old version. In general, you don't need to worry about creating a new revision

for minor changes unless the information being added is critical. For bigger changes, it's helpful to have a backup especially if you're editing content for someone else.

Click the **Edit** button to open the content form and go to the bottom of the page. In the Revision information tab, click the option labeled **Create new revision**. This is all you need to create a new version for the piece of content.

Notice there's another field below the new revision option labeled Revision log message. This is a space to explain the changes you're making and why. This is particularly useful for content that goes between multiple authors. Giving an explanation for changes can help clarify reasons that might not be obvious which helps reduces confusion and mistakes that can be easily avoided.

Fig. 5.12: In this example, the Site Manager is creating a new revision and includes an overview for what was changed in the Revision log message field. The Site Manager saves the changes, and then finds the new revision on the Revisions page.

Where to find revisions

You can access all the revisions of a piece of content by going directly to the page of the published content in View mode. Click the Revisions button to get to the Revisions page to see all the existing revisions and the revision options. The Revisions page gives an overview of all the existing revisions including when the revision was created, by which user, and the revision log message. Here, you can also compare revisions and go back to an old revision (reverting).

Revision options

From the Revisions page, there are additional options for what you can do with the different revisions for the piece of content. The most important options are Compare and Revert.

Compare Some content may have several revisions that are difficult to visually scan what the differences are. To get a clear sense of the distinction between two revisions, you can select revisions to compare on after the other. Click on the selection circle and then click the Compare button at the top of the columns. You'll see more details of each revision in comparison with one another.

Revert As changes are made, you may not create a new revision each time. Revisions typically reflect significant changes to your published content, so lots of small changes can be made in between the previous and current version. You can go back to an older version of a piece content by clicking the Revert link under the Operations column in the row of the revision. Before reverting, it's best to create a new revision of the content with the most recent changes. The most recent revision will still exist in the list of revisions, but you can revert to an older revision, which is what will appear on your DKAN site.

Managing content on DKAN

After you add a piece of content you may want to make changes or updates or you may want to replace a piece of content all together. As a Site Manager, you can make changes to any piece of content regardless of the author, type of content, or state of the content. Creating and editing content on DKAN are critical functions of management, and there are even more ways to manage your content.

Content visibility

In general, most content is published at the same time it's added. Once content is published it appears on your live DKAN site meaning that it can be searched and discovered by the general public. But if you want to work with content

without making it public or if you just want to keep a piece of content rather than delete it but don't want it visible on your DKAN site, you can change the visibility.

There are two states that content can be in that determines its visibility to the public, published and unpublished.

Published content Published content is live on your DKAN site and visible to the public. In some cases, you may want to access and change content that was not created recently. You can visit published content on your DKAN site and edit directly from the page for quick changes (spelling, typos, titles). For bigger changes, it's best to not edit live content. First unpublish the DKAN, make changes, and then publish again.

Unpublished content If content is unpublished that means that it doesn't appear on your live DKAN site, but it still exists behind the scenes. Users can leave content unpublished if they have significant changes or if they want to come back to a piece of content later to finish editing.

As a Site Manager, you can access all the existing content on your DKAN site through through the Content menu item on the Admin Menu bar. Once you unpublish a piece of content, that content is no longer visible to users with lower permissions (even if they are the author).

When content is added, by any user, the content defaults to a published state. Users with fewer permissions can edit the content or delete it, but they can't unpublish content or view unpublished content, even if they authored the content. That means that the content is left visible on your DKAN site until the author makes changes.

In some cases the changes may be minor revisions, but other scenarios could require heavy editing before the content is ready to be made public. Some content is simply time-sensitive; the information is authored ahead of time but shouldn't be made public until a certain date. Publishing states let you manage the visibility of your content to give Site Managers greater control over the content that appears on your DKAN site.

Deleting content

In general, we don't recommend deleting content. On DKAN, once content is deleted there is no way to recover it. You want to be completely sure of your decision before making it permanent.

A better practice is to unpublish content, which keeps the content on the site but doesn't appear to the general public. If you decide that deleting a piece of content is the best action, you can delete a single piece of content from the Content main page using the delete link in the Operations column or directly from the page while in Edit mode using the Delete button at the bottom of the page. You can also delete multiple pieces of content using bulk actions.

5.5.3 Data Management

DKAN supports broader data management strategies with tools that simplify and streamline data migration, storage, and usability. The tools are designed to be straightforward so that you don't need to be a technical expert to use them. Migrate open data efficiently and effectively, store your data, and improve the overall usability of your data.

Harvest

As you build up your DKAN site with content and data, you can add datasets by uploading or linking to resources and APIs. You can also "harvest" datasets from other data portals that have a public data.json or an XML endpoint. All the datasets published in the source are added to your DKAN site as the Dataset content type.

Unlike linking to a file hosted on the web, harvested datasets are fully imported from an external source onto your DKAN site. That means that the datasets exist both on the external source and your site independently from one another with all the same information (including title, metadata, tags, etc.).

By importing datasets from external sources, you can provide more open data on your site without manually managing the dataset content. Site visitors will see that a dataset was harvested and from which portal, promoting visibility

across agencies and sectors. Harvest optimizes importing, publishing and updating imported datasets into a single streamlined process.

How Harvest Works

The source is defined (fetching) First, identify where the datasets should be imported from. Harvest is compatible with data.json and XML endpoints, and your source may be configured before import. By default, all the datasets and their metadata are imported from a source, but you can further configure exceptions to narrow what information is included (more on that in the Harvest Dashboard section). Site Managers define a source by creating a Harvest Source as a new piece of content.

Data is stored locally (caching) as a copy Once the source is identified, Harvest pulls datasets and stores them on the computer's local hard drive. This is called caching. Site Managers have two options for how cached data is handled. Datasets may be cached and migrated (step 3) automatically or they may be only cached to be reviewed and later migrated manually. You may use a different operation depending on the context and can choose on a case-by-case basis.

Harvest adds cached data to your site (migration and publishing) After the dataset information is cached onto the computer's local hard drive as a JSON file, Harvest reads the file and imports it to your DKAN site. Datasets can be migrated and published to your DKAN site automatically as part of the caching operation. Alternatively, datasets can be only migrated (without caching).

Harvest is used to check for changes to migrated datasets Once the dataset information is imported to your DKAN site, its contents exist independently from the original source. That means changes made to a dataset on the original source won't appear on your DKAN site unless the harvested dataset is updated. It also means that if you make changes to the dataset on your DKAN site, the changes will be overwritten when you run a harvest operation to update the file contents and metadata.

With Harvest, you can make updates to your harvested datasets by repeating the process of fetching, caching and migrating. Harvest replaces the old information with the current datasets, updating the information to include any changes made to the original source. With defined sources, the process is a quick operation.

Though most of Harvest works in the background, Site Managers can use the Harvest Dashboard to manage Harvest operations.

Harvest Dashboard

Harvest Sources have special handling. The Harvest Dashboard displays all the Sources on a site and a comprehensive list of harvested datasets.

From here, Site Managers can view Harvest Sources, Source metadata such as date last updated and number of datasets in the Source, view harvested datasets individually, filter and search Harvest content, and perform bulk operations on Harvest content.

The Harvest Dashboard is also used to perform Harvest operations, edit and configure a Source, check the status of a Source, and manage the datasets in a Source.

Harvest operations

Site Managers use special operations on the Harvest Dashboard to manage the harvesting process for existing Sources. From the Sources tab, Sources may be cached, migrated, harvested (cached and migrated) or deleted.

Cache Source(s): This operation parses the Source endpoint (either data.json or XML) stores a subset of the Source data (reflecting parameters set by the Source configuration) on a computer. Caching data from the Source endpoint pulls the latest data, so the datasets on your DKAN site are current.

Migrate Source(s): Migrating a Source imports cached data from local computer storage and uploads files as content to your site. For existing Sources, the new data will replace what was previously published or create a new dataset if it wasn't previously published to your site.

Harvest Source(s): The harvest operation combines the cache and migrate operations for a single, streamlined process. This option automates some of the work that would otherwise be done manually, but it also removes the ability to review datasets before migrating to your site.

Delete Source(s): By deleting a Source, all the datasets associated with the defined endpoint are removed from your DKAN site. This is a permanent change.

Edit and configure a Source

The edit view of a Source opens the same options available when first adding a new Harvest Source.

Here, you can change the basic information about a Source, like the title and the URI. The basic information of a Source doesn't typically change once it's set.

In this same view, Site Managers configure harvests with filters, excludes, overrides, and defaults. With these options, Site Managers can customize what information is pulled from a Source and how metadata values are handled during the harvesting process.

Filters: Filters restrict which datasets imported by setting a pair of key values. For instance, if you are harvesting from a data.json endpoint and want to harvest only health-related datasets, you might add a filter with "keyword" in the first text box, and "health" in the second. With this configuration, only datasets that meet the stated criteria are imported.

Excludes: Excludes are the inverse of filters. Values in this field determine which datasets are left out of an import, while all other datasets are included. For example, if there is one publisher included in a Source whose datasets you donotwant to bring onto your site, you might add "publisher" in the first text box and "Office of Public Affairs" in the second.

Overrides: Values included in the Overrides field will replace metadata values from the Source as it's migrated in a harvest. For example, to change the name of the publisher, you might add "publisher" in the first text box to be replaced by the value in the second text box, like your own agency's name.

Defaults: In some cases, datasets from a Source may not have all metadata fields filled with a value. Use defaults to replace an empty field. For example, the first box might designate the License metadata value to be replaced if empty. The second box designates which value should replace it, like "Creative Commons".

Check the status of a Harvest Source

As Sources go through the harvesting process, Harvest captures the details and displays the results. After a Harvest Source is created and the datasets harvested are published to your DKAN site, the original source may change. Datasets may be added, removed, edited, and otherwise modified. These changes are reflected in a Harvest Source when a harvest operation is performed as part of the status of that Harvest Source.

There are two places to find specific details about a harvest operation on the Harvest Dashboard: the Events tab and the Errors tab.

Events: Each Harvest Source has an event log under the Events tab. When a Source is harvested, the process is recorded as an event. Sources are updated by running the harvest operation, so there may be several events recorded and detailed in this log. The event log is helpful for checking harvest events and getting the status breakdown on the most recent harvest, the number of new datasets created, datasets updated, datasets that failed to upload, datasets that have become "orphaned" on your site, and unchanged datasets.

Errors: Harvest Sources have an error log under the Errors tab to display the details of when a harvest encounters and error with the Source or a dataset in the Source. Error messages appear individually with the time and date it occurred as well as a message for the likely cause of an error. Details in the error log help identify the specifics of an error and find the best solution.

Manage Harvest Source datasets

Though harvested datasets appear alongside directly-published Datasets on your DKAN site, it's best practice for Site Managers to manage harvested datasets with the Harvest Dashboard. The Harvest Dashboard provides more specific information like when a dataset was updated from a harvest, its "orphan" status, and its Harvest Source.

Site Managers can either permanently delete or unpublish (recommended) harvested datasets.

Managing orphan datasets

After a Source is harvested, the datasets belonging to the source may change and may be deleted all together. When a dataset is deleted from the Source, but remains published to your DKAN site, the dataset is considered an orphan.

Because the Source no longer contains the dataset, it isn't updated as part of a harvest operation. But it isn't deleted from your DKAN site automatically. Site Managers must make a judgment call on whether to delete the dataset and stay aligned with the Harvest Source, to unpublish the dataset and hide from public view, or to keep the dataset as a stand-alone dataset that won't be updated through a harvest operation.

Visit the Adding Content section to learn how to add a Harvest Source.

Datastore

DKAN comes standard with a Datastore to house tabular data imported from your CSV files on DKAN. That is, the Datastore can support files with contents that appear as a table (rows and columns). You can think of the Datastore like a basic database. Files that are imported to the Datastore have the contents of the file copied into a table in the Datastore, and the Datastore as a whole is composed of all the tables copied from imported files on DKAN.The Datastore processes data, stores the contents of Resources (if CSV), and makes them ready to be queried.

As a Site Manager, you can manage the Datastore by adding and removing files from the Datastore. In most cases you want all CSV files included in the Datastore to support better data previewing, large files, and a more robust API.

Managing the Datastore

In broad strokes, managing the DKAN Datastore is deciding which Resources to include in the Datastore. There isn't any management further than that, and every user has the ability to import and remove Resources they've authored. As a Site Manager, you can import or remove any Resource regardless of the author. This allows you to manage what data is included in the Datastore API.

There may be some sensitive data that should not be included in the Datastore, but in general we recommend increasing your transparency and usability of your data by importing every Resource possible into your Datastore.

Importing and removing files

Uploading files to the Datastore has major benefits including enhancing the Datastore API and improved user experience of previewing data. The Datastore API makes the Resources more usable and accessible to technical users. Previews display resources as graphs, grids, or maps for geospatial data. In some cases files contain thousands (or

millions) of rows. For data on such high order, users can only properly preview the data if the Resource has been imported into the DKAN Datastore.

Read more on managing the datastore here

5.5.4 DKAN APIs

In the same way that data on a DKAN site is made accessible to general users through sensory devices like visualizations, dashboards, and data stories, DKAN APIs are essential to developers. DKAN comes with three API's, the Dataset API, the Datastore API, and the Dataset REST API. These can be queried to discover metadata associated with the catalog or a specific dataset. It can also be harvested by other portals by complying with the API standard data.json.

Datastore API

The DKAN Datastore houses any CSV files that have been uploaded and then imported into the Datastore. One reason for importing CSV files into the Datastore is to make the file part of the public Datastore API. Allowing data from uploaded files to be included in the Datastore API greatly increases ability to discover the information as well as the usability and accessibility of the data.

The Datastore API enables interactions with specific Resources in the DKAN Datastore down to specific rows. Without the Datastore API data could not be searched with such precision.

With the Datastore API, technical users can write programs to interact with the information held within the API. This means that the data in the DKAN Datastore can be accurately and efficiently queried (searched) and the data then used in other contexts and applications. The Datastore API is another way to show how open data can be used to provide a tangible return for citizens.

Fig. 5.13: Datastore API view on a Resource page.

Using the API

For any Resource imported to the Datastore, click Manage the Datastore. Then click the Data API button to get information about the Resource.

You won't perform API queries from here, but you can get linked to a sample query and find *documentation for more instruction* on how to use the Datastore API. The image below is the information returned on a sample query of the DKAN Datastore using the Datastore API.

Fig. 5.14: This query shows the results in a "raw" form. This is generally more difficult to read, but it is what appears with a standard query.

Fig. 5.15: This image shows a formatted view of another API query using a web extension to make the results easier to read.

Dataset API

The Dataset API works by combining a number of public, supported APIs. Each API is different in what level of detail it pulls out from the information available on a DKAN site. Together, the suite of APIs makes it possible to pull out information with varying specificity. When data is uploaded to DKAN as a Resource or Dataset, it is automatically included in the Dataset API. With the Dataset API, a public data.json file is also automatically published to follow Project Open Data standards.

As a Site Manager, you don't have to worry about managing the Dataset API because all the work happens in the background. At any point you can access the data.json file by simply typing /data.json in the URL after the homepage URL. You'll get a page with all the information from the Datasets and Resources on your DKAN site.

Fig. 5.16: This shows the results in a "raw" form. This is generally more difficult to read, but it is what appears with a standard query.

Fig. 5.17: This shows a formatted view using a web extension to make the results easier to read.

Dataset REST API

REST continues to be one of the most popular API styles in web development, and it's quickly becoming the standard for API design. In large part, that's because REST is designed to be light-weight, fast, portable, and simple to implement.

With REST APIs, external web applications and services can supplement platforms without significant investment or complexity in programmatic communication. The Dataset REST API leverages the opportunities of REST to open greater possibilities on DKAN.

With the Dataset REST API, you can take action on your DKAN site programmatically. For example, adding a Dataset through the API. For one or two Datasets it might not seem like a drastic difference; but imagine automating the action of adding dozens, or even hundreds, of Datasets. Or imagine you have an application built in-house that you want to integrate with DKAN. The Dataset REST API makes these actions (and many more) possible.

We have *technical documentation on the Dataset REST API* that provides more information on how to use it.

5.6 People

Data and content management can't be done alone. Site Managers handle all the users on the site and decide who has access to what. They can also create Groups for organizations, which have even more possibilities for roles and permissions. Build your team to build your site.

5.6.1 DKAN User Accounts

On DKAN, what you can do on the site depends on the permissions given to the role assigned to you. These roles can range from general site visitors to trusted Editors working behind the scenes. User roles and permissions maintain the security of your site, distribute workload without compromising quality, and lead to overall better content on your DKAN site.

Roles and Permissions Overview

There are 6 standard roles with set permissions. The following is a list of each role with a description of its purpose and a general description of what the role is able to do. Multiple roles can be assigned to a user, but generally they are in a hierarchy where any higher level role has equal and greater permissions of a lower level role.

Anonymous User

This is any site visitor accessing the site who is not logged in. Anyone who is not authenticated is an anonymous user. It is sometimes useful to log out of your account to view pages as an anonymous user will see them.

Permissions:

- View and search published content

Authenticated user

All users with login credentials have this role, but the additional functionality is limited. This role is used for a general site visitor who has created an account, but hasn't been given permission to add or edit content on the site.

Permissions:

- Create a profile

Content Creator

Content Creators are able to work with the actual content of your DKAN site. This role is good for a user to add their data to your DKAN site, but who doesn't need to access more sensitive functionality.

Permissions:

- Create most content (Datasets, Resources, Datastories, ...)
- Edit their own content (cannot edit content created by other users or view unpublished content)
- Add Resources to the Datastore

Editor

An Editor is responsible for managing content from a strategic perspective. This role is fit for a user who will create, edit, revise and delete content on a frequent basis, and should be given to a colleague with expertise on the subject matter at hand. This role is able to make changes to content and where it appears, but it doesn't go further into amidninistrative functions.

Permissions:

- Add, edit and delete most content types
- Cannot create, modify, or delete Groups
- Cannot modify the roles of other users

Site Manager

This role is the highest level possible for non-technical users. A Site Manager performs administrative functions, and is a role best suited for a supervisor, manager, or other trusted upper-level employee. The Site Manager is provided with a sweeping overview of the site as well as its content and users. However, they do not deal with the technical back-end.

Permissions:

- Create, edit, and delete all content types created by any user
- Create and manage groups
- Change menu structure
- Administer users
- Configure Harvests
- Modify DKAN specific settings

Administrator

Admins hold the highest level of all roles and permissions and have no restrictions. Administrators are able to modify settings of the underlying Drupal platform, and can modify most things of the site to meet user needs. This role is for a web professional with high technical competency and a good understanding of how Drupal works.

Permissions:

- Modify themes and layouts, and enable or disable modules.
- Modify Drupal settings

Account settings

Some settings in user management are automated to streamline the process of adding new accounts. From the Site Configuration menu, you can change default behaviors for users on the Account settings page. Default behaviors act for all accounts, so you want to make selections that should apply generally to most users.

Where to change the account settings

From the Admin Menu, click the Site Configuration menu link (not an item on the drop-down menu). This link will take you to the main Configuration page. Of the options on this page, find the People section and click on the Account Settings link.

Account registration

Decide who can register an account on your site and how with Account registration settings. On most DKAN sites, an account requires a Site Manager to create the account and login credentials for the user. But there are cases where site visitors should be able to create an account to access certain capabilities.

Who can register accounts? Choose which users register accounts from three options:

- **Administrators only:** With this option, only Site Managers are allowed to add new users to the site and assign roles. This option is best if you expect users to be from within your organization.
- **Visitors:** This option allows site visitors to create an account as soon as they fill out a profile and login to the site with their own password. Visitor accounts automatically assign the lowest-permissions role, Authenticated User. These permissions allow a user to create a profile and leave comments, but they don't have access to any of the content on the site. This option is not recommended unless you have measures like Captcha in place to protect from spamming.
- **Visitors but administrator approval is required: With this option, site visitors can register an account but a Site Manager**
  Require email verification. With this option, users first have to verify their email address before they're allowed to login. Once they verify they will be prompted to change their password. This is an additional option to include in how accounts are registered. This verification can help prevent fake accounts and spamming (recommended).

Automatic email messages

By default, DKAN comes with template responses for certain actions. You can customize these messages with your own text and by using tokens. Tokens are a way to automate certain information. Instead of typing a new username each time for a welcome message, you could simply use the users token. Click on the Browse available tokens link to see all your options.

You can also manage notifications of messages in this menu. You can optionally send a notification when certain actions are taken, but not all of these templates are automatically sent. You'll want to review the email options to make sure the settings meet your needs.

5.6.2 Adding New Users

Adding a user

As a Site Manager, a core piece of your role involves adding users to the site. On your DKAN site you may have generic roles like Authenticated User register for an account that simply needs approval from the Site Manager. But trusted roles with access to content like Content Creators and Editors must be created by a Site Manager. The user can change most of the information on their profile once they access the site, but you'll need to initially provide basic information like a username and password.

You can add new users by clicking on the **People** link and choosing the Create user menu item for quick access or the main People page.

Key information when adding a new user

- **Username:** Create a unique username to create a new user account. The user can change their username once they're logged in as long as it's still unique, but you'll have to choose a name to begin with for the user to first access the site.
- **Email address:** This is how the user will be contacted with notifications about their account and how they can recover a lost password. Choose an email that they are likely to check on a regular basis.

- **Password:** The user should change whatever you originally enter for the password, but you'll need to choose the initial password so that the user can login to their account and change the profile information.
- **Roles:** As you're adding a new user you'll choose which role that person should have from the list of user types detailed in another section. Choosing a role might be obvious in some cases, but in other cases it may be less clear. The role you assign will depend on how much a person needs to do with the site. Higher-level access roles automatically have all the permissions of lower-access roles, but in general we recommend erring on the side of lower-access.

Once you click the **Create new account** button at the bottom the page, the account is created and can now be managed with other existing user accounts.

5.6.3 Managing existing users

Site managers can manage users by clicking 'People' in the Admin Menu. From this screen you can see all existing users, their roles, and details about their account, and by clicking on individual users you can additionally see all the content the user has created. You can also edit their account to change details, add or remove a role, add them to Groups or cancel an account.

To edit an existing user's account:

- Visit your site's User Management page by clicking "People" in the admin menu.
- The displayed list of users on the User Management page can be filtered and sorted using the filters at the top of the page. Once you've found the user you wish to edit in the user table, click the "edit" link at the end of that user's row.
- On the resulting "edit user" page, you can edit the user's username, email, or profile information. You can also set a new password for the user. Click the "Save" button at the bottom of the page to save your changes.
- Use the "Cancel account" at the bottom of the edit user page to delete an account. You will be given the option to preserve or delete any website content added by that user before deletion.

Filtering Users

On some sites, the list of users can be several pages long. To find a specific user or group of users you can use filters to narrow the results. This makes it easy to manage multiple users at the same time or find an individual user without needing to browse through several pages of users.

- **Username** You can enter text into the username filter to limit the results to usernames that contain matching text.
- **E-mail** Filter users by e-mail
- **Active** This filter will limit the results to either Active users or Blocked users.
- **Role** By selecting a role, the results will only show users that are assigned that specific role.

Bulk Operations

When you have several users that require the same action, you can use the **Operations** selector to perform bulk actions on a group of users. Rather than spend extra time performing the same action over and over again for individual users, you can select multiple users and make the change for the group with just one click.

From the People page, check the box next to all of the users you would like to update. Select a task from the Operations drop-down list and click the **Execute** button. You can quickly make changes for a group of users like adding or removing a role and blocking or canceling their accounts.

Blocking a user or canceling an account

At some point, a user account may need to be deleted or blocked. Typically this is for internal employees who move on from the organization, but there are occasions involving external users. There are a number of options for canceling an account or blocking a user to meet a number of scenarios.

Block an account. Blocking an account is the most simple and straightforward way to suspend an account. Blocking a user account keeps a user from logging in, and accounts can easily be unblocked. A blocked account only means that a user cannot login to their account and access your DKAN site. All of their content and profile details will remain, so nothing is lost if you want to unblock an account and restore access.

For users accounts belonging to your organization, blocking an account is typically a temporary action. For user accounts that belong to people who may have registered the account themselves, blocking is likely to be more permanent. By blocking an account, you keep users from creating a new account with the same details and avoid repeating the blocking process.

You can block a single user by editing their profile and changing their status, or you can block several users at once by using the bulk actions function on the People page. In the example below, the Site Manager is blocking a user account by editing an individual user profile. To finalize the changes, the Site Manager must click the Save button at the bottom of the page.

Cancel an account. Canceling an account can be a permanent action, and there are several options to choose from. Some of the actions cannot be reversed, so you should be careful when deciding which option to choose. Below are the options for canceling an account and the implications of selecting the option. While Site Managers can cancel the account of any user on the site, users may also cancel their own accounts.

- **Disable the account and keep its contents:** If you disable the account, the details of the profile remain in tact but the user is blocked from accessing the site with their user login. By keeping the contents, any content that the user published will remain on the live site. Because the account is only disabled (blocked) the user remains as the author of the content and the profile details may still be accessed. This option is similar to just blocking an account, and it's a good temporary measure in most cases.
- **Disable the account and unpublish its contents:** This option blocks the user from accessing the site and all the content that the user has published will be unpublished. This means that their content will not appear on the live site, but it will still exist behind the scenes. It can be managed out of public view and in the mean time, the user cannot do anything else on the site. This is a good option if you need to review the content a user has published and need it to be off the site but still need to access it.
- **Delete the account and make its contents belong to the Anonymous User:** This is a permanent action. Once you delete an account, you cannot recover any of the details that were associated with the user profile. With this option you can delete the entire account as well as keep its contents. Because the account associated with the user who was the original author no longer exists, the content must be assigned to a different author. This option quickly changes the author so that the content remains on the live site, and you can change the author at any time. Again, this is a permanent option so be careful before making this selection.
- **Delete the account and its contents:** This is a permanent action and the most severe choice when canceling an account. This options not only deletes the user account and all the profile details, it also deletes all the content the user added. Neither the account nor the content can be recovered with this selection. As a general best practice, we recommend never deleting content if it can be edited or simply unpublished.

Require email confirmation: For any option you choose when canceling an account, you can make sure the user is aware by requiring email confirmation. An email will be sent to the email address provided in the user's profile details. When you check the Require email confirmation box, the account won't be canceled until the user confirms through the email.

5.6.4 Group Roles and Permissions

With large sites there is often a need to have a subset of the content managed by a specific list of users. Think of a large agency or department with sub-departments or programs that produce content. With DKAN's *groups*, you can silo content and users so that the different departments can easily manage and control only the content they are producing.

Group-level permissions give users the ability to do things they couldn't necessarily do on the site outside of the Group. Read more about *group permissions here*.

Add Group Members

As a site manager, there are two ways that you can add a user to a group: from the user profile, or from the group page.

Add a user by editing their user profile:

- Click "People" in the administration menu at the top
- Click "Edit" on the user's profile who you want to add to the group
- Scroll to the bottom of the page. In the "Group membership" section there are two fields, "your groups" and "other groups".
  - **Your groups:** These are groups that you are a member of. Users are not automatically added to groups, so groups won't appear in this field unless you add yourself to a group.
  - **Other groups:** These are simply groups that you are not a member of. As a site manager, you can add any user to a group regardless if you are a member yourself. But the group names will not automatically appear like in the "your groups" field, so you will have to know the name of the group to enter it in the "other groups" field.
- Once the groups have been selected, click the **Save** button at the bottom to submit the requests.

Add a user from the group page:

- Click the "Group" button at the top of the page
- Click the "Add people" link.
- Begin typing an existing a user into the "user name" field. A list of autocomplete options will appear. A user must already have an account to be added to a group, so if a person needs to be added you should first create a site account for them with the appropriate role.
- By default, a user will only have a Member role in the Group. If you want the new user to be able to manage the content and users of the group, check the "administrator member" box.
- Finish by clicking the "Add users" button at the bottom of the page.

In the example below, the site manager goes to the group "Committee on International Affairs". From the group page the site manager adds a new user by typing the user name and choosing the correction option in from the autocomplete. In this example, Kim Lee should be an administrator member of the group so the site manager checks the administrator member box.

Manage Group members

You can manage group members directly from a group's home page by clicking the "Group" button. From this page you can manage existing members by clicking the "People" link. The Group overview page lists all the members of a group including pending members. From this page you can see how many members are in the group overall, the number of *datasets* associated with the group, access and edit individual member profiles, perform bulk actions and manage membership requests.

Find members: All the members of a group, including pending members, appear on the members list. There are two ways to find members: by "state" and by "name."

- **State:** "State" refers to the status of a member. Active members are users who regularly add *datasets* to the group. Blocked members are unable to add datasets to the group and are not able to request membership. Pending members have requested to join the group and are waiting for approval from the group administrator. Use the "State" drop-down menu to find users who fit a common state. This is helpful when you want to perform bulk actions on multiple users at the same time.
- **Name:** Finding a member by name is a much more specific type of search. You can search for multiple members at the same time by entering the user names in the Name search field separated by commas. This type of search is helpful if you know which specific member you're looking for or if there is a specific group of members that don't have a common state but you want to perform a bulk action on that group of members.

Change a user's role: As a site manager you can change the role of a group member to either be a regular member or an administrator member. Administrator members have full control of the group, its members, and its content so be sure these are trusted users.

From the Group's home page, click the "Group" button and then the "People" link to manage the group members.

Remove or block a member: You can keep users from adding _datasets_ to the group in two ways: blocking a member or removing them from the group.

- **Remove a member:** Removing a member keeps them from adding content to the group, but these users can later request membership. To remove a member, access the group overview page where group members are managed. Find the member you want to remove and click the remove link in the furthest column to the right in the member's row of information.
- **Block a member:** Blocking a user keeps the user from adding content to the group, and these users cannot request membership to the group. To block a member, you'll need to change the status of the member by editing their profile. Find the member you want to block and click the edit link in the furthest column to the right in the member's row of information. From the drop-down Status menu, change the member's status from "active" to "blocked".

Membership requests: Users may also request to join a group. When users submit a membership request, they appear in the list of members with a pending status. If the user entered a request message, it will appear here as part of the member profile (only visible to the site manager and administrator members).

In the example below, a non-member visits the group's home page and submits a request for membership with a request message. The member and her message will now appear on the list of members on the group overview page in a pending state.

The group administrator or site manager can approve the request by changing the user's status from pending to active. To change a member's status, click the edit link in the furthest column to the right in the member's row of information. From the drop-down "status" menu, change the member's status from "pending" to "active."

Membership requests don't send alerts to the group administrator, so the administrator needs to check for members with a pending state. You can find members by state and select pending to show only pending members who need approval.

Bulk actions: Bulk actions help save time because you can select multiple members and perform an action on all of them at once rather than performing the action on each member one at a time.

From the group's home page, click the "Group" button and then the "People" link to manage the Group members.

Find the members you want perform the action on and check the boxes to the left of the member's name for each member. There are three types of bulk actions: "Modify OG user roles," "Remove from group," and "Modify membership status."

- **Modify OG user roles:** OG user roles refer specifically to group roles and permissions, which are "Member" and "Administrator Member." You can change the role of multiple members at the same time with this action. Check all the members whose roles you want to change–for this action, all the members must have the same role to be changed to a new role. Find the drop-down "operations" menu, select "Modify OG user roles," and click the "Execute" button. On the next screen you can choose to either add the "Administrator member" role to the selected users or remove the "Administrator member" role to make the users general members. Click the "Next" button at the bottom of the page and confirm the change on the next screen.
- **Remove from group:** Remove several members at once with this bulk action. Check all the members who you want to remove from the group, find the "Remove from group" option on the drop-down "Operations" menu, and click the "Execute" button. You'll be asked to confirm on the next page and then the members will be removed from the group.
- **Modify membership status:** Members in a group may have one of three statuses: "active," "pending," and "blocked."
  - The "active" status means that the member is able to add datasets to a group and edit datasets that they have created.
  - Pending members are waiting for approval from the group administrator.
  - Blocked members exist but are not active and do not have permission to add datasets to the group or submit another request for membership.
  - Check all the members for whom you want to change the status. For this action, all the members must have the same status to be changed to a new status. Find the drop-down "Operations" menu, select "Modify membership status," and click the "Execute" button. On the next screen, choose the new status and then confirm the changes.

In the example below, the group administrator (Kim Lee) first finds only members with a pending status using the "State" drop-down menu. Then she selects all of the members and chooses the bulk action "Modify membership status" from the drop-down "Operations" menu. On the next screen, she changes the members status to "active." This is an example of an easy way to find and approve membership requests quickly using search functions and bulk actions.

More on Group Membership

Requesting membership and unsubscribing from a Group

Users can request to join a group by clicking the "Request group membership" link on the group home page. A site manager or administrator member must approve the request for the user to become an active member.

Active members, both members and administrator members, can leave the group by clicking the "Unsubscribe From Group" link on the group home page. Once members are removed or leave the group they can no longer add content to the group. Users with the "site manager" role do not need to be members of a group to add content to the group.

Associating Datasets to specific Groups

In general, your group members will publish *datasets* associated with the group, so adding will be more common. On occasion, a *dataset* may be added to a group that does not belong and should be removed (though not deleted).

Add a Dataset: Datasets should be added a group as part of the initial content creation process of the dataset. As group members add new content to the site they should associate the *dataset* with the *group* before finally publishing. Users who created the content can later edit the *dataset* to add to a *group*, and as a site manager you can add any dataset regardless of the author. Once the *dataset* is published and associated with the *group*, it will appear on the group's dataset list. Read the section *Adding a Dataset* to find more detail on how to add a dataset to a group.

Remove a Dataset: Though datasets appear on the group's home page once a dataset is published and associated with the group, they are not managed within the group. To remove a dataset, the content author, administrator member, or site manager needs to edit the *dataset* directly and remove the *group* associated with the *dataset*. Once the *group* is removed from the *dataset* it will no longer appear on the group's dataset list.

5.7 Structure

As a Site Manager you handle the structures that other users operate within. This framework reflects the broader organizational priorities and brings greater visibility to the content on your DKAN site. Many of the tools that help organize the content on your DKAN site also make it easier to navigate and find the right content. All of this improves the experience of a site visitor and makes it more likely that citizens can use the data published on your site.

5.7.1 Site Navigation

Menus

Menus organize content and play a critical role in directing site visitors to the content they need. As Pages are added with new content, adding new menu links are the connectors for site visitors to your content.

Some content (like Datasets or Data Stories) has a menu link in the main menu by default, but new Pages require a new link to navigate to the content. You can customize the default structure of the main menu as it makes sense for your site content.

Using menus

From the Admin Menu, hover over the menu link **Site Configuration**. From the drop-down list, select the **Menus** link to access the Menus page.

The main menu is persistent, meaning that it appears on every page. This menu acts as an anchor for site visitors as they make their way through your site. Details like the order of menu links, link titles, and number of menu links can all impact how a site visitor interacts and engages with the content of your site.

Click the **list links** button to view and manage all the links on the main menu.

As a Site Manager, you can manage the details from the main menu page in the following ways: Add a new link, Order menu links, Enable and disable links, and Edit or delete a menu link.

Add a new link

By default, DKAN comes with preset menu links. These links point to pages that automatically collect content as it's published, like Datasets and Data Stories.

For new Pages with new content, create a link and add it to the main menu by clicking the **Add link** button at the top-left of the page. Complete all the fields in the form and click the Save button at the bottom of the page.

Menu link title The title is what the site visitor sees as they navigate with the main menu.

Path The menu link must point to content, so that when a site visitor clicks the link they are appropriately directed. Paste in the URL for the link.

Description The description essentially acts like alt-text. Make sure you include a description, but keep it brief.

Enabled checkbox By default, when you create a new link, it's also enabled. Only enabled links appear on the main menu. Uncheck this box if the link should not appear on the main menu.

Show as expanded This option only applies to menu links with child menu items. When this box is checked all the child items are displayed.

Parent link By default, the parent link is the main menu. Links with this parent link will appear on the top-level menu bar. Click the Parent link drop-down menu and select the correct parent.

Weight Weight determines the order of menu links. High numbers (heavier) are lower on the menu, and smaller numbers (lighter) appear near or at the top of the menu.

Order menu links

Change the order of how links appear on the main menu with the simple drag-and-drop interface. Click and hold the cross icon to drag a menu link into a new order.

If an asterisk appears next to the title of a menu link it indicates that the menu link has been altered, but changes will not be made until saved.

Note: You can also use the drag-and-drop interface to nest or un-nest menu links.

To nest a link (make a link a child of another menu link), click and hold the cross icon and drag the menu link under and to the right of the new parent menu link.

To un-nest a link (make a child link a top-level link on the main menu), click and hold the cross icon and drag it above the previous parent link.

Click the **Save Configuration** button at the bottom of the page to finalize your changes.

Enable and disable links

Instead of deleting a menu link to remove it from the main menu, you can enable and disable menu links. Disabled links remain in the list for you to edit and manage, but site visitors won't see them on the main menu.

To disable a menu link, uncheck the box in the **Enabled** column in the row of the menu link you want to disable.

Edit or delete links

Change the details of a menu link by clicking the **Edit** link in the Operations column in the row of the menu link you want to edit. When editing a menu link you can change details like the title, the path (URL), and parent link.

Click the **Save** button at the bottom of the page to finalize the changes.

Deleting a menu link is a permanent action and cannot be undone. If you delete a link and want to add it back, you'll have to create it as a new menu link. Disabling a link is recommended before fully deleting a menu link.

To delete a link, click the **Delete** link in the Operations column in the row of the menu link you want to delete.

5.7.2 Adding and editing Topics, Tags, and data formats

Topics

Enhanced user experience, branding, and information architecting–the Topics feature touches on all of these areas to add ease-of-use for both site visitors and users working behind the scenes.

With so much data in a single place, DKAN sites have a number of tools to organize the information for citizens to easily and intuitively find what they are looking for. Topics catch site visitors' attention right from the Home page. As with most DKAN features, Topics are built to provide a crisp, engaging user experience that is adaptable for customization. Either font icons or images can be used, and these graphics can be customized in appearance and color to fit the individual look and feel of any agency. The default DKAN icon font set includes over 100 icons, and new icon font sets can be added anytime for a broad range of options.

Topics typically reflect broad categories like transportation or education. These are areas of interest that should remain consistent themselves and collect timely, relevant content. Topics appear on the front page (unless otherwise modified) along with their assigned icons.

Fig. 5.18: The default DKAN frontpage with highlighted Topics section.

Where Topics are managed

Topics are included in the Structure section of this guide because they're considered a taxonomy. Taxonomies are a high-level way of collecting and organizing information. The smaller pieces of information that make up taxonomies are called terms. So for example, you could have a taxonomy of fruit and individual terms might be "apple" or "banana".

You'll find Topics in the **Taxonomy** drop-down menu item under the **Site Configuration** menu link on the Admin Menu.

Fig. 5.19: Click on the **Topics** link to get to the main page where Topics are managed.

Adding and managing terms

DKAN comes with six preset Topics: Transportation, Education, Finance and Budgeting, Health Care, Public Safety, and City Planning. You can add to the existing terms or replace them with different terms. The Topics for your DKAN site depend largely on the data on the site and the areas of interest you believe your site visitors are most interested in.

Add a term On the Topics page, click on the Add term button to open the options for adding a new term (a single Topic) to be included in the collection of Topics. The only fields that are important for adding a new Topic are the Title field and icon selection. The title is the name of your Topic that will appear on the front page of your DKAN site under the icon you select.

Icon Type: Font

Icon Choose an icon from your font icon library. Whichever you choose, you want something the is both eye-catching and expressive. Site visitors should know exactly what kind of content the Topic includes from the graphic without relying on the text.

Icon Color For font icons, you can choose a color for the icon of your Topic. Images do not have an option to change the color. It's best to choose a single color that all your icons have in common. The color should also fit the look and feel of the rest of your site, either as a matching or complementary color.

Icon Type: Image

If you have an image that you prefer to use for the Topic icon, select the Image selection instead of font icon. Click the **Choose File** button to find the image on your computer and then click the Upload button to finalize the selection.

Note: there are limits on the file size and type.

Fig. 5.20: In this example, the Site Manager is adding a new term to the Topics taxonomy called "Environment". The Site Manager chooses an icon from the library and then chooses a color for the icon. Once the Topic is saved it will appear on the front page.

When all your selections are made, click the **Save** button at the bottom of the page to finalize your choices and save the term. Once the new Topic is saved it will appear on the front page with all the other Topics.

It's best to keep your site looking orderly, so you don't want Topics to appear like they do in the image below. For the purposes of this example, a different color was chosen to demonstrate the change, but in practice color schemes should remain consistent across your site.

Additionally, the number of Topics pushes down to a second line and appears uneven. Keep the number of Topics even appearance by maintaining a certain number of Topics at any given time. Since there are six Topics to a line, that means the number of Topics should follow a factor of six.

Manage terms

As the appearance of your DKAN site evolves, you may need to also change the appearance and even titles of your existing Topics. It's best to edit existing Topics rather than create completely new ones. If you delete a Topic then all the content that was before associated with the Topic will no longer have any association because the Topic doesn't exist. But if you update a Topic, then the content will still be associated with the Topic with the updated title and icon.

You can change the title, icon, and icon color of existing terms by clicking the edit link under the Operations column on the row of the Topic you want to edit. The only fields that are important for adding a new Topic are the Title field and icon selection. The title is the name of your Topic that will appear on the front page of your DKAN site under the icon you select.

Re-order Topics

While many site visitors will be attracted to a Topic based on its visual element (the icon) many site visitors will read Topics from left to right like with text. You might have some Topics that you want to promote to ensure that site visitors don't miss them. You can re-order Topics to follow the rank of importance to the site visitor.

Notice the compass arrow to the left of each Topic name. You can change the order of the Topics by clicking and holding the compass arrow, then dragging and dropping each Topic either higher or lower on the list. The order determines how the Topics appear on the front page of your DKAN site. The first Topic on the list appears as the first Topic on the front page. It appears on the far left and the rest follow horizontally across the page.

Fig. 5.21: Administrative view of the re-ordered Topics.

Fig. 5.22: Front page view of the re-ordered Topics.

Customizing icon options

A consistent appearance on your DKAN site that aligns with the image of your overall organization is important. While your DKAN site is dedicated to data publishing, getting your data into a central location, and increasing transparency to engage with the broader public, it should be clear to site visitors that the effort is part of the larger organizational priorities. If you do use the default font icon library, you can also manage how the library is used.

Uploading new font icon libraries

To add greater flexibility in the appearance of Topics to align with your organizational image, you can add your own font icon libraries to choose from for your Topics. By default, DKAN comes with a font icon library called DKAN Topics. It includes over 100 icons to choose from for your Topics. Alternatively, you can also upload your own font icon library if you have one that you prefer.

You'll need to name your library with the Title field to manage it among all the uploaded font libraries. Uploading a library also requires four standard files that make a font: EOT, SVG, TTF, and WOFF. If you don't have these files then you'll either need to get the files or use the default DKAN Topics font.

Editing existing font icon libraries

Once a font icon library is uploaded, you can go back and make edits to an individual library under the Uploaded Font Libraries tab.

Edit Font Options

Click the edit font link to change the details of the font icon library like the title and the files containing the visual elements of the library.

Edit Font Blacklist

Your font icon library may contain hundreds of icons, but there might be some icons that you don't want users to assign to Topics. You can add unwanted icons to a blacklist to block those icons from appearing as an option. Edit the font library blacklist to choose which icons cannot be assigned to Topics. Click on an icon or remove an icon from the blacklist by click a red icon.

Delete a library

You can delete an entire library from your site with one click. If a font icon library is in use it can't be deleted, so you'll need to change the active library before deleting a library. Once a library is deleted, the action cannot be undone so be careful when removing font icon libraries.

5.7.3 Adding and managing Tags

As content is added by users of all types, authors can add Tags to their content. Tags are free-form, so they can be newly added in the field and can contain any words. Think of Tags as keywords either within or related to the content. By including Tags on your content it will appear when the terms are included in a search.

Fig. 5.23: Tags field on content edit pages.

Adding Tags to content

Users can add Tags to Datasets and Data Stories, whether they have been added before or not. Simply type the key term you want to tag your content with and hit the space key on your keyboard.

Tags are added in single terms, so if you have multiple or compound words they will either have to be combined for a single term or separated with dashes. If a Tag has been used before, DKAN will autocomplete the term for you. If the Tag has not been added previously, then a new Tag will be created and can be used in the future. As a Site Manager, you can add Tags without adding them to content on the Tags menu.

Fig. 5.24: Administrative view of the tags list.

Managing Tags

Because Tags are free-form and can be newly added by any user, you may end up with many unnecessary Tags that are counterproductive. Too many Tags can actually make it more difficult to sort through data and confuse site visitors.

As a Site Manager, you have access to all the terms collected as Tags. You'll want to keep an eye on your Tags to make sure they stay orderly and relevant. You can edit or remove redundant or incorrect terms from the Tags menu under the **Site Configuration** link on the Admin Menu.

Fig. 5.25: In this example, the Site Manager is deleting a partial (and redundant) Tag as well as editing a Tag that is useful, but spelled incorrectly. This will help users avoid associating their content with the wrong Tags.

5.7.4 Data formats

Data added to your DKAN site may come in a number of file formats. You can think of data, or file formats, as labels to tell site visitors what type of data is contained in the file–adding a data format does **not** change or transform the contents of the files themselves. File formats describe how the data is formatted and indicates how it should be read.

In some cases file formats may be similar like CSV or XLS, which both contain information organized into rows and columns. But in many cases, indicating the file format makes an important difference. Geospatial data formatted in an XML file is read differently than similar data formatted in a JSON file. Data formats allow users to choose the right data format and tell site visitors which type of data they should expect when they download the file.

Add data formats

DKAN comes standard with the most common data formats, but you can add more file formats as needed. File formats fall under the **Taxonomy** menu item on the Admin Menu. Use the Add term button to include an additional file format. Keep in mind that data formats are just labels to tell site visitors what type of data is contained in the file–adding a data format does **not** change or transform the contents of the files themselves.

Manage data formats

If the language used for the data formats changes, you can edit the name of the format rather than delete the old one and add a new one. This is important because any data content that has the old data format will require a new label to tell site visitors what type of format the data is in. If instead you update the existing data format by editing it, all the data content with the old data format with update automatically.

5.7.5 URL aliases

Though it may not be obvious, URL aliases are important to the overall experience a site visitor has even before they get to your DKAN site. Because every URL must be unique to work properly by default most URLs are a string of arbitrary numbers, symbols and letters.

This serves the purposes of creating a unique URL, but they can be hard to remember, too long to share, and cause alarm about the credibility of the source. Aliases let you create a URL that is easy to read, understand, and share while still maintaining a unique location on the web.

Managing URL aliases

On DKAN, URL aliases are automatically generated for content, unless you (as a Site Manager) specifically unselect that option on a piece of content. Rather than using the automated URL path that is generated for technical administration (usually an indiscernible cluster of numbers of letters) an alias is created that is easy to read, remember, and type into the search bar. On DKAN, the alias is typically the title of a piece of content.

While aliases are created at the time content is generated, you can also change the URL alias any time. You might find some terms make more sense or appear more frequently in site visitor searches. You can also delete the alias and leave only the administrative title for a piece of content (not recommended).

From the Admin Menu, click the the **Site Configuration** menu item and find the Search and metadata section. Select the **URL aliases** link to manage URL aliases.

Options in the URL aliases page under the Search and metadata section show a list of the existing URL aliases and gives you options for either changing the alias or deleting it. In general, you won't need to make bulk updates or delete groups of aliases, so we recommend using the List button when making changes.

Find the URL that you want to change and click the edit link in the column furthest to the right of the URL. Only change the Path alias field; choose a name that is easy to remember, relatively brief so that it's easy to share, and a name that is engaging and tells a site visitor what they'll be reading.

URL aliases are an excellent opportunity to improve how the content on your DKAN site is discovered because you can include key search terms that commonly appear in site visitors' searches.

Fig. 5.26: In this example, the Site Manager is editing the URL of a Dataset to make it more specific so that it appears in searches for bike lanes in Miami.

CHAPTER 6

Extending and Customizing DKAN

Much additional functionality can be added simply by installing one of the tens of thousands of contrib modules from the Drupal community. However, as a Drupal Distribution DKAN is a flexible framework which developers can also build off of and add to.

DKAN consists of of a distribution profile which manages the initial installation, 3rd party libraries and drupal modules, and DKAN specific modules.

Below is a simplified version of where the DKAN code sits within the fully packaged version:

```
profiles/
    dkan/
      libraries/ (3rd party libraries)
      modules/
         dkan/ (dkan modules)
         contrib/ (3rd party module dependencies)
      themes/ (dkan themes)
sites/
   all/
      libraries/ (your libraries)
      modules/ (your modules)
      themes/ (your themes)
```

After installing DKAN additional functionality should be added to the "sites" directory.

In the future, this section will feature more detailed information on developing custom extentions to DKAN. For now, read additional information about:

6.1 Customizing the License Field

In order to add options to the existing ones you need to implement *hook_license_subscribe* in the following fashion:

```
// Let's assume we want to do this as part of the fictitious license_options_extra␣
↪module
function license_options_extra_license_subscribe() {
  return array(
    'tcl' => array(
      'label' => 'Talis Community License (TCL)',
      'uri' => 'http://opendefinition.org/licenses/tcl/',
    ),
  );
}
```

The code above add the **Talis Community License (TCL)** license referencing it to the **tcl** key. It also provides a link to the license (optional). You can provide as many options as you want through the array being returned.

6.1.1 Removing License Options

In order to remove options from the existing ones you need to implement *hook_license_unsubscribe* in the following fashion:

```
// Let's assume we want to do this as part of the fictitious license_options_extra␣
↪module
function license_options_extra_license_unsubscribe() {
  return array(
    'notspecified',
  );
}
```

The code above removes the **notspecified** option. You can provide as many options as you want through the array being returned.

6.1.2 Additional notes about the behavior of both hooks

- The options provided through the license drupal field configuration are **COMPLETELY** ignored.
- `hook_license_subscribe()` implementations are of course called before `hook_license_unsubscribe()` implementations.
- Options subscribed through `hook_license_subscribe()` are processed as they come through the order of modules provided by the drupal registry.
- If multiple options are provided using the same key then it grabs the first one that comes in and ignores the rest
- If you want to **replace** and item that already exists, unsubscribe the existing key and provided an alternative one for your option

6.1.3 References to the code

- Hooks are invoked in dkan_dataset_content_types.license.field.inc
- Field formatter implementation for the license field is in dkan_dataset_content_types.module

6.2 Community Contributions

What follows is a list of modules from the community that extend DKAN's functionality. Many are developed by the DKAN team themselves but are still in an expiremental stage and not part of the core DKAN software. Contact us to add your mdodule to this list!

- DKAN Datastore CartoDB Integration

6.3 Metadata Source Module

6.3.1 Summary

This documentation is for a module that is not part of the DKAN distribution but can be added to an existing DKAN site. This module, along with further documentation, is here: DKAN Metadata Source

6.3.2 DKAN Metadata Source

Metadata is the "Who, What, When, Where, Why" of each dataset and its associated resources. When data contributors make sure to provide appropriate and thorough information for each dataset, users will have an easier time understanding the source and purpose of each dataset, and they can more easily plug it into their application of choice.

How is metadata standardized?

The International Organization for Standardization (ISO) is an global standard-setting body composed of representatives from various national standards organizations, and has determined a wide array of specific protocols for various types of data. Other standards setting organizations include the U.S. Federal Geographic Data Committee (FGDC) and the European INSPIRE Metadata Directive.

What is geospatial metadata?

Geospatial metadata pertaining to geographic digital resources such as Geographic Information System (GIS) files, geospatial databases, and earth imagery must involve core library catalog elements including, but not limited to, Title, Abstract, and Publication Data; geographic elements such as Geographic Extent and Projection Information; and database elements such as Attribute Label Definitions and Attribute Domain Values. (Source: U.S. FGDC)

Federal agencies in the U.S. are encouraged to follow ISO standards when working with or uploading geospatial data. For more information, see the U.S. FGDC's geospatial metadata documentation.

The majority of geospatial data tools such as ESRI ArcGIS and GeoNetwork allow users to export metadata content to a stand-alone XML file that is formatted correctly for each standard or profile, and can be validated using the appropriate XML schema. The exported XML file can also then be published to a metadata catalog such as geodata.gov.

How can I ensure that my datasets and resources are associated with their proper metadata under these standards?

When working with metadata created outside of DKAN, it would be tedious to import it with a 1 to 1 field ratio in DKAN Dataset given that you'd have to take the time to manually add each additional field for your standard of choice.

For example, the ISO 19115 metadata specification has dozens of fields that go above and beyond the Project Open Data fields included in DKAN.

Fortunately, the DKAN metadata source module was built to solve this dilemma.

Installation

Install like any other Drupal module. Once enabled you will have the new content type and taxonomy term.

Metadata Sources and Types

The DKAN metadata source module creates a Metadata Source content type and a Metadata Type taxonomy term.

Creating a Metadata Type

The Metadata Type is a taxonomy term used for linking multiple metadata sources with the same specification. To add or remove Metadata Types visit */admin/structure/taxonomy/extended_metadata_schema*.

Click Add term to create a new type or Edit on an existing type to update or modify it.

Creating a Metadata Source

The Metadata Source content type allows DKAN datasets and resources to be associated with metadata that has either been directly uploaded or linked from an outside source.

Creating a Metadata Source

To create a new Metadata Type visit `node/add/metadata`:

Link on Dataset

Once a Metadata Type has been created the Metadata Source will be displayed on the linked Dataset:

Clicking 'View' will allow you to check out a preview of the metadata content; clicking the 'Download' button will allow you to download it directly.

6.4 Adding additional file types to the *resource* allowed extensions list

Many different file types can be uploaded on a *resource* (though **only** CSV files can be imported to the *Datastore*.

To view the allowed file extensions that come with DKAN visit `admin/dkan/dataset_forms`.

To add additional file types

1. Navigate to DKAN > DKAN Dataset Forms
2. Enter the extensions into the "Additional allowed file extensions" field
3. Click "Save configuration".

You can also add additional extensions in a custom module or with drush by using variable_set().

```
variable_set('dkan_custom_extensions', 'ods dct');
```

6.5 Display Number of Datasets, Topics or Other Items

Datasets, resources, tags and topics in DKAN are created using Drupal's content types and vocabularies. This means that we can use many of the Drupal modules and tools to extend the site display and functionality.

The Views module is one of the most popular and powerful Drupal modules. It provides a user interface to query content and other site assets and choose display options without any code.

The following recipe will use Views and Panals modules to create some simple site statistics and display them on the home page. The end result will look like:

6.5.1 Step 1

- Go to *admin/structure/views/import* and import the following two views:

Terms Count View:

```
$view = new view();
$view->name = 'term_counts';
$view->description = '';
$view->tag = 'default';
$view->base_table = 'taxonomy_term_data';
$view->human_name = 'Term Counts';
$view->core = 7;
$view->api_version = '3.0';
$view->disabled = FALSE; /* Edit this to true to make a default view disabled␣
↪initially */

/* Display: Master */
$handler = $view->new_display('default', 'Master', 'default');
$handler->display->display_options['title'] = 'Tags';
$handler->display->display_options['use_more_always'] = FALSE;
$handler->display->display_options['group_by'] = TRUE;
$handler->display->display_options['access']['type'] = 'perm';
$handler->display->display_options['cache']['type'] = 'none';
$handler->display->display_options['query']['type'] = 'views_query';
$handler->display->display_options['exposed_form']['type'] = 'basic';
$handler->display->display_options['pager']['type'] = 'none';
$handler->display->display_options['pager']['options']['offset'] = '0';
$handler->display->display_options['style_plugin'] = 'default';
$handler->display->display_options['row_plugin'] = 'fields';
/* Field: COUNT(DISTINCT Taxonomy term: Name) */
$handler->display->display_options['fields']['name']['id'] = 'name';
$handler->display->display_options['fields']['name']['table'] = 'taxonomy_term_data';
$handler->display->display_options['fields']['name']['field'] = 'name';
$handler->display->display_options['fields']['name']['group_type'] = 'count_distinct';
$handler->display->display_options['fields']['name']['label'] = '';
$handler->display->display_options['fields']['name']['alter']['word_boundary'] =␣
↪FALSE;
$handler->display->display_options['fields']['name']['alter']['ellipsis'] = FALSE;
```

```
$handler->display->display_options['fields']['name']['element_type'] = 'h1';
$handler->display->display_options['fields']['name']['element_label_colon'] = FALSE;
$handler->display->display_options['fields']['name']['element_default_classes'] =␣
↪FALSE;
/* Filter criterion: Taxonomy vocabulary: Machine name */
$handler->display->display_options['filters']['machine_name']['id'] = 'machine_name';
$handler->display->display_options['filters']['machine_name']['table'] = 'taxonomy_
↪vocabulary';
$handler->display->display_options['filters']['machine_name']['field'] = 'machine_name
↪';
$handler->display->display_options['filters']['machine_name']['value'] = array(
  'tags' => 'tags',
);

/* Display: Topics Block */
$handler = $view->new_display('block', 'Topics Block', 'block_1');
$handler->display->display_options['defaults']['title'] = FALSE;
$handler->display->display_options['title'] = 'Topics';

/* Display: Tags Block */
$handler = $view->new_display('block', 'Tags Block', 'block_2');
```

Node Counts view:

```
$view = new view();
$view->name = 'nodecounts';
$view->description = '';
$view->tag = 'default';
$view->base_table = 'node';
$view->human_name = 'Node counts';
$view->core = 7;
$view->api_version = '3.0';
$view->disabled = FALSE; /* Edit this to true to make a default view disabled␣
↪initially */

/* Display: Master */
$handler = $view->new_display('default', 'Master', 'default');
$handler->display->display_options['title'] = 'Datasets';
$handler->display->display_options['use_more_always'] = FALSE;
$handler->display->display_options['group_by'] = TRUE;
$handler->display->display_options['access']['type'] = 'perm';
$handler->display->display_options['cache']['type'] = 'none';
$handler->display->display_options['query']['type'] = 'views_query';
$handler->display->display_options['query']['options']['query_comment'] = FALSE;
$handler->display->display_options['exposed_form']['type'] = 'basic';
$handler->display->display_options['pager']['type'] = 'none';
$handler->display->display_options['style_plugin'] = 'default';
$handler->display->display_options['row_plugin'] = 'fields';
$handler->display->display_options['row_options']['inline'] = array(
  'type_1' => 'type_1',
  'type' => 'type',
);
$handler->display->display_options['row_options']['separator'] = ': ';
/* Field: COUNT(Content: Type) */
$handler->display->display_options['fields']['type']['id'] = 'type';
$handler->display->display_options['fields']['type']['table'] = 'node';
$handler->display->display_options['fields']['type']['field'] = 'type';
$handler->display->display_options['fields']['type']['group_type'] = 'count';
```

```
$handler->display->display_options['fields']['type']['label'] = '';
$handler->display->display_options['fields']['type']['element_type'] = 'h1';
$handler->display->display_options['fields']['type']['element_label_colon'] = FALSE;
$handler->display->display_options['fields']['type']['element_default_classes'] =␣
↪FALSE;
$handler->display->display_options['fields']['type']['separator'] = '';
/* Filter criterion: Content: Type */
$handler->display->display_options['filters']['type']['id'] = 'type';
$handler->display->display_options['filters']['type']['table'] = 'node';
$handler->display->display_options['filters']['type']['field'] = 'type';
$handler->display->display_options['filters']['type']['value'] = array(
  'dataset' => 'dataset',
);

/* Display: Datasets Block */
$handler = $view->new_display('block', 'Datasets Block', 'block');

/* Display: Resources Block */
$handler = $view->new_display('block', 'Resources Block', 'block_1');
$handler->display->display_options['defaults']['title'] = FALSE;
$handler->display->display_options['title'] = 'Resources';
$handler->display->display_options['defaults']['filter_groups'] = FALSE;
$handler->display->display_options['defaults']['filters'] = FALSE;
/* Filter criterion: Content: Type */
$handler->display->display_options['filters']['type']['id'] = 'type';
$handler->display->display_options['filters']['type']['table'] = 'node';
$handler->display->display_options['filters']['type']['field'] = 'type';
$handler->display->display_options['filters']['type']['value'] = array(
  'resource' => 'resource',
);
```

These two views provide blocks for Dataset, Resource, Topic, and Tag counts.

- Clear the Drupal cache.

6.5.2 Step 2

Add Blocks to Home page:

- Go to the Panelizer page for the home page. You can find this by finding the "Welcome" page in the content menu and clicking "edit". Replace the "edit" term in the URL with "panelizer". The URL should look like *node/1/panelizer*.
- Click "content" in the default view mode.
- Click the gear Icon under "Triplet First Column" and "Add Content"

Click "Views" in the left column and then "Node Counts"

Select "Dataset Block" as the display. Click "Finish" and then "Save" on the panelizer page.

- Repeat the process for the other types of statistics you want to add. To add Topics or Tags statistics select "Term Counts" instead of "Node Counts".

6.6 Developing with local Solr server

DKAN is shipped with a search powered by the database, though it can be used in production environments using a Solr server. Now, DKAN also provides the module DKAN Custom Solr, by enabling this module, you'll be able to use a Solr server in your local development.

This module depends on *Search API Solr* module and it provides the settings for creating a local development Solr server using DKAN Tools and will update the datasets and groups search indexes provided by DKAN.

Note: this server will be created using settings based on the DKAN Tools Solr container, if you are not using DKAN Tools, you'll still be able to use the module but may need to update the respective configuration in `admin/config/search/search_api/server/dkan_custom_solr/edit`.

If you want to stop using the local Solr server provided by this module, simply disable this module and revert the indexes to their original config in `admin/config/search/search_api`.

6.7 DKAN Feedback

The DKAN Feedback module allows site visitors to give feedback on published datasets. This module is not included in DKAN core, but it can be added as an additional feature.

Requirements

- Full installation of core DKAN >=7.x-1.13.
- Dependencies include Captcha, Custom Publishing Options, Rate, reCAPTCHA, and Voting API.

6.7.1 Installation

Using drush make

```
cd sites/all/modules/contrib
git clone https://github.com/GetDKAN/dkan_feedback
drush make --no-core sites/all/modules/contrib/dkan_feedback/dkan_feedback.make
drush en dkan_feedback
```

Manual installation

- Download the zip file from https://github.com/GetDKAN/dkan_feedback
- Unzip the file in `/sites/all/modules/contrib`
- Download all dependent contrib modules from the Requirements list above and add them to `/sites/all/modules/contrib`
- `drush en dkan_feedback -y`

6.7.2 Usage

Once enabled, a "Feedback" menu item will be added to the main menu. Clicking this link will display a list of all published feedback.

Besides giving feedback, visitors can also "up vote" or "down-vote" feedback provided by other users, and leave comments on published feedback.

Multiple search and sort options are provided on the primary Feedback page, by default, feedback is listed from newest to oldest.

6.7.3 Administration

As feedback does not require authentication, it is important for site managers to stay on top of new entries, captcha is not fool-proof.

From the administration menu, a site manager can access the feedback and comment administration dashboards to bulk publish, unpublish, or archive feedback and comment content. For agencies that are required to NOT delete content, the 'Archive' option allows site managers to avoid repeated moderation of the same unpublished content.

6.7.4 Providing feedback on DKAN Feedback

Submit contributions, bug reports or enhancement requests on Github at https://github.com/GetDKAN/dkan_feedback.

CHAPTER 7

API Guide

DKAN includes a number of APIs to allow it to communicate with external applications.

7.1 Dataset REST API

The DKAN Dataset REST API uses the Services module to create CRUD endpoint at `api/dataset/node`. By default, this endpoint provides full CRUD access to a website's content nodes, and limited access to users (to allow authentication). The endpoint can be customized at `/admin/structure/services/list/dkan_dataset_api/resources`.

7.1.1 Services Documentation

The DKAN Dataset API module is only a light wrapper around the Services module, which has extensive documentation. Here are some entries of interest:

- Testing Resources
- Identifying field names
- Using REST Server with 2-Legged OAuth Authentication (Example with Java Servlet)
- Services CSRF Token with FireFox Poster

The Sessions module also has a thriving community on the Drupal Stack Exchange.

By visiting */admin/structure/services/list/dkan_dataset_api/resources*, you can inspect every resource made available by default via the Dataset REST API.

7.1.2 Server Types

DKAN Dataset REST API comes with a REST Server. Other server types are also incldued in the Services module but not turned on. Those include:

- OAUTH
- XML-RPC

7.1.3 Response Formats

- bencode
- json
- jsonp
- php
- xml
- yaml

7.1.4 Request Parser Types

- application/json
- application/vnd.php.serialized
- application/x-www-form-urlencoded
- application/x-yaml
- application/xml
- multipart/form-data
- text/xml

7.1.5 Authentication Modes

Session Authentication

Session authentication is enabled by default. With session authentication an inital request is made to the user login to request a session cookie. That session cookie is then stored locally and sent with a request in the X-CSRF-Token header to authenticate the request.

Token Authentication

Token authenticaion is not currently available out-of-the-box. However, it can be enabled by adding the Services Token Access module to your site. This is less secure but is easier for community members to use, and may be added to the DKAN distribution in a future release.

Authentication Permissions

The permissions with which a user is granted depend on the user role. User roles and permissions are easily configured in the user administration screen at `admin/people`, and DKAN comes with a number of pre-configured default roles via the DKAN Permissions module.

7.1.6 Request Examples

URL Query Parameters

- (string) fields - A comma separated list of fields to get.
- (array) parameters - Filter parameters array such as parameters[title]="test"
- (int) page - The zero-based index of the page to get, defaults to 0.
- (int) pagesize - Number of records to get per page (max = 20).

The output from a url is paginated and displays 20 items per page by default. You can specify which page to view by adding a *page* value. And you can change the number of items per page by including a *pagesize* value. For example, the following query will display items 11 - 20:

```
http://demo.getdkan.com/api/dataset/node.json?page=2&pagesize=10
```

To return only a specific node type, include a *type* parameter:

```
http://demo.getdkan.com/api/dataset/node.json?parameters[type]=dataset
```

To return only the title and node id of published resources:

```
http://demo.getdkan.com/api/dataset/node.json?fields=title,nid&
↪parameter[type]=resource&parameter[status]=1
```

Below you can find examples in PHP for a basic set of CRUD operations on datasets and resources. This documentation is a work in progress. The examples are raw HTTP requests, with a short example of how to execute a query in PHP as well.

For an example of a fully-functional python-based client to the DKAN REST API, see the pydkan project.

Login

Request

```
POST http://docker:32774/api/dataset/user/login
Accept: application/json
Content-Length: 29
Content-Type: application/x-www-form-urlencoded

{
  "username": "admin",
  "password": "admin"
}
```

Response

```
{
    "sessid": "OBoeXKMQx3zmaZrS_v3FOP7_Ze66fYA61TGhtm9s0Qk",
    "session_name": "SESSd14344a17ca11d13bda8baf612c0efa5",
    "token": "C2dfCUcN4hjgt6u2Xmv15mc1nkj5uv1Iqpa8QE3d7E8",
    "user": {
        "access": "1492546345",
```

```
        "created": "1488377334",
        "data": false,
        "field_about": [],
        "init": "admin@example.com",
        "language": "",
        "login": 1492546519,
        "mail": "admin@example.com",
        "name": "admin",
        "og_user_node": {
            "und": [
                {
                    "target_id": "1"
                },
                {
                    "target_id": "2"
                },
                {
                    "target_id": "3"
                }
            ]
        },
        "picture": null,
        "roles": {
            "2": "authenticated user"
        },
        "signature": "",
        "signature_format": null,
        "status": "1",
        "theme": "",
        "timezone": "UTC",
        "uid": "1",
        "uuid": "5eb4da39-cfba-4d43-bb45-691ebcde8f70"
    }
}
```


Retrieve session token

Request

```
POST http://docker:32774/services/session/token
Accept: application/json
Cookie: SESSd14344a17ca11d13bda8baf612c0efa5=OBoeXKMQx3zmaZrS_v3FOP7_
↪Ze66fYA61TGhtm9s0Qk
Content-Length: 0
```

Response

```
XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
```

Create dataset

Request

```
POST http://docker:32774/api/dataset/node
Content-Type: application/json
Accept: application/json
X-CSRF-Token: XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
Cookie: SESSd14344a17ca11d13bda8baf612c0efa5=OBoeXKMQx3zmaZrS_v3FOP7_
↪Ze66fYA61TGhtm9s0Qk
Content-Length: 44

{
  "type": "dataset",
  "title": "Test Dataset"
}
```

Response

```
{
    "nid": "75",
    "uri": "http://docker:32774/api/dataset/node/75"
}
```

Delete dataset

Request

```
DELETE http://docker:32774/api/dataset/node/123
Content-Type: application/json
X-CSRF-Token: XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
```

Response

Create resource

Request

```
POST http://docker:32774/api/dataset/node
Content-Type: application/json
Accept: application/json
X-CSRF-Token: XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
Cookie: SESSd14344a17ca11d13bda8baf612c0efa5=OBoeXKMQx3zmaZrS_v3FOP7_
↪Ze66fYA61TGhtm9s0Qk
Content-Length: 97

{
  "type": "resource",
  "field_dataset_ref": {"und": [{"target_id": "75"}]},
  "title": "Test Resource"
}
```

Response


```
{
    "nid": "76",
    "uri": "http://docker:32774/api/dataset/node/76"
}
```


Retrieve parent dataset to check resource ID

Request

```
GET http://docker:32774/api/dataset/node/75
Accept: application/json
X-CSRF-Token: XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
Cookie: SESSd14344a17ca11d13bda8baf612c0efa5=OBoeXKMQx3zmaZrS_v3FOP7_
↪Ze66fYA61TGhtm9s0Qk
```

Response


```
{
    "body": [],
    "changed": "1492544349",
    "comment": "0",
    "created": "1492544348",
    "data": "b:0;",
    "field_additional_info": [],
    "field_author": [],
    "field_conforms_to": [],
    "field_contact_email": [],
    "field_contact_name": [],
    "field_data_dictionary": [],
    "field_data_dictionary_type": [],
    "field_frequency": [],
    "field_granularity": [],
    "field_harvest_source_ref": [],
    "field_is_part_of": [],
    "field_landing_page": [],
    "field_language": [],
    "field_license": {
        "und": [
            {
                "format": null,
                "safe_value": "notspecified",
                "value": "notspecified"
            }
        ]
    },
    "field_modified_source_date": [],
    "field_orphan": {
        "und": [
            {
                "value": "0"
            }
```

```
        ]
    },
    "field_pod_theme": [],
    "field_public_access_level": {
        "und": [
            {
                "value": "public"
            }
        ]
    },
    "field_related_content": [],
    "field_resources": {
        "und": [
            {
                "target_id": "76"
            }
        ]
    },
    "field_rights": [],
    "field_spatial": [],
    "field_spatial_geographical_cover": [],
    "field_tags": [],
    "field_temporal_coverage": [],
    "field_topic": [],
    "language": "und",
    "log": "",
    "name": "admin",
    "nid": "75",
    "og_group_ref": [],
    "path": "http://docker:32774/dataset/test-dataset-16",
    "picture": "0",
    "promote": "0",
    "revision_timestamp": "1492544349",
    "revision_uid": "1",
    "status": "1",
    "sticky": "0",
    "title": "Test Dataset",
    "tnid": "0",
    "translate": "0",
    "type": "dataset",
    "uid": "1",
    "uuid": "d53881b3-d80f-49c2-8815-897321fe926e",
    "vid": "117",
    "vuuid": "c4663ada-0162-4780-8ee5-347c6c037429"
}
```


Note: The correct resource ID, *76*, has been added to *field_resources*.

Update dataset

Request

```
PUT http://docker:32774/api/dataset/node/75
Content-Type: application/json
```

```
Accept: application/json
X-CSRF-Token: XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
Cookie: SESSd14344a17ca11d13bda8baf612c0efa5=OBoeXKMQx3zmaZrS_v3FOP7_
↪Ze66fYA61TGhtm9s0Qk
Content-Length: 34

{"title": "Updated dataset title"}
```

Response

```
{
    "nid": "75",
    "uri": "http://docker:32774/api/dataset/node/75"
}
```

Add a file to a resource

Request

```
POST http://docker:32774/api/dataset/node/76/attach_file
Accept: application/json
X-CSRF-Token: XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
Cookie: SESSd14344a17ca11d13bda8baf612c0efa5=OBoeXKMQx3zmaZrS_v3FOP7_
↪Ze66fYA61TGhtm9s0Qk
Content-Length: 474
Content-Type: multipart/form-data; boundary=5f8b431c36a24c828044cd876b3aa669

--5f8b431c36a24c828044cd876b3aa669
Content-Disposition: form-data; name="attach"

0
--5f8b431c36a24c828044cd876b3aa669
Content-Disposition: form-data; name="field_name"

field_upload
--5f8b431c36a24c828044cd876b3aa669
Content-Disposition: form-data; name="files[1]"; filename="tension_sample_data.csv"

tension,current,timestamp
220,10,2016-05-27T19:56:41.870Z
50,15,2016-05-27T19:51:21.794Z
230,10,2016-05-27T19:56:41.870Z
--5f8b431c36a24c828044cd876b3aa669--
```

Note: Setting the `attach` parameter to `0` ensures that the file will replace any existing file on the resource. Setting it to *1* will result in a rejected request if the resource already has an attached file (but it will work if the resource's file upload field is empty).

Response


```
{
  "fid": "38",
  "uri": "http://docker:32774/api/dataset/file/38"
}
```


Retrieve the resource to check the file field

Request

```
GET /api/dataset/node/76
Accept: application/json
X-CSRF-Token: XBWI44XD33XBIANLpyK-rtvRa0N5OcaC03qLx0VQsP4
Cookie: SESSd14344a17ca11d13bda8baf612c0efa5=OBoeXKMQx3zmaZrS_v3FOP7_
↪Ze66fYA61TGhtm9s0Qk
```

Note: We still use the */api/dataset* endpoint to retrieve a resource node (or any other type of node) by nid.

Response


```
{
    "body": [],
    "changed": "1492544350",
    "comment": "0",
    "created": "1492544349",
    "data": "b:0;",
    "field_dataset_ref": {
        "und": [
            {
                "target_id": "75"
            }
        ]
    },
    "field_datastore_status": {
        "und": [
            {
                "value": "2"
            }
        ]
    },
    "field_format": {
        "und": [
            {
                "tid": "32"
            }
        ]
    },
    "field_link_api": [],
    "field_link_remote_file": [],
    "field_orphan": {
```

```
        "und": [
            {
                "value": "0"
            }
        ]
    },
    "field_upload": {
        "und": [
            {
                "alt": "",
                "delimiter": null,
                "embed": null,
                "fid": "38",
                "filemime": "text/csv",
                "filename": "tension_sample_data.csv",
                "filesize": "120",
                "graph": null,
                "grid": null,
                "map": null,
                "metadata": [],
                "service_id": null,
                "status": "1",
                "timestamp": "1492544350",
                "title": "",
                "type": "undefined",
                "uid": "1",
                "uri": "public://tension_sample_data.csv",
                "uuid": "87019111-7ef0-48e5-b8a4-ea2c392f2e56"
            }
        ]
    },
    "language": "und",
    "log": "",
    "name": "admin",
    "nid": "76",
    "og_group_ref": [],
    "path": "http://docker:32774/dataset/updated-dataset-title/resource/34b45055-bc10-
↪416f-a8ba-8b9f27718362",
    "picture": "0",
    "promote": "0",
    "revision_timestamp": "1492544350",
    "revision_uid": "1",
    "status": "1",
    "sticky": "0",
    "title": "Test Resource",
    "tnid": "0",
    "translate": "0",
    "type": "resource",
    "uid": "1",
    "uuid": "34b45055-bc10-416f-a8ba-8b9f27718362",
    "vid": "118",
    "vuuid": "ac1c1aa3-f6ee-4f76-a2f9-6510d9504680"
}
```


Check the datastore status for a resource

This and the following two examples allow you to control the datastore for a particular resource.

Request

```
GET /api/dataset/datastore/21 HTTP/1.1
Content-Type: application/json
X-CSRF-Token: SbKKNwXWRpBmTPVylBoWQA7SEg6DI0-n_e1Sx8g6GPg
Accept: */*
cookie:␣
↪SSESS22474bf3c2765681586dfeb396a82796=PrtS2ZEX9SBBLfNYNF6pwF94Lt0M9AEuhp4vYQ9J3nA
```

Response

```
---
class: Dkan\Datastore\Manager\SimpleImport\SimpleImport
storage: 0
data_import: 2
configurable_properties:
    delimiter: ,
    quote: "
    escape: \
    trailing_delimiter: false
```

Note: Currently this endpoint returns plain text in YAML format; a later version will return propper JSON or XML.

Import resource file into datastore

Request

```
POST /api/dataset/datastore/21/import HTTP/1.1
Content-Type: application/json
X-CSRF-Token: ftdT-Kpj4jg-UliwMml5-ZaxMFdRHnsFyIPbE9ACeg8
cookie:␣
↪SSESS22474bf3c2765681586dfeb396a82796=PrtS2ZEX9SBBLfNYNF6pwF94Lt0M9AEuhp4vYQ9J3nA
```

Response

```
---
- true
```

Remove datastore table and record

```
DELETE /api/dataset/datastore/21 HTTP/1.1
Content-Type: application/json
X-CSRF-Token: ftdT-Kpj4jg-UliwMml5-ZaxMFdRHnsFyIPbE9ACeg8
cookie:␣
↪SSESS22474bf3c2765681586dfeb396a82796=PrtS2ZEX9SBBLfNYNF6pwF94Lt0M9AEuhp4vYQ9J3nA
```

Response

```
---
- true
```

7.1.7 Testing it out

Command Line (curl)

If you want to quickly test that the functionality is working, you can use curl to send requests a terminal.

Authentication

```
curl -X POST -i -H "Content-type: application/json" -H "Accept: application/json" -c␣
↪cookies.txt -X POST http://docker:32774/api/dataset/user/login -d '{
  "username":"admin",
  "password":"password"
}'
```

This will return the cookie and the **CSRF token** that we need to reuse for all the authenticated user iteration via the API.

Create a new dataset

This will need an authenticated user with appropriate permissions. The headers include the user credentials (cookie and CSRF token).

```
curl -X POST -i -H "Content-type: application/json" -H "X-CSRF-Token:␣
↪8RniaOCwrsK8Mvue0al_C6EMAraTg26jzklDdLLgvns" -b cookies.txt -X POST http://
↪docker:32774/api/dataset/node -d '{
  "title":"A node created via DKAN REST API",
  "type":"dataset",
  "body": {
    "und": [{"value": "This should be the description"}]
  }
}'
```

In a PHP script

Log In and get the Session Cookie

```
// Setup request URL.
$request_url = 'http://docker:32774/api/dataset/user/login';

// Prepare user data.
$user_data = array(
    'username' => 'theusername',
    'password' => 'theuserpassword',
);
$user_data = http_build_query($user_data);
```

```
// Set up request.
$curl = curl_init($request_url);
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Accept: application/json'));

// Accept JSON response.
curl_setopt($curl, CURLOPT_POST, 1); // Do a regular HTTP POST.
curl_setopt($curl, CURLOPT_POSTFIELDS, $user_data); // Set POST data.
curl_setopt($curl, CURLOPT_HEADER, FALSE);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($curl, CURLOPT_FAILONERROR, TRUE);

// Execute request and get response.
$response = curl_exec($curl);

// Process response.
$logged_user = json_decode($response);

// Save cookie session to be used on future requests.
$cookie_session = $logged_user->session_name . '=' . $logged_user->sessid;
```

Get the CSRF Token

```
// Setup request URL.
$request_url = 'http://example.com/services/session/token';

// Setup request.
$curl = curl_init($request_url);
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Accept: application/json')); // Accept␣
↪JSON response.
curl_setopt($curl, CURLOPT_POST, 1); // Do a regular HTTP POST.
curl_setopt($curl, CURLOPT_COOKIE, "$cookie_session"); // Send the cookie session␣
↪that we got after login.
curl_setopt($curl, CURLOPT_HEADER, FALSE);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($curl, CURLOPT_FAILONERROR, TRUE);

// Execute request and save CSRF Token.
$csrf_token = curl_exec($curl);
```

Create a Dataset

```
// Set up request URL.
$request_url = 'http://example.com/api/dataset/node';

// Set up dataset data.
// A great explanation on how to target each node field can be found on the
↪'Identifying field names' article linked on the 'Documentation' section.
$dataset_data = array(
    'type' => 'dataset',
    'title' => 'Example dataset',
    'status' => 1,
    'body[und][0][value]' => 'The description',
    'field_resources[und][0][target_id]' => 'Madison Polling Places (5)', // Resource␣
↪title plus node id
```

```
    'field_author[und][0][value]' => 'Bob Lafollette'
);
$dataset_data = http_build_query($dataset_data);

// Set up request.
$curl = curl_init($request_url);
curl_setopt($curl, CURLOPT_HTTPHEADER, array('Accept: application/json', 'X-CSRF-
↪Token: ' . $csrf_token));
curl_setopt($curl, CURLOPT_POST, 1); // Do a regular HTTP POST.
curl_setopt($curl, CURLOPT_POSTFIELDS, $dataset_data); // Set POST data.
curl_setopt($curl, CURLOPT_COOKIE, "$cookie_session");
curl_setopt($curl, CURLOPT_HEADER, FALSE);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, TRUE);
curl_setopt($curl, CURLOPT_FAILONERROR, TRUE);

// Execute request and get response.
$response = curl_exec($curl);
```

Python client

Be sure to look at the pydkan Python client to see a working API client you can build on for your own applications.

Safe FME Integration

Building on the pydkan client, the FME Workflows repo provides code and instructions for integrating DKAN into Safe FME workflows.

7.1.8 Known issues

- Datasets and other content nodes can only be queried via node id or other entity. UUID support pending.
- There is currently no way to request a previous revision of a dataset or resource.
- Upon attaching a file to a resource via the API, DKAN will immediately import this file to the Datastore if it is a valid CSV. This may not always be the desired behavior; more control over datastore behavior should be available to API clients.
- Greater-than (>) and lesser-than (<) operations are not currently supported in queries. Equals (=) however should always work.

7.2 Datastore API

DKAN offers a Datastore API as a custom endpoint for the Drupal Services module.

To import and otherwise control individual resources in the Datastore, use the datastore functions in the *Dataset REST API*.

This API is designed to be as compatible as possible with the CKAN Datastore API.

7.2.1 Parameters

- **resource_id** (*mixed*) – id (string) or ids (array) of the resource(s) to be searched against.
- **filters** (*mixed*) – array or string of matching conditions to select
- **q** (*string*) – fulltext search
- **offset** (*int*) – offset this number of rows
- **limit** (*int*) – maximum number of rows to return
- **fields** (*array or comma separated string*) – fields to return (default: all fields in original order)
- **sort** (*string*) – comma separated field names with ordering
- **join** (*array*) – array of fields to join from multiple tables
- **group_by** (*array*) – array of fields to group by

Note: If **limit** is not used in a query, 10 records will be returned by default. If **limit** is used, the API allows users to ask for up to 100 records. To get more than 100 records, the API must be used as a user with the **"Perform unlimited index queries"** permission.

7.2.2 Aggregation functions

- **sum** (*string*) – field to compute the sum
- **avg** (*string*) – field to compute the average
- **min** (*string*) – field to compute the minimum
- **max** (*string*) – field to compute the maximum
- **std** (*string*) – field to compute the standard deviation
- **variance** (*string*) – field to compute the variance
- **count** (*string*) – field to compute the count

7.2.3 URL format

Parameters passed by URL share a common format:

```
param_name[resource_alias][field_name]=value,value1
```

- **param_name**: the param you are using (e.g. offset)
- **resource_alias(optional)**: an alias to reference an specific resource in further params.
- **field_name(optional)**: a field name used by the param name.
- **value**: a list of values divided by commas

Note that `resource_alias` and `field_name` arguments are optional and depend on what you want to query. For example, if you need to limit the number of records, you need to use the limit parameter. However, it doesn't make sense to specify an alias or a field in such a case. You only need to provide the number of records you need to retrieve:

```
...&limit=5
```

There is one exception: Even when the `sort` parameter shares the above syntax, it also accepts an alternative format:

```
...&sort[field1]=desc
```

7.2.4 Multiple queries

Sometimes you want to do mutiple datastore queries in one network request (e.g., to feed a data dashboard). In that case you can post a JSON object to http://EXAMPLE.COM/api/action/datastore/search.json with all the queries to perform.

The request body should have a format similar to this:

Request body

```
{
  "my_query": {
    "resource_id": {
      "states": "d2142282-9838-4cca-972f-f1741410417b",
      "gold_prices":"d3c099c6-1340-4ee5-b030-8faf22b4b424"
    },
    "limit": 5
  },
  "my_query1": {
    "resource_id": {
      "gold_prices": "d3c099c6-1340-4ee5-b030-8faf22b4b424"
    },
    "limit": 5
  }
}
```

Response

```
{
  "my_query": {
    "help": "Search a datastore table. :param resource_id: id or alias of the data␣
↪that is going to be selected.",
    "success": true,
    "result": {
      "fields": [
        {
          "id": "nombre",
          "type": "text"
        },
        {
          "id": "state_id",
          "type": "int"
        }
      ],
      "resource_id": {
        "states": "d2142282-9838-4cca-972f-f1741410417b",
        "gold_prices": "d3c099c6-1340-4ee5-b030-8faf22b4b424"
      },
      "limit": 1,
      "total": 5,
```

```
      "records": [
        {
          "nombre": "Alabama",
          "state_id": "1",
          "feeds*flatstore_entry*id": "1",
          "timestamp": "1466096874",
          "feeds*entity*id": "13"
        }
      ]
    }
  },
  "my_query1": {
    "help": "Search a datastore table. :param resource_id: id or alias of the data
↪that is going to be selected.",
    "success": true,
    "result": {
      "fields": [
        {
          "id": "date",
          "type": "datetime"
        },
        {
          "id": "price",
          "type": "float"
        },
        {
          "id": "state_id",
          "type": "int"
        }
      ],
      "resource_id": {
        "gold_prices": "d3c099c6-1340-4ee5-b030-8faf22b4b424"
      },
      "limit": 1,
      "total": 748,
      "records": [
        {
          "date": "1950-01-01",
          "price": "34.73",
          "state_id": "1",
          "feeds*flatstore_entry*id": "1",
          "timestamp": "1466036208",
          "feeds*entity*id": "12"
        }
      ]
    }
  }
}
```


7.2.5 Response formats

Requests can be sent over HTTP. Data can be returned as JSON, XML, or JSONP. To retrieve data in a different format, change the extension in the url.

Instead of using this:

```
http://EXAMPLE.COM/api/action/datastore/search.json
```

Use this:

```
http://EXAMPLE.COM/api/action/datastore/search.xml
```

Or this:

```
http://EXAMPLE.COM/api/action/datastore/search.jsonp
```

7.2.6 Limitations

- The `q` parameter doesn't work in combination with the `join` parameter.
- Filters don't work with float (decimals) values

7.2.7 Examples

The following is a simple example with two resources that contain four records each. Note that the resource `id` would be a UUID not single digit number in real scenario.

Resource 1:

country	population	id	timestamp
US	315,209,000	1	1359062329
CA	35,002,447	2	1359062329
AR	40,117,096	3	1359062329
JP	127,520,000	4	1359062329

Resource 2:

country	squarekm	id	timestamp
US	9,629,091	1	1359062713
CA	9,984,670	2	1359062713
AR	2,780,400	3	1359062713
JP	377,930	4	1359062713

Simple query example

```
http://EXAMPLE.COM/api/dataset/search?resource_id=d3c099c6-1340-4ee5-b030-
↪8faf22b4b424&filters[country]=AR,US&fields[]=country&fields[]=population,timestamp&
↪sort[country]=asc
```

Returns the country, population, and timestamp fields for US and AR from dataset 1 sorting by the country in ascending order.

Text Search

Requests with the 'query' argument will search the listed fields within the dataset:

```
http://example.com/api/dataset/search?resource_id=d3c099c6-1340-4ee5-b030-
↪8faf22b4b424&&fields[]=country&fields[]=population&query=US
```

This will return the country and population from US.

Joins

If you wish to query multiple tables, indicate the table as an array key in the following fields:

```
http://example.com/api/dataset/search?resource_id[pop]=d3c099c6-1340-4ee5-b030-
↪8faf22b4b424&resource_id[size]=d3c099c6-1340-4ee5-b030-8faf22b4b424&
↪filters[pop][country]=US,AR&join[pop]=country&join[size]=country
```

Returns the `country`, `population`, `squarekm` and `id` for "US" and "AR" from datasets 11 and 13.

Caching

GET and POST request are cached by Drupal. The params passed through the request are used to create a cache id to store the data to be retrieved on further requests.

Since Datastore API uses the Drupal cache system under the hood, the Datastore API cache will be cleared at the same time as the rest of the Drupal cache. This could be when the cache is wiped manually, or when the cache lifetime ends.

These options can be configured at `admin/config/development/performance`

Field Names

In order to get data for specific fields, you need to add the argument 'fields' as an array to your request:

```
http://example.com/api/dataset/search?resource_id=d3c099c6-1340-4ee5-b030-
↪8faf22b4b424&fields[]=country
```

If you wish to get multiple fields, add the 'fields[]' parameter followed by the field name as many times as fields needed, for example:

```
http://example.com/api/dataset/search?resource_id=d3c099c6-1340-4ee5-b030-
↪8faf22b4b424&fields[]=field_name_1&fields[]=field_name_2
```

Important Note: if the resource you are querying has a file with column names which contains spaces or capital letters (e.g. 'School Name'), you should NOT specify the field in the request in that way, instead, it should be referenced as lower case with underscores instead of spaces (e.g. school_name).

Read more about changes to the Datastore after upgrading to DKAN 7.x-1.16 here

Filters

If you wish to filter data based on the value of a field, you'll need to specify the 'filters' parameter in the request, it should be formatted like:

```
filters[field_name_1]=value,value2
```

When you need to specify filters for multiple fields, then you'll just join them with &, like this:

```
filters[field_name_1]=value&filters[field_name_2]=value2
```

So, for example you could execute the query:

```
http://example.com/api/dataset/search?resource_id=d3c099c6-1340-4ee5-b030-
↪8faf22b4b424&filters[country]=AR,US
```

That query will return the records in which the 'country' is set to US or AR.

7.3 CKAN Dataset API

DKAN provides a number of public, read-only APIs that are designed to provide catalog and dataset information as well as updates that allow observers to track and pull in changes. These public APIs are specifically designed to allow CKAN sites to harvest from DKAN based off of the APIs used for the CKAN Harvester.

All the APIs listed on this page are provided via the Open Data Schema Map module.

7.3.1 Supported APIs

site_read

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.site_read

Demo: http://demo.getdkan.com/api/3/action/site_read

revision_list

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.revision_list

Demo: http://demo.getdkan.com/api/3/action/revision_list

package_list

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.package_list

Demo: http://demo.getdkan.com/api/3/action/package_list

current_package_list_with_resources

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.package_list

Demo: http://demo.getdkan.com/api/3/action/current_package_list_with_resources

package_show

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.package_show

Demo: http://demo.getdkan.com/api/3/action/package_show?id=5dc1cfcf-8028-476c-a020-f58ec6dd621c

resource_show

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.package_show

group_package_show

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.package_show

package_revision_list

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.package_show

group_list

See: http://docs.ckan.org/en/latest/api/index.html#ckan.logic.action.get.package_show

Demo: http://demo.getdkan.com/api/3/action/group_list?order_by=name&all_fields=TRUE

7.4 Open Data APIs

In addition to the Drupal and CKAN-based APIs supplied with DKAN, two major open data standards are supported. Both are supplied by and configurable through the Open Data Schema Map module.

7.4.1 Project Open Data

DKAN provides full support and mapping for U.S. Project Open Data, in both its federal and non-federal variants, with a data.json endpoint. The optional *Open Data Federal Extras* module is needed for full federal POD compliance. See a demo here.

7.4.2 DCAT-AP

DKAN also provides endpoints and configurable field mappings for DCAT-AP, the application profile for data portals in Europe, developed by the European Commission. DCAT-AP is of course based on the Data Catalog Vocabulary (DCAT), but provides stricter definitions of catalogs, datasets, distributions and other objects. Through Open Data Schema Map, DKAN provides both a catalog endpoint (see demo) and individual RDF endoints for each dataset (see by going to any dataset on http://demo.getdkan.com/ and clicking the "RDF" link on the lefthand sidebar).

7.4.3 Field Comparison

Catalog

Dataset

DKAN Field/Property label	DKAN Field/Property machine name	DCAT-AP property	POD property
Title	title	dct:title	title
Description	body	dct:description	description
Tags	field_tags	dcat:keyword	keyword
License	field_license		license
Author	field_author		
Spatial / Geographical Coverage Area	field_spatial_geographical_area		
Spatial / Geographical Coverage Location	field_spatial_geographical_cover	dct:spatial	spatial
Frequency	field_frequency	dct:accrualPeriodicity	accrualPeriodicity
Publisher	og_group_ref	dct:publisher	publisher
Temporal Coverage	field_temporal_coverage	dct:temporal	temporal
Granularity	field_granularity		
Data Dictionary	field_data_dictionary		dataDictionary
Contact Name	field_contact_name	dcat:contactPoint.vcard:fn	contactPoint
Contact Email	field_contact_email	dcat:contactPoint.vcard:hasEmail	mbox
Public Access Level	field_public_access_level	dct:accessRights	accessLevel
Additional Info	field_additional_info		
Resources	field_resources	dcat:distribution	distribution
Related Content	field_related_content		references
Data Standard	field_conforms_to		conformsTo
	uuid	dct:identifier	identifier
	modified_date	dct:modified	modified
	release_date	dct:issued	issued

The following properties are provided by the Open Data Federal Extras module and have no equivilant in DCAT. They are only relevant to U.S. federal agencies.

DKAN Field/Property label	DKAN Field/Property machine name	POD property
Bureau Code	field_odfe_bureau_code	bureauCode
Data Quality	field_odfe_data_quality	dataQuality
Primary IT Investment UII	field_odfe_investment_uii	primaryITInvestmentUII
Program Code	field_odfe_program_code	programCode
System of Records	field_odfe_system_of_records	systemOfRecords

Resource / Distribution

DKAN Field/Property label	DKAN Field/Property machine name	DCAT-AP property	POD property
		dcat:accessURL	accessURL
		dct:conformsTo	conformsTo
			describedBy
			describedBy-Type
Description	body	dct:description	description
Link File || Upload	field_link_remote_file || field_upload	dcat:downloadURL	downloadURL
Format	field_format		format
	field_upload:mime	dcat:mediaType	mediaType
Title	title	title	dct:title

CHAPTER 8

Releases

8.1 Release Notes

Release notes here will be identical to the releases kept in the Github repository's releases section.

8.1.1 DKAN 1.17.1 Release notes

- #2982 #2986 Remove admin_views
- #2977 Move documentation images to S3
- #2959 Remove deprecated ahoy files
- #2973 Update list of professional help in docs
- #2972 Fix branch/tag in drupal-org.make file
- #2967 Upgrade ODSM and enable fast token browser
- 5282845 Adjustments to behat tests
- 8953e45 Restore phpunit.xml file
- #2964 Fix polling places file
- #2963 Fix Call to a member function getTimestamp() on null error
- fd331bc Update dkan sites list

8.1.2 DKAN 1.17 Release notes

What's new

PHP 7.2 Support

DKAN code and all contrib modules have been updated to be compatible with PHP 7.2.

DKAN Periodic Updates

The DKAN Periodic Updates module provides automated datastore updates on existing resources on a daily, weekly or monthly basis. If you have resources that change often and need the data to be re-imported to the datastore on a regular basis this feature is for you. It is included with DKAN core but will not be enabled by default. Consult the documentation on how to set up and configure.

Miscellaneous improvements

This release includes updates and bug fixes to DKAN Workflow, Open Data Schema Mapper, and accessibility compliance.

The decommissioning of Feeds

The 7.x-1.16 release introduced a refactored DKAN Datastore that no longer relied the feeds module. In this release we are disabling feeds and its companion modules so that in the next release they can be completely removed from the code base.

If you have custom code:

- If you are NOT using feeds, feeds_field_fetcher, or feeds_flatstore_processor in your custom code, you can proceed with the upgrade without additional preparations.
- If you DO use feeds, feeds_field_fetcher, or feeds_flatstore_processor in your custom code:

  This release **disables** and **uninstalls** the feeds module. If you are using feeds on your site outside of the DKAN open data operations, be sure to backup your database. Comment out the update hook *dkan_update_7035*. You will also need to create custom permissions to assign permissions for these modules to your user roles.

Manual Deployment steps

- `drush rr`
- `drush updb -y`
- `drush cc all`
- As part of the update hook *dkan_update_7035* all feeds related modules are disabled so it is possible that the next time you log in to your site you get a message stating that "The content access permissions need to be rebuilt", if that's the case you can proceed safely to execute the rebuild process.

7.x-1.17

- #2956 Update new WCAG-compliant default theme colors to be consistent across the site.
- #2957 Update frequency values on default content files.

- #2954 Update contrib modules, adjust DKAN code to support PHP 7.2
- #2951 Update default greetings on workbench email templates.
- #2908 Add DKAN Periodic Updates module
- #2804 Disable and uninstall feeds, feeds_flatstore_processor, and feeds_field_fetcher modules
- #2945 Upgrade imagecache_actions to 1.11
- #2944 Specify behavior on dkan_workflow moderation set state action to avoid extra node_save.
- #2943 Add documentation for DKAN Feedback
- #2940 Update Open Data Schema Map to 7.x-2.4
- #2935 Add documentation for DKAN custom solr
- #2931 Add patch for NULL controller class
- #2921 Fix gaps in WCAG compliance

8.1.3 DKAN 1.16.13 Release notes

- #2928 Update ODSM to 2.3 to fix broken RDF/JSON links
- #2918 Avoid importing/orphaning when harvest cache is not present for unavailable sources.
- #2919 Security update for imagecache_actions

8.1.4 DKAN 1.16.12 Release notes

- #2892 Harvest support range dates separated by dashes
- #2894 Remove user field on unpublished nodes tab on workbench moderation UI.
- #2889 Make Page Title reflect dataset/resource content
- #2893 Update Open Data Schema Map to 7.x-2.2
- #2885 Security update for uuid and tablefield modules

8.1.5 DKAN 1.16.11 Release notes

- #2876 Update the datastore lock code to use drupal functions
- #2879 Update dkan_workflow to add option to delete unpublished nodes
- #2880 Fix drush command that deletes orphaned datastore tables
- #2870 Remove workflow emails from dkan_workflow
- #2878 Remove hard-coded access to revisions for anonymous users

8.1.6 DKAN 1.16.10 Release notes

- #2873 Patch globalredirect to fix network issue
- #2874 Security update for Drupal to 7.67 (SA-CORE-2019-007)
- #2730 Behat test improvements

- #2862 Add primary key to datastore tables. Create hook_datastore_post_import to allow users to execute actions right after data has been imported into the datastore.
- #2842 Fix harvest migration message logging for batched migrations

8.1.7 DKAN 1.16.9 Release notes

- #2864 Tablefield security update
- #2864 Drupal core security update
- #2863 Use the data.json mediaType value to assign the resource extension during Harvest if the format is not given.
- #2809 Remove 'Feedback' from the list of content types managed by DKAN Workflow
- #2809 Apply patch to views to fix a bug on set_group_operator when using hook_views_query_alter
- #2809 Remove user picture from comment.tpl.php
- #2860 Document new datastore behavior
- #2852 Add 'count' to the list of functions supported by the Datastore API

8.1.8 DKAN 1.16.8 Release notes

- #2854 Wrap words on harvest summary sidebar.
- #2856 Switch to forked version of font_icon_select module.
- #2858 Security Upgrade of Services to 3.24.

8.1.9 DKAN 1.16.7 Release notes

- #2780 Run tests with dktl.
- #2853 Security Upgrade of Module Filter to 2.2.

8.1.10 DKAN 1.16.6 Release notes

- #2848 Upgrade Drupal to 7.65
- #2812 Use ISO 8601 Date Format for Temporal Coverage

8.1.11 DKAN 1.16.5 Release notes

- #2846 Update for views 7.x-3.21 and ctools 7.x-1.15
- #2824 Handle missing harvest source with a warning rather than mass un-publishing of content.
- #2840 Skip the dataset_changelog step when harvesting, datasets always get a new revision with out it
- #2839 Add drush command for deleting orphaned data tables `drush dkan-datastore-drop-orphan-tables`
- #2828 Implement the –feedback flag when harvesting to avoid SSH time outs during long harvests

8.1.12 DKAN 1.16.4 Release notes

- #2837 Security updates for the services and path_breadcrumbs contrib modules
- #2834 Add drush command drush ddpr to reduce duplicate dataset revisions from the same timestamp to a single revision.
- #2830 Add patch to workbench_moderation to avoid deadlock on update
- #2821 Declare file size of remote files as 'unknown' since the value given can be inaccurate
- #2823 Update dkan_dataset_validate_date() function to work on additional date formats

8.1.13 DKAN 1.16.3 Release notes

- #2832 Security updates for link and restws
- #2816 Fix resource titles when harvesting from Socrata - use the format when a title is not available

8.1.14 DKAN 1.16.2 Release notes

- #2805 Update Drupal Core to 7.63
- #2800 Update Drupal Core to 7.62
- #2792 Fix text/tab-separated-values file mime types to work with the datastore
- #2793 Fix documentation on aggregation functions min and max.

8.1.15 DKAN 1.16.1 Release notes

- #2778 Add check for partial date elements when running harvest
- #2787 Update rest-api documentation
- #2788 Update contrib modules: diff, field_hidden, file_entity, honeypot, libraries, menu_block, migrate, panopoly_images, panopoly_widgets, search_api, tablefield, taxonomy_menu
- #2784 Update harvest node machine name field settings to limit length and replace single hyphen
- #2785 Fix harvest support for dataset identifiers that end with a '/'
- #2782 Fix dataset and resource orphan field update on post-harvest import operations
- #2781 Use dkan_allowed_extensions() during harvest
- #2760 Remove content page create/edit permissions from the editor role
- #2775 Add check for mysql reserved words in column names when importing data to the datastore
- #2777 Add check for SelectQuery object when running dkan_datastore_api_debug
- #2776 Update recline revision
- #2767 Update workbench email notification text
- #2774 Update the license options uri values
- #2773 Fix Only variables should be passed by reference errors in dkan_datastore_api
- #2772 Fix CartoDB preview error
- #2763 Update dkan_environment key from 'stage' to 'test'

- #2762 Update Drupal to 7.61

8.1.16 DKAN 1.16 Release notes

What's New

Complete Refactor of DKAN Datastore

The Feeds module has long been a key component of the DKAN Datastore. While it has served us well over the years as a community-supported framework for importing CSV and similar files into database tables, it has also added a lot of overhead and bloat to what should be a simple system. At the same time, much of the datastore code is written specifically to interact with Feeds, and separating the two proved to be impossible. The Datastore module has now been completely re-written to make it faster, more stable, and more modular. A properly object-oriented, decoupled architecture allows its various classes to be extended and plans to support additional options for datastore infrastructure (such as a second MySQL database or even a 3rd party service like Carto) are in the works.

New CSV importer w/o Feeds

A simple parser will try to import the entire file into the datastore, and continue in the background on the next cron run if the file is too big. Improvements can be found both in the code and in the UI. See the docs for more information.

Better API support

Datastore behaviors can also now be controlled through the Dataset REST API, so that you can automate actions like importing resources to the datastore and dropping datastore tables.

Note: While the system for getting data into the datastore has changed significantly, the API for querying the datastore remains the same.

Improvements to DKAN command-line tooling

While not a change to the core DKAN codebase, this release marks the release of a new command-line tool for working with DKAN, DKAN Tools. This will make it easier for anyone to stand up DKAN locally, manage Docker containers, and use different CI pipelines.

Support for non-date values in the dataset "modified" field

The "Modified Date" field has been converted to a text field to accommodate ISO 8601 repeating interval values such as `R/P1D` and `R/P2W`. Previously, this field had been a date field, which was incompatible with certain values that would be allowed in Project Open Data's modified field.

Support PHP 7

Many great performance improvements happened in PHP 7. DKAN has been updated to be compatible with php 7.1 and allows users to take advantage of those improvements. Our CI and testing infrastructure has also been updated to use PHP 7.1.

Additional improvements in this release

- #2179 Refactor Datastore Importer
- #2627 Add support for ISO-8106 duration values in "modified" when harvesting
- #2729 Update recline to 2.1
- #2713 CSV Parser Improvements: trailing commas & escape chars
- #2740 Fix dkan_workflow anonymous access to revisions and unpublished content
- #2724 Update contrib modules:
  - better_exposed_filters
  - file_entity
  - media
  - panopoly_widgets
  - search_api_db
  - uuid
  - date: fixes issue 2843367
- #2718 Update odsm_dkan token values for remote file
- #2707 DKAN workflow documentation updates
- #2723 Include dkan.profile to prevent undefined function in update

8.1.17 DKAN 1.15.5 Release notes

- #2710 Replace data stories page view with a non-search index version
- #2712 Remove unused xss code

8.1.18 DKAN 1.15.4 Release notes

- #2701 Update drupal core to 7.60
- #2541 Updates for php 7.1 compatibility
- #2702 Fix 'a non-numeric value encountered' error
- #2696 Peg drush to a specific version
- #2670 Update ODSM to 2.1 for php 7 compatibility
- #2692 Patch the radix theme to use local bootstrap.min.js file
- #2664 Add UI to change allowed extensions on resources
- #2668 Add environment and environment_indicator modules to DKAN
- #2656 Update workbench_moderation, drafty, and dkan_sitewide_panels to fix 'View draft' button
- #2658 Fix behat test @workflow_19 to avoid false positive result
- #2644 Automate group role assignments
- #2650 & - #2634 Documentation updates

- #2637 Fix version check on available updates for empty value
- #2636 Add created date as a sort option on search pages
- #2628 Update Datastore API documentation

8.1.19 DKAN 1.15.3 Release notes

- #2605 Remove group member count and link from the group block
- #2604 Update select_or_other to 2.24
- #2601 Update file_resup to 1.5
- #2418, #2600 & #2606 Documentation updates
- #2599 Adds additional helper function to get remote file info
- #2592 CSS updated to add ppt/pptx file icons
- #2589 Security update for uuid (1.1)
- #2586 Switch to saving the data.json url by default when harvesting remote files rather than the effective url
- #2585 Add kml, kmz, shp to allowed extensions on resource file upload and remote file fields
- #2588 More granular field validation for datasets and resources.
- #2557 Wrap Dkan defined string to improve multilingual support

8.1.20 DKAN 1.15.2 Release notes

- #2576 Add redirect to search page on 404 status
- #2572 Open Data API table documentation tweaks
- #2561 Test improvements
- #2553 Add an example on how to set the dkan_map_tile_url variable
- #2552 Run linting before tests
- #2567 Fix typo in README.md
- #2544 Update contrib modules
- #2563 Remove changelog checkbox from PR template
- #2562 Fix the docker download url

8.1.21 DKAN 1.15.1 Release notes

- #2533 #2546 Updates docker-compose to use updated cli container
- #2542 Add theme setting to adjust hero region height
- #2531 Add table tags to allowed html
- #2527 Stop xdebug in web and cli containers for tests
- #2521 Add custom validation rules to the REST API node creation
- #2498 Fix the link to installation info on README.md

- #2463 Update circle setup to use local DEV containers
- #2485 Check the character length of harvested resource urls
- #2476 Update getExtension() and createResources() functions
- #2452 Fix front page link paths on logo and site name
- #2489 Create variable for map tile url
- #2492 Security update for file_resup module, upgrade to version 1.5
- #2358 Fix datastore error reporting when UUID is missing

8.1.22 DKAN 1.15 Release notes

What's New

API improvements

A solid API is the window for DKAN to easily integrate with any other useful tool that allows us to get more value from our Open Data. DKAN will not be able to be all thing to all people, but with a solid API we can be the central repository for your data without any fear of any limitations. In this set of improvements, we'll be making some fixes to allow our API to consume the same kinds of content it creates.

DKAN Link Checker

This new feature is included in the distro but not enabled by default. It is based on the Link checker module with some additional code to add views support and file url support. It is intended to help site managers see where links are failing in their data catalog and make it easy to weed out or fix the data.

Once enabled, urls in datasets, resources and harvest sources will be checked when cron runs. If a link is found to be broken it will be added to a report visible to site managers. The report will include the error code, the url that triggered the error, a link to the content where the error occurs, the contact name and email if available, a date when it was last checked, and an option to exclude the url from future checks. The report also allows you to filter on public access (public, restricted, or private), or by content type, or error type.

Additional improvements in this release

- #2434 Disable option to assign content to anonymous user
- #2431 Fix calls to the empty function that check the return value of a function or method
- #2405 & #2427 Improve tests

8.1.23 DKAN 1.14.7 Release notes

- #2473 Security upgrades to Drupal 7.59 and Media 2.9
- #2470 Get harvest tests passing
- #2468 Update urls to use https rather than http
- #2467 Update command center menu links
- #2464 Move functions out of the dkan_sitewide_menu.install file

8.1.24 DKAN 1.14.6 Release notes

- #2461 Update CONTRIBUTING.md
- #2456 Remove exra docker dir
- #2458 Fix the chrome issue
- #2450 Incorrect Readme Links

8.1.25 DKAN 1.14.5 Release notes

- #2435 Update Drupal core to 7.58
- #2412 Update installation instructions

8.1.26 DKAN 1.14.4 Release notes

- #2404 Avoid undefined offset on nuboot_radix/includes/panel.inc line 45
- #2402 Update Drupal core to 7.57
- #2375 Improvements to automated tests

8.1.27 DKAN 1.14.3 Release notes

- #2377 Upgrade entity to 1.9

8.1.28 DKAN 1.14.2 Release notes

- #2359 Upgrade filefield_sources to 1.11
- #2341 Modifies curl settings so url headers and info can be properly acquired with curl
- #2370 Patch features to add new line after <?php
- #2346 Updated Help page; can add your company to listing
- Remove out-dated owners.md file
- #2345 Site Manager Monthly Maintenance checklist
- #2347 Moving the "Adding New Content" page to main admin doc tree
- #2321 Update @resource_all_09 scenario
- #2327 Add results count to dkan admin views
- #2337 Pin sphinx to 1.5.6
- #2332 Remove redundant debug output

8.1.29 DKAN 1.14.1 Release notes

- #2280 Fix contextual filter on list_of_users_groups view
- #2284 Change the default title for harvested resources to use the format value rather than url value
- #2265 Fix hero background image setup
- #2288 Use geojson icon for GeoJSON files in listings
- #2283 Share MySQL Credentials With Docker Containers
- #2273 Remove dkan_workflow_permissions dependency
- #2275 Docker Proxy Config Updates
- #2263 Fix harvest validation for accessLevel values
- #2249 Change timezone handling to 'none' on temporal coverage field
- #2103 Fixed field mapping defaults for open data schema APIs
- #2237 Migration of CircleCI from V1 to V2

8.1.30 DKAN 1.14 Release notes

Read up on the latest release of DKAN, 1.14! While not nearly as big as the previous DKAN point release, 1.13, it brings a number of useful new features and improvements, as well as some important bug fixes. The primary focus of this release is on

1. Improvements and fixes to data visualizations and previews, and
2. Refinement of the Harvester functionality introduced in 1.13.

For a full, detailed list of changes in this release, please consult the CHANGELOG.

What's New

DKAN Harvest

In 1.13, the Harvester would run an entire migration as a single "step" in the web UI, which resulted in timeouts and memory errors on sources with more than a few dozen datasets. When run from the UI, the Harvester now better leverages Drupal's [Batch API] to process the migrations in 5-dataset "chunks." You'll notice more incremental progress on the status bar that appears while a source is being harvested, and hopefully even the largest sources will complete without exhausting server resources.

A Topics field has also been added to *harvest source* nodes. When this field contains one or more of the topics from your data catalog, those topics will be applied to every dataset that is harvested from the source. Additionally, the way the Harvester handles *resource* nodes has been improved and better-documented.

The error messages from the Harvester when filter values are not not present on the source's items have been made clearer.

Visualization Entity Charts

We've made some small but significant improvements to the chart creation experience. Help text is now available for all fields on the chart configuration form, accessed by clicking on a small "?" icon beside the field. Additionally:

- The chart type selection now includes a descriptive name of the chart type (rather than just an image).

- Tick values for axes are validated so that values which would result in an unreadable chart are rejected.
- We now use relative paths for file URLs and set the X-Frame headers to allow easier embedding of charts on external websites.
- Support for values with commas for thousands separators

Dataset Previews

- Tab delimiters and TSV files are now supported for previews.
- Embedded previews no longer show the Grid/Graph/Map tabs.
- Help text has been added to explain the pager buttons on data previews.

Datastore

- Tab delimiters and TSV files are now supported for Datastore import.
- An inconsistency in how the limit (by default, 100 records) on Datastore API requests is applied to results was corrected.
- A bug in the "fast import" method which occasionally led to files being imported to the datastore without dropping the existing records (causing the tables to balloon in size after several consecutive imports of the same file) has been fixed.
- Developer's note: A hook has been added to the Datastore API module to allow altering the fields excluded (see example).

Open Data Schema Mapper

The ODSM module includes a "filecache" system that writes certain open data endpoints to static files on disk rather than generating them on every page request (which is unusably slow on larger sites). A new column on the ODSM main configuration page displays the filecache status of each endpoint and gives the user the ability to generate or delete each cache.

A bug on the mapping for the "language" field in Project Open Data (data.json) that caused many catalogs to fail validation has been fixed.

Workflow

Drafts that were submitted to an editor and then rejected under workflow would never return to the "My Drafts" screen of the original submitter, meaning a contributor could never act on an editor's feedback. This has been fixed.

Administration Views

DKAN now includes the Administration Views module, which replaces Drupal's under-powered default content and user administration pages with more powerful, filterable and searchable versions.

Search

Author facet removed from the search filters. It has been noted that the "author" of a dataset - the Drupal user that the node is connected to - is often not useful metadata at all for an end data consumer. The author is in most cases an arbitrary person on staff who had nothing to do with the creation of the dataset. It is unlikely any data consumer will want to view all datasets created by a particular user.

Upgrade notes

Upgrades to this release should be fairly straightforward – it's a much lower-impact update than version 1.13 was. One note: previous upgrades may have failed on an update to the Search API module. This version contains a patch to avoid this error.

8.1.31 DKAN 1.13.7

This is a "patch" release of DKAN, containing bug fixes and minor updates, but adding no new functionality.

Improvements in this release

- Important security updates to 'views' and 'entityreference' modules.
- Multiple testing improvements.

8.1.32 DKAN 1.13.6

This is a "patch" release of DKAN, containing bug fixes and minor updates, but adding no new functionality.

Improvements in this release

- Add file_resup to the list of up to date projects (#2036)
- Change the 'Datasets' field on the Resource form to use autocomplete (#2035)
- Include language selectors in the 'chosen_jquery_selector' config (#2034)
- Module upgrades: fieldable_panels_panes, media, and search_api (#2020)

Testing Improvements

- Additional test/devops improvements needed for deployment. See (#2012, #2014, #2015, #2016, #2018) for specifics.
- Add configurable arguments via Behat Contexts. (#2025)
- Add unique tags to more test scenarios (#2021)

8.1.33 DKAN 1.13.5

This is a "patch" release of DKAN, containing bug fixes and minor updates, but adding no new functionality.

Improvements in this release

This is a small release adressing some minor issues in 1.13.4. The only significant change is the addition of the devel module to DKAN core. The devel module is used by almost all Drupal developers to debug, test expiremental code, generate dummy content, and perform other developer tasks. DKAN developers are constantly adding the devel module after building DKAN, and a new set of Behat scenarios added to the test suite need to make use of the dummy content generation functions to pass.

The devel module and its submodules *will* be disabled by default in DKAN however, and should only be enabled in development environments. Enabling devel in production should be considered a security risk.

All other improvements are minor issues related to installation and tests that arose during deployments of DKAN 1.13.4. Check the CHANGELOG and Pull Request #2003 where most of these changes are captured.

8.1.34 DKAN 1.13.4

This is a "patch" release of DKAN, containing bug fixes and minor updates, but adding no new functionality.

Improvements in this release

The 1.13.4 release of DKAN contains a very large number of minor fixes. For a full list check the CHANGELOG. Noteable improvements include:

- Some logic concerning when previews were rendered for resources has been simplified. DKAN will now attempt to render a preview for any resource that contains an *uploaded* or *remote* file. No *API or Website URL* links in resources will be previewed other than with an iframe to the URL. Note that in some cases, resources that previously did have Recline previews may now simply display an iframe to the remote resource.
- Relatedly, various issues with previews for resources hosted on external domains have been addressed.
- Publishing options ("draft," "published" etc) were not available when the Workflow module was enabled. This has been fixed.
- A bug where data previews broke when a resource contained *only* numeric data has been fixed.
- A number of Workflow permissions issues have been resolved, allowing users to access "stale drafts" and other data that had been hidden.
- The groups page now lists groups alphabetically rather than by create date.
- Fixed the output for "language" in Project Open Data and DCAT to be standards-compliant.
- Fixed issues with how timestamps are applied to harvested datasets. See below for details.
- An existing hook_update function that upgraded from a previous version of the Open Data Federal Extras module has been fixed to not fail when the fields are not present.
- The `drush dsu` command, to update a datastore from a local file, was broken and has been restored.
- Windows line endings are now supported on Datastore "fast imports"
- Several issues causing the Harvester to break on large harvests have been fixed. Remote files on servers (like Socrata) that will not supply a HEAD request are briefly stored on disk instead of in memory to obtain metadata such as filesize. Also, resource previews are no longer rendered for purposes of indexing for Search API, which was causing indexing at the end of a large harvest to fail.
- Drupal core was updated to 7.56 and many contrib modules were updated to their latest stable version.

Special Notes

- Previous versions of DKAN provided a field "Modified Source Date" (machine name `field_modified_source_date`) to store Project Open Data's "modified" property. The way we capture sources' `modified` and `issued` properties has since been revamped (see #1802), and an update function was needed to remove the old field. However, previous versions had put hook_update functions in the wrong order and some sites did not recieve the change when updating. This release should clear up those remaining issues on sites that still have the `field_modified_source_date` field.

8.1.35 DKAN 1.13.3

This is a "patch" release of DKAN, containing bug fixes and minor updates, but adding no new functionality.

Improvements in this release

- Groups permissions were given better logic. Editors will now become administrators of any group they are added to, giving them permission to edit or moderate any content in that group. Also, Site Manager was given permissions to edit the order of featured groups.
- We now provide parsing options on the Datastore Fast Import. This means a user can set what delimiters and line terminators are used when importing a CSV file, avoiding some import errors users were experiencing.
- Some errors were fixed that caused installing DKAN with the browser (using /install.php rather than drush). Also, special characters were removed from default content filenames, fixing an installation bug on Windows systems.
- The Rules module was patched to prevent a "cache rebuild lock" (see https://www.drupal.org/node/2851567)
- Several bugs in the Harvest module were fixed:
  - The Site Manager can now use the main dashboard view to initiate harvest actions in bulk
  - Inconsistencies in date fields (Harvest source "modified" vs Drupal's "modified" dates) were addressed by adding new fields to the *dataset* content type.
  - A bug preventing the "temporal coverage" field from being harvested was fixed.
- The restws and media modules were updated to latest versions
- Several other smaller bug fixes and improvements; see the CHANGELOG for more information.

Special Notes

Changes to Harvester's date handling

This update changes the way the Harvest module reflects the issued and modified dates of harvested datasets. Project Open Data and most other metadata standards provide an "issued" or "created" date for datasets, as well as a "modified" date. The original release of Harvester simply overwrote Drupal's `created` and `modified` node properties with the source's dates for these fields, but we've run into two problems with this:

1. A node's modified date is easily overwritten by other actions in Drupal
2. It can be useful to store both the date that a dataset was issued in its source, and the date it was added to the portal harvesting it.

An existing field, `field_modified_source_date`, was already handeling some of this but we decided to scrap that and start from scratch. Starting with this release, a source's `issued` and `modified` dates will be stored in the new fields `field_harvest_source_issued` and `field_harvest_source_modified`. When these two fields are present, those will be the dates shown on a dataset's landing page on DKAN and in the site's data.json and DCAT RDF feeds.

All sources should be re-harvested after updating to this patch release to ensure that all date fields and properties are accurate.

8.1.36 DKAN 1.13.2 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates, but adding no new functionality.

See full 1.13 release notes here.

Improvements in this release

- Updated documentation.
- DKAN Datastore: fixed null values being imported as 0.
- Fixed errors related to the open_data_schema_apis_features_rebuild() function.
- Fixed errors when users with content creator role are editing or adding resources.
- Fixed access to the POD validation screen for site managers.
- Fixed access to the featured groups block for anonymous users.
- DKAN Harvest: fixed permissions for the site manager role, they now have access to the cache, delete, harvest, and migrate bulk operations from the Harvest Dashboard.
- DKAN Harvest: added support for importing contact name and contact email.
- Updated contrib modules: services, visualization_entity and views.

8.1.37 DKAN 1.13.1 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 1.13, but adding no new functionality. It was released very shortly after 1.13 to address bugs that surfaced during deployments and upgrades. If you have not yet upgraded to 1.13, upgrade *directly* to this release and skip 1.13; if you have already upgraded, we recommend updating to 1.13.1 immediately.

See full 1.13 release notes here.

Improvements in this release

- Fix validation page permission check using wrong permission name odsm.
- Fixed a bug in the home page conversion function
- Fixed the page title (`<head><title>`) so that it's just the site name (not the node or panel title)
- Updated the DKAN API link on dataset pages to use the new documentation site page.
- Fixed error messages appearing on homepage after upgrade
- Fixed panelizer permissions to hide the "Customize Display" button for Site Managers.

- Additional minor bug fixes to code and tests

8.1.38 DKAN 1.13 Release notes

Read up on our latest release, 1.13! This is a *huge* release and we have new features and enhancements that will make using DKAN better than ever. Major new features include a Harvester, API improvements, new data previews, and DCAT-AP support.

Have questions or thoughts? Let us know on our public DKAN Repo in the issues queue or chat with us in our Gitter room.

What's new

DKAN Harvest

DKAN's new Harvest module gives DKAN administrators and site managers the ability to harvest datasets from other sources for their DKAN site. Built on the widely-used Drupal Migrate system, the Harvest module can fetch data and publish it to a DKAN site.

DKAN Harvest is fully configurable via the DKAN web interface. Harvest sources can be added like any other type of content, and data can be overridden or manipulated before being published. Migrations can be managed via an intuitive dashboard interface or triggered with a cron job.

Get more details on DKAN Harvest by reading its documentation pages.

DKAN Dataset REST API

Previous DKAN APIs have been read-only and focused on CKAN compatibility. A new RESTful API for DKAN datasets provides an endpoint that allows other applications to interact with DKAN, including creating and modifying datasets and resources.

Get more details on the Dataset REST API.

DCAT-AP Compliance

DKAN's DCAT RDF publishing has been re-engineered to be more consistent with its other open-data standards implementation, and, more importantly, to implement the new DCAT-AP protocol. DKAN's fields can be easily mapped to DCAT-AP fields using the Open Data Schema Mapper. Now, DKAN meets US standards (Project Open Data) as well as European Union standards (DCAT-AP).

New Default Content

On a basic install of DKAN, the site came with only a few example Datasets and Resources. We added more default content in order to demonstrate the full capabilities on DKAN and give new users a better experience in testing out various DKAN features. The default content module is part of core DKAN, so all new installs of DKAN contain the content. Simply disabling the "DKAN Default Content" module will remove all the default content to give you a blank slate to start building.

What we've improved

The functionality isn't new, but it's working even better than before. These improvements make DKAN better on the whole including user experience, efficiency, and scale.

Site Managers can assign roles

The Site Manager role is part of a core group of user roles on DKAN. Site Managers have a high degree of control over the entire site, but are expected to be less technical than a full admin. Previously, it wasn't possible for Site Managers to assign roles to new users. By adding the RoleAssign module we have provided this more fine-grained permission.

Site Managers can add pages

One of the most basic content types on DKAN is a *page*. And though the content type is straightforward it can have implications for the structure and appearance of the site. Originally, Site Managers were not able to add or manage Pages; in DKAN 1.13 Site Managers now have this permission.

Even more permissions for Site Managers

- Theme Settings: Includes page elements and ability to add custom logo and favicon from "Site Configuration > Theme Settings" at /admin/appearance/settings
- Colorizer: Ability to create custom color schemes for a DKAN site from "Site Configuration > Colorizer" at /admin/appearance/colorizer
- Open Data Schema Mapper (ODSM): Ability to add, edit, and delete APIs and their mappings to DKAN fields from "Site Configuration > Open Data Schema Mapper" at /admin/config/services/odsm
- Menus: Permission to manage the main menu links by adding, editing, and deleting links from "Site Configuration > Menu" at /admin/structure/menu
- Topics Icons: When adding or editing terms in the Topics taxonomy at path /admin/structure/taxonomy/dkan_topics, site managers can choose an "Icon Type" of either "Font Icon" or "Image." If they choose "Font Icon," the edit form displays a list of available icons from which to choose. If they choose "Image," the form lets them upload an image.
- Enabling External Previews: Site Managers can enable previews so that site visitors can look at Resource contents with visualization tools, Carto and ArcGIS. To enable External Previews, go "DKAN > Data Previews" at /admin/dkan/dataset_preview
- DCAT and POD validation: Site Managers can select the settings for how Datasets are validated against Project Open Data and DCAT-AP standards.
- Site managers can confirm that the website's data.json feed is working correctly by using the POD online validator at https://labs.data.gov/dashboard/validate
- Site managers can confirm that the website's DCAT feed is working correctly from Site Configuration > Open Data Schema Mapper > DCAT Validation at /admin/config/services/odsm/validate/dcat or by using the DCAT online validator at http://www.dcat.be/validator/
- Order of Groups if using the featured groups block: Site Managers can arrange the order in which Groups appear in the featured group block from "DKAN > Featured Groups Sort Order" at /admin/dkan/featured-groups-sort-order
- Recline size configuration: Manage size constraints for Recline, which powers internal previews, from "DKAN > Recline Configuration" at /admin/dkan/recline

Open Data Federal Extras (ODFE) is now part of core DKAN

Project Open Data (POD) sets a standard for the information about datasets (metadata) included when the data is published. DKAN collects metadata with fields on the Dataset form when the data is published. For most agencies, the fields reflect basic requirements. However *US federal* agencies are required to provide additional information about the data published. The extra fields appear as part of the Dataset form with our ODFE module.

We've moved this module into core DKAN so that it's part of every install, though not enabled by default. When enabled, the ODFE module makes federal agencies compliant with POD standards.

Get more details on Federal Extras on DKAN.

In-place Editor for Site Managers + look and feel

Previously, the In-place Editor (IPE, an interface for changing page layouts) for Site Managers included several options that weren't appropriate for their level of permissions. This UI was confusing, so we improved the interface to only include options that are relevant to non-admin user roles. We also improved the overall look and feel of the IPE to be more user-friendly.

Improved Data Previews

The Data Previews feature is designed to let site visitors take a sneak peek at the content of a Resource before downloading the file. In this release, we've improved Data Previews to support more file formats beyond CSV and XLS. Now previews also support JSON, geojson, XML, ArcGIS REST, WMS, images, PDF, and ZIP data formats.

Improvements to Data Previews also offer better support for Resources that are hosted remotely. Previously, Resources that were linked to a web source could only be previewed if they were first imported into the DKAN Datastore (which only supported CSV files). In DKAN 1.13, linked Resources can be previewed with the Data Previews feature.

Improved Datastore API

The DKAN Datastore API makes it possible to query for contents of Resources uploaded to the Datastore as detailed as a cell within a table. The improved Datastore API is enhanced to open greater possibilities of requesting complex queries from contents within a Datastore on a DKAN site as well as multiple queries in a single request.

Get more details on the Datastore API

New Dataset fields added

A number of metadata fields that are common requirements for open data standards like Project Open Data and DCAT-AP have been added to the basic dataset content type in DKAN. These fields are:

- Data Standard
- Data Dictionary Type
- Homepage URL
- Data Standard (conformsTo)
- Rights
- Language

The following fields are also included but hidden. This allows the harvester to ingest metadata with these fields and present them in data.json but they are not yet supported in the UI for locally-created datasets:

- Collection (isPartOf)
- Theme

See the Project Open Data schema for more information on these fields.

NuBoot Radix now in core DKAN

Previously, the NuBoot Radix theme on DKAN was not part of the core repository. The old Omega-based theme and its tools (the Delta module) have been removed, and NuBoot Radix is no longer maintained in a separate repository. Legacy themes will not be maintained or shipped with DKAN and require manual installation if you want to use the Omega theme. Alternatively, you can make customizations to the default theme.

Additionally, the admin theme setting is set to use the default theme rather than the NuBoot Radix theme. Because the Colorizer tool requires both the admin and default to be the same, setting the admin theme setting to use the default theme enables Colorizer to work normally for both the NuBoot subtheme and other custom subthemes without admins needing to manually override this setting.

Added POD-based validation on Datasets

A new option was added on DKAN Dataset Forms at /admin/dkan/dataset_forms in order to enable form validation based on Project Open Data. If "Validate dataset form according to Project Open Data" is enabled, then all POD required fields will be also required in DKAN. See the POD schema for details of the validation standards: https://project-open-data.cio.gov/v1.1/schema/

Minor improvements

- We added more frequency update options to the Dataset form to comply with Project Open Data standards
- Added icons to the Topics drop-down menu.
- Renamed the default HTML text format from "HTML" to "Markdown HTML." This is also reflected in the UI when adding new content.
- Made URLs in the *Additional info* section of Datasets display properly and made them clickable
- Removed the Conditional Fields module from DKAN, which was adding bloat and creating lots of unwanted artifacts in features, and only used by two fields. A few lines of Form API code was enough to reproduce the functionality.
- Several improvements were made to enhance accessibility and 508 compliance on DKAN including labels, alt text in the UI, table headers and more.
- The group field has been simplified in the *dataset* node form and removed from the *resource* node form (*resources* always inherit their *datasets*' groups.)

What we fixed

Some DKAN features weren't working as expected and causing issues. We fixed those, so you can keep using DKAN as expected.

- Data proxy for Google spreadsheets was broken. This meant that the data in a Resource could not be parsed properly, and these files could not be used to generate Charts with the DKAN Visualization Entity. We fixed this so Resources that are not internal to DKAN can be used to create a Chart on DKAN.
- The DKAN API endpoints for JSON and RDF outputs did not generate the proper file when called. We fixed this so that the JSON endpoint works normally. With our new DCAT-AP compliance feature, the RDF output will generate an XML file.
- Several bugs in the Visualization entity resulted in a frustrating user experience when attempting to create a Chart. A number of fixes were made to result in the expected behavior of the Chart.
- Only certain roles had permission to use the "Markdown HTML" text format, which resulted in many users unable to edit text to content which had been created using that format. We fixed that so now all roles can add text to a text box regardless of the text format.
- The toolbar that appears at the top of the text area when using the "Markdown" text format used to include some buttons for styling options that DKAN did not actually support, which created confusion. We've removed the unsupported buttons from the toolbar so it only contains supported options (headings, bold, italic, lists, blockquote, link, image, line break).

Upgrading to the latest version

Removing EVA module

WARNING: The EVA is removed with this release. If you are using EVA in your project you need to add it to your `sites/all/modules` folder.

One view previously used EVA, `user_profile_search`. You can avoid an error by reverting the feature `dkan_sitewide_user`. Note that we recommend reverting all features `drush fra -y` for the upgrade.

Features Reverts

The following commands should be run for the upgrade:

```
drush rr
drush fr dkan_dataset_content_types -y
drush fr dkan_permissions -y
If ODFE is enabled then: ahoy drush fr open_data_federal_extras -y
If ODFE is enabled then: rm -rf sites/all/modules/open_data_federal_extras
drush fr open_data_schema_map_dkan -y
drush updatedb -y
drush en dkan_ipe -y
drush en dkan_harvest_dashboard -y
drush en menu_admin_per_menu -y
drush fra -y
drush rr
```

New Documentation location

We have moved DKAN's technical documentation from the Drupal-based site where it used to live to the /docs folder within the main DKAN repo. Documentation changes will now be part of code commits and pull requests. The documentation is built on ReadTheDocs, but lives at the same URL as the old site: https://docs.getdkan.com/

Special Notes

Front Page - Upgrade Instructions

Front page configuration has been removed from features and the dkan_sitewide_demo_front feature has been deprecated. To save existing front page configuration, run the following command `drush php-eval "dkan_sitewide_convert_panel_page('<page-name>', TRUE);"` This will convert the front page to a panelized node.

Also note that if you use any of the default DKAN blocks for your front page, some may dissapear after the upgrade. A user with the administrator role should be able to restore them by editing the panel and finding them in the Miscellaneous list.

Added Modules

- DKAN Harvest
- DKAN Harvest Dashboard
- DKAN IPE
- Menu Admin per Menu

Removed Modules - The following modules were removed from the code base and upgrading will remove the files and db tables associated with them.

- Conditional Fields
- Entity RDF
- RDF UI
- RDF Extensions
- Delta
- EVA

Removed Themes

- Omega

Consolidation - External repos have been merged into DKAN core. This means that dkan_dataset, dkan_workflow, dkan_datastore and nuboot_radix are all included in the main dkan profile repo.

To remove newly untracked files run:

```
git clean -f
```

Known issues

- Site managers do not have permission to see the Project Open Data validator screen, which tells them if their site is currently producing a valid data.json. This will be fixed in 1.13.1. The equivalant screen *is* availabe for DCAT-AP, under the configuration menu. If you need to check your site's data.json feed in the meantime, you can use Data.gov's online POD validator at https://labs.data.gov/dashboard/validate.

8.1.39 DKAN 1.12.14 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Added css for chart visualizaitons, fixes issue in IE.
- Updates drupal core to 7.53
- Upgrade visualization_entity to 7.x-1.1
- Upgrade services module (via datastore module) to 3.19

8.1.40 DKAN 1.12.13 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Fixed broken recline (resource CSV) preview embeds caused by how recline module loads bootstrap.
- Added patch to remote_stream_wrapper to avoid memory exhausted errors on big files.
- Fixed publisher token in open_data_schema_map_dkan. It was showing only URL rather than publisher name.
- Remote linked files are now proxied through a DKAN site so that we don't run into CORS issues for the visualizations. That can cause memory issues with large files. This only proxies files smaller than 50MB.

8.1.41 DKAN 1.12.12 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Group assignment cleanup update

This release contains a database update that may take some time to complete on larger sites. As described in #1267, on older DKAN sites it was very easy to add resources to datasets that would not be added to the parent dataset's group.

Many sites may have this issue and not even realize, but it will create problems for organizations that use DKAN's out-of-box group roles and permissions; it give users permission to edit content in their own group but not content outside of it. If a resource is not assigned to a group, this could block a group user from editing their groups' own datasets.

This patch release contains a database update that will identify datasets with resources missing their group assignments, and correct them. See pull request #1491 for the fix itself.

Other improvements in this release

- Data previews can be captured in a certain state for embedding into other web pages as an iframe, but these previews were not cached by Drupal. This has been fixed; embedded previews are now cached, which is a significant performance boost for some websites.
- Upgraded Drupal core to 7.52, ctools to 1.11, and Media to 2.0-beta13.
- Fixed links to group pages from group node teasers that broke when site has clean urls disabled
- Fixed some issues with certain filters/facets on the search page being unexpectedly collapsed.

- Fixed a bug introduced on a previous patch release that hid the body field on page nodes.
- Improved the access check and security on datastore pages - unauthorized users could perform certain datastore functions by guessing URLs.
- Fixed "page not found" errors when clicking certain topics links with spaces or special characters in them.

8.1.42 DKAN 1.12.11 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Fixed a bug in Recline regarding file objects and node forms that caused errors when using the "view changes" button on Dataset or Resource edit form
- Center group images in "Groups" page and group node page sidebar
- Add "2x," "3x" etc to Dataset teasers when more than one resource of a particular format present.
- Update the default jquery library setting from 1.7 to 1.10
- Fixed topics menu and facet links if special characters are used in topic terms.
- Fixed dataset form redirect when validation fails, was sending user to node/add/dataset rather than node/%/edit
- Patch fontyourface to remove from the "standard text" selector, to prevent unpredictable results from this option
- Fixed issue in open_data_schema_map with "Data Dictionary" field not displaying URL in data.json file

8.1.43 DKAN 1.12.10 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Fixes an issue causing JavaScript errors in IE browsers, preventing recline previews from displaying. Bug was introduced in 7.x-1.12.7.
- Provides a better upgrade path for Markdown text format and bueditor improvements added in #1085

8.1.44 DKAN 1.12.9 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Fix to a bug introduced in the Recline module in 1.12.7. Previews were not displaying if the resource file was smaller than max preview size, even if loading the resource from datastore.
- Fix a problem with paths for image icons for topics. Uploaded image icons will now display correctly in the Topics menu.

- Fixed the "data dictionary" field to use HTML input format. This field, at least in U.S. Project Open Data, is really intended to have only URL values, and using HTML ensures that links will be clickable. Also made improvements to the field's display and help text.
- Updated the DKAN Workflow content type legend to include all content types with correct icons. This is mostly to prepare for future content types' inclusion in Workflow but also to support sites that already have additional content types in Workflow.

8.1.45 DKAN 1.12.8 Release Notes

Special note: This release contains *security updates* and should be applied immediately.

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Updates the Panels and Panelizer modules following critical security releases. See Drupal security advisories DRUPAL-SA-CONTRIB-2016-047 and DRUPAL-SA-CONTRIB-2014-048.
- Added validation to groups node form to prevent user from creating duplicate groups. Saving a group will now fail if there is an existing group with the same name.

8.1.46 DKAN 1.12.7 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Updates contrib modules (uuid, services, manualcrop, markdown, panelizer, panopoly_widgets and panopoly_images).
- Fixes a bug that produced a "Cannot use a scalar value" error when users trying to add a content pane to a data story node in panels.
- Changes theming on resources to remove iframe to "API or link URL" if remote file field also populated. The only real use-case this applies to is when harvesting datasets from a data.json that includes both an accessURL *and* a downloadURL for a single distribution; both these fields should not be used at once on resouces created using the node form.
- Updates the options in the Dataset "frequency" field to reflect ISO 8601 standards for DCAT and Project Open Data
- Provides a simple file proxy to serve remote CSVs through local domain and resolve cross-origin issues with previews n these resources
- Somes changes to the Resource node form to improve the UX of links and file attachments, especially by replacing the label "Link to an API" to "API or Website URL."
- Fix bug on relative paths for links in theme template files, which caused some links to break when DKAN lives at a subdirectory rather than a website's domain root.
- Fixes a minor typo in the DKAN topics menu link that was exposed by the XSS fix in the 7.x-1.12.6 release.

8.1.47 DKAN 1.12.6 Release Notes

Special note: This release contains *security updates* and should be applied immediately.

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Adds validation to search parameters. Without this, text can be passed into search parameters and passed unsanitized to the browser. The way the facets are themed allowed Javascript to be injected into the attributes of any facet items with icons, (such as content types). The fix validates all facet input and returns a "page not found" error if a search parameter passed in the URL arguments does not match available options. See #1271
- Starts sanitizing output to the facets on the search page, as an additional safeguard against malicious input from the search parameters.
- Adds icons to the topics drop-down menu rather than just the title of the topic.
- Added a new module called Role Assign, which gives site managers the ability to assign roles to other users without giving them access to the entire Permissions module, which meant previously only admins could assign roles.
- Made links and emails in dataset metadata clickable in certain places that they hadn't been for a better user experience.

8.1.48 DKAN 1.12.5 Release Notes

Special note: This release contains *security updates* and should be applied immediately.

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Updates the RESTFull Web Services module to 7.x-2.6, fixing a highly critical security issue announced by the Drupal Security team on July 13.
- Updates the version of recline.js used in the Recline module to point to a new source for map tiles. Previously map tiles were served by MapQuest, but a change in policy has broken map previews on all applications using recline.js. See the original PR on recline.js: okfn/recline#501
- **Updates Drupal core to version 7.50.**

Special Note

This is a fairly big update to Drupal and you may want to familiarize yourself with it by reading the release announcement.

Of note to DKAN users is that "Administer fields" is now a separate permission. This means, for instance that site managers and editors, who by default have "administer taxonomy vocabularies" permissions, will no longer be able to add, remove, reorder or otherwise modify fields in a taxonomy vocabulary.

Only administrators will be able to do administer fields. We see this as a welcome improvement, as making minor changes to, say, the topics vocabulary should not require giving a site manager permission to alter the entire data architecture.

8.1.49 DKAN 1.12.4 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Fixes an oversight that broke RDF endpoints for Datasets (the "RDF" link in the left sidebar of a dataset page). The DKAN Permissions module did not give the "Access the resource node" *RESTFull Web Services* permission to anonymous users.

Upgrade steps

Only necessary if you are using DKAN Permissions module:

```
$ drush features-revert dkan_permissions
```

8.1.50 DKAN 1.12.3 Release Notes

Special note: This release contains *security updates* and should be applied immediately.

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Open Data Schema Map update: Improves performance of package_show endpoint, which became nearly unresponsive on sites with thousands of datasets.
- Point to new release of Visualization Entity, which cleans up some makefiles and brings visualization_entity_charts into the same project repository.
- Fix editor permissions to allow access to visualizations list from admin menu
- Upgrade Views to 7.14
- Upgrade Drupal core to 7.44 (security update)
- Add `<br>, <h2>. <h3>, <center>, <iframe>` to allowed html tags in "Martkdown HTML" format
- Renamed the default HTML text format to "Markdown HTML"
- Improved markdown text editor toolbar
- Several improvements aimed at removing unstable configuration like menus from features:
  - Stories main menu link removed from stories view
  - Groups main menu link removed from page manager groups page config
  - Main menu links added on install function rather than through features
- Add body field to Dashboard content type
- Front page page manager (panels) config moved to the dkan_sitewide_demo_front feature
- Front page group views moved to the dkan_dataset_groups feature
- Add functions to convert between iso and dkan fields #241

Upgrade steps

```
$ drush fr dkan_dataset_content_types -y
```

8.1.51 DKAN 1.12.2 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- Fixed a bug in chart creation with Visualization Entity Charts due to a reference to an incorrect version of csv.js in the Recline makefule

8.1.52 DKAN 1.12.1 Release Notes

This is a "patch" release of DKAN, containing bug fixes and minor updates to DKAN 7.x-1.12, but adding no new functionality. Upgrading should be straightforward.

Improvements in this release

- New version of Visualization Entity Charts fixes a javascript error that was causing multiple UI bugs, and restores support for Google Sheets and Data Proxy data backends
- Fix XSS vulnerability by adding sanitization for titles on workbench view
- Update of a number of contrib modules: Colorizer, Admin Menu Source, Honey Pot, Panopoly Widgets, Panopoly Images, Pathauto, Rules, Restws, Manual Crop, Image Cache Actions, Features, Search API, Organic Groups, Chosen, Date, Entity, Facet API, Facet API Bonus, Facet API Pretty Paths, UUID, Views, Select or other, Remote Stream Wrapper, Link, Libraries, Beauty Tips, Gravatar, OG Extras, Services and Workbench Email.
- Patch colorfield module to avoid incorrect status error. See https://www.drupal.org/node/2696505

8.1.53 DKAN 1.12 Release Notes

Check out what's new in the latest version of **DKAN 7.x-1.12**! Have questions or thoughts? Let us know on our public DKAN Repo with issues or chat with us in our Gitter room.

New Features and Enhancements

New DKAN Workflow Module

DKAN Workflow is a new module that ships with DKAN but is not enabled by default. Once the module is enabled, DKAN Workflow creates an editorial publishing workflow on DKAN for moderating content. Within this module is a secondary set of permissions and roles layered on core DKAN roles and permissions. This module can be very useful to establish more controls among large teams managing content.

Read more about Workflow in the main DKAN Documentation.

Large File Support and Fast Import

Through the file_resup module we've added the ability to support large file uploads on DKAN. There are no limits technically imposed by the software, but file sizes limited up to few gigabytes will produce the best results. For large files, we also added a mySQL statement to quickly import the files into the DKAN datastore. We're calling this DKAN Data Datastore Fast Import module and it appears as such in the module configuration on DKAN. The module is disabled by default, and must be enabled and then configured to be utilized in DKAN. OOB DKAN defaults to using the Drupal file importer that also processes the data (better for smaller files). In configuration, the mySQL statement may replace the Drupal file importer or appear as an additional option for importing a file into the DKAN Datastore. Rather than processing the data, the mySQL statement favors a faster load time by importing directly to a mySQL database.

Large file support also makes for a more robust DKAN Datastore API. Files imported into the DKAN Datastore are included in the public DKAN Datastore API.

Topics

DKAN Topics is a new module added into DKAN core with the 7.x-1.12 release. Topics are preset by an administrator that reflect the areas of interest to the target audience. Topics are configured by default to appear as a panel on the homepage of a DKAN site.

In DKAN, Topics are a taxonomy with a vocabulary (also set by the admin) and can be associated to datasets by lower-access users as they contribute data to DKAN. Additionally admins can set which icons may or may not be used within a font set for Topics by other users. OOB DKAN includes a font set of over 100 icons to choose from, and new font sets can easily be added. Alternatively, images may be uploaded for Topics icons.

Minor Upgrades and Improvements

- Changed UI of DKAN Charts to include "Tick Values" "Step" and "Goal" line in chart. This gives users added functionality in what they can do with Charts. The UI is also more organized to tell users how the options affect which parts of the Charts.
- Upgrade Drupal core to 7.43 to fix security vulnerabilities.
- Security update for Features
- Security update for Fieldable Panels Panes
- Move permissions dependency of from features_roles_permissions to dkan_permissions module.
- Remove export of administrator permissions when new modules are enabled, so that dkan_permissions does not appear as overridden. With the admin_role module, admin roles automatically receive permissions for modules as they are enabled. By removing the export of administrator permissions, overrides will only appear when true changes to the dkan_permissions module have been made.

User experience improvements

- Enable Pretty Paths Facet API to make URL paths on DKAN pages human-readable and SEO optimized. This module is combined with the Drupal Search API module.
- With the new search page redesign in DKAN 7.x-1.11, this release includes optimized search functionality. Previously only datasets were displayed in search results; in version 1.12 all content appears in search results. The /search path will lead to the main search page and the /dataset page will redirect to the /search?type=dataset for backward compatibility, pre-filtered for dataset content search results.

- The dkan_sitewide_profile_page module was removed and new code added to existing modules (detailed documentation as code comments). With the removal of this module, the DKAN command center no longer exists and profiles of any user role appear the same.
- A "Data dashboard" creation link was added on user admin menu.
- Facet titles between user, search and group pages were standarized.

Patches and Bug Fixes

- Added permission to allow anonymous users to see any piece of content created with the visualization entity module such charts and maps.
- Created patch to solve bug that kept fonts enabled after being turned off.
- Fixed regex used to rewrite font file paths on Dkan Topics, which was breaking filepaths in data.json.

Upgrade Notes

If you upgrade an existing site to 7.x-1.12 and enable DKAN Topics, you may encounter issues with your search index. For troubleshooting tips, see the DKAN Topics Readme.

Upgrade instructions

1. `drush updatedb -y`
2. `drush en dkan_featured_topics -y`
3. Enable 'field_topics' on dataset index (admin/config/search/search_api/index/datasets/fields)
4. `drush fr dkan_sitewide_search_db -y`
5. `drush sapi-c datasets`
6. `drush sapi-r datasets`
7. `drush sapi-i datasets`
8. `drush fra -y`
9. Remove the 'Add Dataset' link from main menu
10. Remove the 'Datasets' link from main menu that point to the old /dataset url for search
11. `drush rr`

8.1.54 DKAN 1.11 Release Notes

Check out what's new in the latest version of **DKAN 7.x-1.11**! Have questions or thoughts? Let us know on our public DKAN Repo with issues or chat with us in our Gitter room.

New Features and Enhancements

New roles and permissions module

We made a new module ("DKAN Permissions") to replace the previous "Sitewide Roles and Permissions" module. The previous feature largely left it to the administrator to determine user roles and what each role could do. While

administrators will still have the ability to customize user roles and permissions, we wanted to make it easier to use DKAN right after installation. Learn more here: DKAN roles and permissions. Read up on how this standard DKAN feature makes user management easier out-of-the-box.

Group roles and permissions in a new module

The old "Sitewide Roles and Permissions" module also included group roles and permissions. As the new "DKAN Permissions" replaces the old module, groups were also updated. The capabilities of groups, the roles and permissions within, and how groups interact with the rest of a DKAN site have been moved to a new module, "DKAN Dataset Group Perms". Learn more here: DKAN group roles and permissions.

Re-designed dataset/search page

In this version of DKAN, we improved the layout and visual appearance of the datasets page to make it easier for site visitors to browse, search and find content. In previous versions, the search box on the homepage was the only way for site visitors to search all content types. The re-designed datasets page directs site visitors to a page with results pre-selected to filter the datasets content type. Site visitors have the option to add and remove filters on the datasets page in order to browse all content types as well use the search bar at the top of the page to find specific content.

Related visualizations on resource page enhancement

The visualization entity on DKAN makes it easy for content contributors, editors, and site managers to create charts, graphs and maps based on resources in their DKAN data catalog. With the resource-visualizations enhancement, site visitors will see all the visualizations made with a resource as related content at the bottom of the specific resource page.

Minor Upgrades and Improvements

User experience improvements

- **Dataset page layout:** The dataset page now has a cleaner layout and overall better visual design that is a more intuitive user experience. We also added icons for content types that are standard across the site, so that site visitors can quickly recognize what type of content they are clicking on.
- **User profile image UX:** In this release we removed the confusion in the user experience of changing a user's profile image including the existing Gravatar integration.
- **Groups view:** We changed the default number of groups displayed on a page from 10 to 12. This keeps the layout of groups evenly separated in neat rows of 3.

Patches and Bug Fixes

- **Colorizer feature bug fix**: fix to the colorizer feature on DKAN so that style changes don't disappear after running the Poormanscron Drupal module.
- **Group node page bug fix:** a quick fix to remove an extra page on the Groups page.
- **Warning message bug fix:** a quick fix to remove redundant error messages for admins upon installing DKAN.
- **Fontyourface module patch:** makes it easier to see fonts changes on your DKAN site by updating as soon as changes saved.

- **FacetAPI module patch:** patched a potential security breach that appeared through cross-site scripting.

Special Notes

Permissions/roles upgrade

As mentioned above, this release includes a new module called DKAN Permissions (dkan_permissions), which is meant to replace the old DKAN Sitewide Roles and Permissions (dkan_sitewide_roles_perms). New installs of DKAN will enable this module and ignore the old one, while existing sites will see no change (but are recommended to upgrade). See more information in the module's README file.

The command line method, including deleting the now-deprecated "storyteller" role, would look like this:

```
$ drush dis dkan_sitewide_roles_perms -y
$ drush rdel 'storyteller'
$ drush en dkan_permissions -y
$ drush fra -y
$ drush fr dkan_permissions -y
```

Group-level permissions have been moved to DKAN Dataset Groups Permissions. To start using them, enable the new module, and revert it using Features. Do this via the UI or on the command line with Drush:

```
$ drush en dkan_dataset_groups_perms -y
$ drush fr dkan_dataset_groups_perms -y
```

Avoiding PHP errors after upgrade

In some cases, changes to the codebase may cause PHP errors when the Drupal bootstrap process looks for a file that no longer exists. As always, backing up your db before upgrading is recommended. In addition, if you get to a state in which Drupal will not bootstrap successfully due to an issue with views_autocomplete_filters, try some combination of:

```
$ drush sql-query 'DELETE FROM registry WHERE filename LIKE "%views_autocomplete_
↪filters_handler_filter_string.inc%"';
$ drush cc all
$ drush rr
```

Search page redesign

This release of DKAN includes a redesigned search page, which is also the page used to browse datasets. This redesign required some changes to the included search indexes. Use the following drush commands to rebuild your search indexes, or go to admin/config/search/search_api/index/datasets and disable, re-enable and re-index the *datasets* search index.

```
$ drush search-api-disable datasets -y
$ drush search-api-enable datasets -y
$ drush sapi-r -y
$ drush sapi-i -y
```

DKAN Dataset

See the DKAN Dataset release notes for 7.x-1.11 for notes specific to the DKAN Dataset module.

8.1.55 DKAN 1.10 Release Notes

Check out what's new in the latest version of **DKAN 7.x-1.10**! Have questions or thoughts? Let us know on our public DKAN Repo with issues or chat with us in our Gitter room.

DKAN Distribution

- A number of improvements to our test infrastructure
- Improved user experience for user photos and Gravatar image fallback
- Improvements to views on group pages
- Enabled and improved UX of tools for adding existing visualizations directly into panels layouts
- Fixed an extremely annoying bug in the Colorizer module that blew away colorizer CSS every time Drupal cron ran (sites using external/system cron were unaffected)
- Added better HTTPS by loading certain external images over HTTPS
- Upgraded to Drupal version 7.x-1.41
- Added a CONTRIBUTING.md file to provide community contribution guidelines for DKAN project.

DKAN Dataset Module

https://github.com/GetDKAN/dkan_dataset

- Various improvements to dataset teaser displays.
- Support for external previews (opening catalogued resources instantly in third-party visualization services, * including ArcGIS and CartoDB)
- Support for Flaticon at module level, so vector icons work on any theme
- Minor improvements and bugfixes

NuBoot Radix Theme

https://github.com/GetDKAN/nuboot_radix

- Style improvements for "open with" button
- Fix default logo path when svg not available

8.1.56 DKAN 1.9 Release Notes

Check out what's new in the latest version of **DKAN 7.x-1.9**! Have questions or thoughts? Let us know on our public DKAN Repo with issues or chat with us in our Gitter room.

New Features and Enhancements

- Added "Data Dashboards" and "Data Stories" content types with customizable layouts
- Added Panopoly Widgets for use in Dashboards, Stories, the front page and panel pages
- Added new custom DKAN widgets for use on panel pages
- The Visualization Entity module, along with the additional Charts bundle, are now included in DKAN core
- Added a "command center" to the user's own profile page to more easily find common functions
- @font-your-face module added w/configuration for easier font admin
- Multiple security updates and other contrib module and theme updates.
- Drupal now on version 7.39.
- Many bug fixes and code cleanups
- See release notes for nuboot_radix theme and individual DKAN modules for additional release notes

Search API Database upgrade issue

This release updates the Search API DB module, which, if you are using the database backend, may break your search indexes, resulting in either a) nothing, or b) all nodes regardless of type on your search/datasets pages.

To fix, follow these steps after updating your database:

1. Go to /admin/config/search/search_api/index/datasets/edit to edit the database node index
2. Set the Server to "< No Server >" and Save. Your index will now show as "disabled."
3. Edit it again (by clicking "Edit" next to "Database Node Index") and set the Server back to "Database Server" and Save.
4. Click the "enable" link while viewing the resulting page, or edit the index again and check the "Enabled" box.
5. Once the index is enabled, you should see "0/[total] Indexed" on the index page. Click "Index Now" at the bottom of the page to re-index all your datasets.
6. Repeat for any other indexes you have running on a database server.

This issue has been identified and documented for other Drupal distributions using Search API. See release notes for the Panopoly and OpenAtrium distros on Drupal.org.

8.1.57 DKAN 1.8 Release Notes

Special note: This release contains *security updates* and should be applied immediately.

Check out what's new in the latest version of **DKAN 7.x-1.8**! Have questions or thoughts? Let us know on our public DKAN Repo with issues or chat with us in our Gitter room.

New Features and Enhancements

- Drupal core update to 7.36
- Some tweaks to display properly drupal admin pages using nuboot_radix
- Removed unnecessary drupal warning messages (colorizer, updates, etc)
- Tweaks to behat tests

- Security updates for contrib modules

Note: The files in this section should use the Markdown syntax rather than RST, for better compatibility with Github.

8.2 Changelog

8.2.1 7.x-1.17.4

- #2997 Add views patch 3054091

8.2.2 7.x-1.17.3

- #2996 Security release for Drupal 7.69

8.2.3 7.x-1.17.2

- #2989 Bump symfony/dependency-injection from 2.8.38 to 2.8.52 in /test

8.2.4 7.x-1.17.1

- #2982 #2986 Remove admin_views
- #2977 Move documentation images to S3
- #2959 Remove deprecated ahoy files
- #2973 Update list of professional help in docs
- #2972 Fix branch/tag in drupal-org.make file
- #2967 Upgrade ODSM and enable fast token browser
- 5282845 Adjustments to behat tests
- 8953e45 Restore phpunit.xml file
- #2964 Fix polling places file
- #2963 Fix Call to a member function getTimestamp() on null error
- fd331bc Update dkan sites list

8.2.5 7.x-1.17

- #2956 Update new WCAG-compliant default theme colors to be consistent across the site.
- #2957 Update frequency values on default content files.
- #2954 Update contrib modules, adjust DKAN code to support PHP 7.2
- #2951 Update default greetings on workbench email templates.
- #2908 Add DKAN Periodic Updates module
- #2804 Disable and uninstall feeds, feeds_flatstore_processor, and feeds_field_fetcher modules

- #2945 Upgrade imagecache_actions to 1.11
- #2944 Specify behavior on dkan_workflow moderation set state action to avoid extra node_save.
- #2943 Add documentation for DKAN Feedback
- #2940 Update Open Data Schema Map to 7.x-2.4
- #2935 Add documentation for DKAN custom solr
- #2931 Add patch for NULL controller class
- #2921 Fix gaps in WCAG compliance

8.2.6 7.x-1.16.13

- #2928 Update ODSM to 7.x-2.3 to fix broken RDF/JSON links
- #2918 Avoid importing/orphaning when harvest cache is not present for unavailable sources.
- #2919 Security update for imagecache_actions

8.2.7 7.x-1.16.12

- #2892 Harvest support range dates separated by dashes
- #2894 Remove user field on unpublished nodes tab on workbench moderation UI.
- #2889 Make Page Title reflect dataset/resource content
- #2893 Update Open Data Schema Map to 7.x-2.2
- #2885 Security update for uuid and tablefield modules

8.2.8 7.x-1.16.11

- #2876 Update the datastore lock code to use drupal functions
- #2879 Update dkan_workflow to add option to delete unpublished nodes
- #2880 Fix drush command that deletes orphaned datastore tables
- #2870 Remove workflow emails from dkan_workflow
- #2878 Remove hard-coded access to revisions for anonymous users

8.2.9 7.x-1.16.10

- #2874 Security update for Drupal to 7.67 (SA-CORE-2019-007)
- #2730 Behat test improvements
- #2862 Add primary key to datastore tables. Create hook_datastore_post_import to allow users to execute actions right after data has been imported into the datastore.
- #2842 Fix harvest migration message logging for batched migrations

8.2.10 7.x-1.16.9

- #2864 Tablefield security update
- #2864 Drupal core security update
- #2863 Use the data.json mediaType value to assign the resource extension during Harvest if the format is not given.
- #2809 Remove 'Feedback' from the list of content types managed by DKAN Workflow
- #2809 Apply patch to views to fix a bug on set_group_operator when using hook_views_query_alter
- #2809 Remove user picture from comment.tpl.php
- #2860 Document new datastore behavior
- #2852 Add 'count' to the list of functions supported by the Datastore API

8.2.11 7.x-1.16.8

- #2854 Wrap words on harvest summary sidebar.
- #2856 Switch to forked version of font_icon_select module.
- #2858 Security Upgrade of Services to 3.24.

8.2.12 7.x-1.16.7

- #2780 Run tests with dktl.
- #2853 Security Upgrade of Module Filter to 2.2.

8.2.13 7.x-1.16.6

- #2848 Upgrade Drupal to 7.65
- #2812 Use ISO 8601 Date Format for Temporal Coverage

8.2.14 7.x-1.16.5

- #2846 Update for views 7.x-3.21 and ctools 7.x-1.15
- #2824 Handle missing harvest source with a warning rather than mass un-publishing of content.
- #2840 Skip the dataset_changelog step when harvesting, datasets always get a new revision with out it
- #2839 Add drush command for deleting orphaned data tables drush dkan-datastore-drop-orphan-tables
- #2828 Implement the –feedback flag when harvesting to avoid SSH time outs during long harvests

8.2.15 7.x-1.16.4

- #2837 Security updates for the services and path_breadcrumbs contrib modules
- #2834 Add drush command drush ddpr to reduce duplicate dataset revisions from the same timestamp to a single revision.

- #2830 Add patch to workbench_moderation to avoid deadlock on update
- #2821 Declare file size of remote files as 'unknown' since the value given can be inaccurate
- #2823 Update dkan_dataset_validate_date() function to work on additional date formats

8.2.16 7.x-1.16.3

- #2832 Security updates for link and restws
- #2816 Fix resource titles when harvesting from Socrata - use the format when a title is not available

8.2.17 7.x-1.16.2

- #2805 Update Drupal Core to 7.63
- #2800 Update Drupal Core to 7.62
- #2792 Fix text/tab-separated-values file mime types to work with the datastore
- #2793 Fix documentation on aggregation functions min and max.

8.2.18 7.x-1.16.1

- #2778 Add check for partial date elements when running harvest
- #2787 Update rest-api documentation
- #2788 Update contrib modules: diff, field_hidden, file_entity, honeypot, libraries, menu_block, migrate, panopoly_images, panopoly_widgets, search_api, tablefield, taxonomy_menu
- #2784 Update harvest node machine name field settings to limit length and replace single hyphen
- #2785 Fix harvest support for dataset identifiers that end with a '/'
- #2782 Fix dataset and resource orphan field update on post-harvest import operations
- #2781 Use dkan_allowed_extensions() during harvest
- #2760 Remove content page create/edit permissions from the editor role
- #2775 Add check for mysql reserved words in column names when importing data to the datastore
- #2777 Add check for SelectQuery object when running dkan_datastore_api_debug
- #2776 Update recline revision
- #2767 Update workbench email notification text
- #2774 Update the license options uri values
- #2773 Fix Only variables should be passed by reference errors in dkan_datastore_api
- #2772 Fix CartoDB preview error
- #2763 Update dkan_environment key from 'stage' to 'test'
- #2762 Update Drupal to 7.61

8.2.19 7.x-1.16

- #2179 Refactor Datastore Importer
- #2627 Add support for ISO-8106 duration values in "modified" when harvesting
- #2729 Update recline to 2.1
- #2713 CSV Parser Improvements: trailing commas & escape chars
- #2740 Fix dkan_workflow anonymous access to revisions and unpublished content
- #2724 Update contrib modules: better_exposed_filters, file_entity, media, panopoly_widgets, search_api_db, uuid, date: fixes issue 2843367
- #2718 Update odsm_dkan token values for remote file
- #2707 DKAN workflow documentation updates
- #2723 Include dkan.profile to prevent undefined function in update

8.2.20 7.x-1.15.5

- #2710 Replace data stories page view with a non-search index version
- #2712 Remove unused xss code

8.2.21 7.x-1.15.4

- #2701 Update drupal core to 7.60
- #2541 Updates for php 7.1 compatibility
- #2702 Fix 'a non-numeric value encountered' error
- #2696 Peg drush to a specific version
- #2670 Update ODSM to 2.1 for php 7 compatibility
- #2692 Patch the radix theme to use local bootstrap.min.js file
- #2664 Add UI to change allowed extensions on resources
- #2668 Add environment and environment_indicator modules to DKAN
- #2656 Update workbench_moderation, drafty, and dkan_sitewide_panels to fix 'View draft' button
- #2658 Fix behat test @workflow_19 to avoid false positive result
- #2644 Automate group role assignments
- #2650 & #2634 Documentation updates
- #2637 Fix version check on available updates for empty value
- #2636 Add created date as a sort option on search pages
- #2628 Update Datastore API documentation

8.2.22 7.x-1.15.3

- #2605 Remove group member count and link from the group block
- #2604 Update select_or_other to 2.24
- #2601 Update file_resup to 1.5
- #2418, #2600 & #2606 Documentation updates
- #2599 Adds additional helper function to get remote file info
- #2592 CSS updated to add ppt/pptx file icons
- #2589 Security update for uuid (1.1)
- #2586 Switch to saving the data.json url by default when harvesting remote files rather than the effective url
- #2585 Add kml, kmz, shp to allowed extensions on resource file upload and remote file fields
- #2588 More granular field validation for datasets and resources.
- #2557 Wrap Dkan defined string to improve multilingual support

8.2.23 7.x-1.15.2

- #2576 Add redirect to search page on 404 status
- #2572 Open Data API table documentation tweaks
- #2561 Test improvements
- #2553 Add an example on how to set the dkan_map_tile_url variable
- #2552 Run linting before tests
- #2567 Fix typo in README.md
- #2544 Update contrib modules
- #2563 Remove changelog checkbox from PR template
- #2562 Fix the docker download url

8.2.24 7.x-1.15.1

- #2533 #2546 Updates docker-compose to use updated cli container
- #2542 Add theme setting to adjust hero region height
- #2531 Add table tags to allowed html
- #2527 Stop xdebug in web and cli containers for tests
- #2521 Add custom validation rules to the REST API node creation
- #2498 Fix the link to installation info on README.md
- #2463 Update circle setup to use local DEV containers
- #2485 Check the character length of harvested resource urls
- #2476 Update getExtension() and createResources() functions
- #2452 Fix front page link paths on logo and site name

- #2489 Create variable for map tile url
- #2492 Security update for file_resup module, upgrade to version 1.5
- #2358 Fix datastore error reporting when UUID is missing

8.2.25 7.x-1.15

- #2339 Adds a linkchecker to DKAN.
- #2409 Allows the services API to be able to consume datasets with the same format given when retrieving a dataset through the API
- #2434 Disable option to assign content to anonymous user
- #2431 Fix calls to the empty function that check the return value of a function or method
- #2405 & #2427 Improve tests

8.2.26 7.x-1.14.7

- #2473 Security upgrades to Drupal 7.59 and Media 2.9
- #2470 Get harvest tests passing
- #2468 Update urls to use https rather than http
- #2467 Update command center menu links
- #2464 Move functions out of the dkan_sitewide_menu.install file

8.2.27 7.x-1.14.6

- #2461 Update CONTRIBUTING.md
- #2456 Remove exra docker dir
- #2458 Fix the chrome issue
- #2450 Incorrect Readme Links

8.2.28 7.x-1.14.5

- #2435 Update Drupal core to 7.58
- #2412 Update installation instructions

8.2.29 7.x-1.14.4

- #2404 Avoid undefined offset on nuboot_radix/includes/panel.inc line 45
- #2402 Update Drupal core to 7.57
- #2375 Improvements to automated tests

8.2.30 7.x-1.14.3

- #2377 Upgrade entity to 1.9

8.2.31 7.x-1.14.2

- #2359 Upgrade filefield_sources to 1.11
- #2341 Modifies curl settings so url headers and info can be properly acquired with curl
- #2370 Patch features to add new line after <?php
- #2346 Updated Help page; can add your company to listing
- Remove out-dated owners.md file
- #2345 Site Manager Monthly Maintenance checklist
- #2347 Moving the "Adding New Content" page to main admin doc tree
- #2321 Update @resource_all_09 scenario
- #2327 Add results count to dkan admin views
- #2337 Pin sphinx to 1.5.6
- #2332 Remove redundant debug output

8.2.32 7.x-1.14.3

- #2377 Upgrade entity to 1.9

8.2.33 7.x-1.14.2

- #2359 Upgrade filefield_sources to 1.11
- #2341 Modifies curl settings so url headers and info can be properly acquired with curl
- #2370 Patch features to add new line after <?php
- #2346 Updated Help page; can add your company to listing
- Remove out-dated owners.md file
- #2345 Site Manager Monthly Maintenance checklist
- #2347 Moving the "Adding New Content" page to main admin doc tree
- #2321 Update @resource_all_09 scenario
- #2327 Add results count to dkan admin views
- #2337 Pin sphinx to 1.5.6
- #2332 Remove redundant debug output

8.2.34 7.x-1.14.1

- #2280 Fix contextual filter on list_of_users_groups view
- #2284 Change the default title for harvested resources to use the format value rather than url value
- #2265 Fix hero background image setup
- #2288 Use geojson icon for GeoJSON files in listings
- #2283 Share MySQL Credentials With Docker Containers
- #2273 Remove dkan_workflow_permissions dependency
- #2275 Docker Proxy Config Updates
- #2263 Fix harvest validation for accessLevel values
- #2249 Change timezone handling to 'none' on temporal coverage field
- #2103 Fixed field mapping defaults for open data schema APIs
- #2237 Migration of CircleCI from V1 to V2

8.2.35 7.x-1.14

- Added new module Token Tweaks to allow ui control of token recursion level in effort to help avoid memory issues with Token (see https://www.drupal.org/node/1058912)
- #2208 Fix access to groups field when adding a resource inside the dataset form.
- #2188 Move DKAN repository to GetDKAN organization in Github
- #2033 Remove "author" facet from search page (generally never relevant information)
- #2205 Update docs files with organization changes (NuCivic to GetDKAN)
- #2206 Update autocomplete_deluxe, better_exposed_filters, file_entity, media, media_youtube, tablefield and views to latest version
- #2189 #2190 Documentation updates (Add help, license, dkan sites pages. Update site manager guide. Update dkan README.)
- #2186 Move the 'back to dataset' local task position so that the 'revisions' link position matches the standard location.
- #1979 Allow users to set Groups on standalone resources.
- #2151 Adding print styling so that images, table rows, and charts work with page breaks.
- #2141 Exclude header search form from printed output
- #2174 Fix access to revisions for anonymous users when dkan_workflow is enabled.
- #2175 Removed video content from the About page. Removed or replaced references to Granicus.
- #2075 DKAN Harvest: Modified batch process to split harvest migrations in chunks.
- #2145 Front page hero image no longer requests empty URL if variable empty.
- #2124 Add results count and increase results per page on the harvest dashboard view.
- #2082 Add UI for generating ODSM file caches.
- #2058 Improve messages displayed when creating a harvest source.
- #2058 Fix harvest bug when filtering by a key that has 0 matches.

- #2102 Hide toggle buttons inside preview embeds.
- #1946 Fix Datastore API limit not applying consistently.
- #2109 Fix ODSM language token to solve validation errors.
- #2019 DKAN Workflow: Fix 'My Drafts' view to restore content to original user after rejection.
- #1970 Adds topics to harvested datasets on migration.
- #1975 Add landingPage to data.json feed.
- #1967 Added search title to /search panels.
- #1997 Recline.view.nvd3.js: Fixed tooltips that were not being displayed when using some percentage formats.
- #1988 Count and delete resources on Harvest source deletion.
- #1951 Fixed display of number of Groups a user belongs to on the user profile page.
- #1968 Fixed "Reset" button on search page to also resets facets.
- #1985 Add documentation for resource behavior on harvested datasets.
- #1907 Added back TAB delimiter option for datastore imports.
- #1955 Update ahoy docker tooling for local environment usecase.
- #1903 Add support for tab-delimited files (TSV) in resources/recline preview.
- #2001 Improve REST API documentation.
- #1991 Fix host IP replacement on script used to recconect the MySQL container.
- #1942 DevOps: Restore drush commands to deal with orphaned resources.
- #1943 Update tests to be more compatible with client sites, multiple testing and devops improvements.
- #1962 Recline.view.nvd3.js: Fixed display of data with zero as values.
- #1888 Added validation on axis tick settings on visualization_entity.
- #1725 Fix *ahoy dkan uli* output login link.
- #1881 Fix yaml syntaxt to make it standard.
- #1853 Update chart documentation.
- #1839 Add help text and improve UI on the chart form for better clarity.
- #1827 Remove option to import TABed CSV files into datastore.
- #1807 Add admin_views module for enhanced content and file filtering.
- #1990 DevOps: Automatically populate required fields when running behat tests.
- #1938 DevOps: Allow behat tests to be skipped in circle.
- #1842 Add help text to explain the pager on data previews.
- #1542 Creates migration on harvest source import.

8.2.36 7.x-1.13.7 2017-08-22

- #2064 Update entityreference to 1.15
- #2061 Update views to 3.17
- #2060 Update tests to adjust to client customizations.

- #2060 Removed unused modules menu_token and remote_file_source from drupal-org.make

8.2.37 7.x-1.13.7 2017-08-22

- #2064 Update entityreference to 1.15
- #2061 Update views to 3.17
- #2060 Update tests to adjust to client customizations.
- #2060 Removed unused modules menu_token and remote_file_source from drupal-org.make

8.2.38 7.x-1.13.6 2017-07-28

- #2036 Add file_resup to the list of up to date projects.
- #2035 Change the 'Datasets' field on the Resource form to use autocomplete.
- #2034 Add language selectors to the 'chosen_jquery_selector' configuration.
- #2020 Module upgrades: fieldable_panels_panes, media, and search_api.
- #2018 Additional test/devops improvements needed for deployment. See #2012, #2014, #2015 and #2016 for specifics.
- #2025 Add configurable arguments via Behat Contexts.
- #2021 Add unique tags to more test scenarios.

8.2.39 7.x-1.13.5 2017-07-14

- #2003 Fix dkan_bueditor_markdown_install(), which was using a variable before it was initialized.
- #2003 Fix to dkan_update_7016, which assumed the bueditor ID was always '5' and failed when it was not.
- #2003 Fix dkan_topics_field_formatter_view(), which does not check if a term exists before calling it by tid.
- #2003 Make Harvest tests clean up after themselves more completely (resources, other nodes were not being cleared).
- #2003 Fix issues with registering/deregistering and rolling back default content migrations on install/uninstall.
- #2003 Fix dkan_migrate_base warning: wrong type supplied to foreach.
- #1963 Allow Behat dkanExtension to handle custom fields via devel generate
- #1938 Allow skipping of features test via config
- #1970 Automatically populate required fields when running behat tests.

8.2.40 7.x-1.13.4 2017-06-30

- #1983 Apply services security update 3.20 for DRUPAL-SA-CONTRIB-2017-054
- #1877 Fix broken update that tries to migrate fields that may not exist (primarily due to Federal Extras upgrade)
- #1960 Update field_group_table to 1.6

- #1950 Update modules: chosen to 2.1, context to 3.7, date to 2.10, diff to 3.3, double_field to 2.5, entityreference to 1.4, menu_badges to 1.3, module_filter to 2.1, panels to 3.9, panopoly_widgets to 1.45, panopoly_images to 1.45, rules to 2.10, roleassign to 1.2, search_api to 1.21, search_api_db to 1.6, simple_gmap to 1.4, token to 1.7, uuid to 1.0, views_bulk_operations to 3.4, workbench_email to 3.12, drafty to 1.0-beta4, file_entity to 2.2, media to 2.8, media_youtube to 3.4, menu_token to 1.0-beta7, views to 3.16, field_group_table to 1.6, radix to 3.6 and file_resup to revision 6cf030c.
- #1965 Update Drupal to 7.56.
- #1925 Fix issues with remote_stream_wrapper and recline causing Harvest and Search API to fail on certain larger resources
- #1916 Federal extras: Fix spelling of "Government" in group label.
- #1939 Removed references to DKAN Demo files from PHPUnit tests.
- #1932 Fix windows delimiter for datastore_fast_import.
- #1906 Refactor datastore import to restore drush command (dsu).
- #1898 Add validation to field data dictionary when strict POD validation is enabled.
- #1920 Update Bureau codes for Open Data Federal Extras.
- #1927 Fixed adding shapes to the spatial geographic coverage area field.
- #1924 Fixed dkan_update_7012, if a site is still using a deprecated module, the data is not stripped from the db.
- #1921 Drop the description label from the Harvest Source node view page.
- #1912 Added 'ahoy dkan unittests' command.
- #1917 Fixed 'ahoy dkan reinstall' command.
- #1865 Removed old dkan_dataset_api module.
- #1899 Override default error message on duplicate path aliases with more information.
- NuCivic/visualization_entity#72 Switch to relative path for uploaded resources on visualization entity charts
- #1836 Update language codes to use dashes rather than underscores for POD compliance.
- #1836 Fix the POD and DCAT language output to use key values rather than label values.
- #1891 Expose publishing options for content creators who do not belong to a group.
- #1891 Disabled authoring information fields for content creators who do belong to a group.
- #1901 Upgrade media module to 2.1
- #1897 Fixed handling of dataset language setting to pass DCAT validation.
- #1896 Fixed bug on open_data_schema_map: Multiple urls cause data.json describedBy field to be truncated.
- #1890 Fixed the chosen widget exclusion settings to not apply to group search filters.
- #1887 Removed unnecessary 'Select' button from the remote file input field on the resource form.
- #1869 Add validation check on the API or Website URL field to require a full url.
- #1883 Removed required status from the Rights field if the public access level is set to 'none'.
- #1871 Better handling of remote resources with BOM.
- #1873 Fix dkan_workflow_permissions: Allow Workflow Contributors and Workflow Moderators to view stale drafts and Workflow Moderators to view stale reviews.
- #1858 Fixed default panelizer setting for page node title.

- #1874 Fix dkan_update_7013 to not use drush commands.
- #1876 Removed hard coded colors in the menu.scss file.
- #1825 Adapt the 'dataQuality' input value druing POD harvest.
- #1817 Load empty cells as null in fast import.
- #1846 Adds a hook_update to exclude the data dictionary field from using the markdown toolbar.
- #1813 Update the groups page view to sort alphabetically rather than by post date.
- NuCivic/recline#89 Only load previews for resources using files; API/website links should always display iframe.
- #1841 Fixed mis-named function on dkan_dataset_content_types.api.php
- #1814 Update dkan_workflow_permissions to rebuild permissions on enable/disable.
- #1810 Stop throwing exception on tables with only numeric columns, to prevent preview breaking.
- #1834 Use last day of year (December 31) for 4 digits end year during temporal POD field import.
- #1832 Fallback to modified date for field_harvest_source_issued if not available during POD Harvest.
- #1828 Field Frequency Harvest POD integration is missing support for "irregular" value.
- #1820 Use "accessURL" during resources harvest if "downloadURL" field is not available.
- #1852 Allow the use of multi-polygonal data for Dataset Spatial field.
- #1857 Fixed publishing options not accessible when dkan_workflow is enabled.
- #1932 Fix windows delimiter for datastore_fast_import.

8.2.41 7.x-1.13.3 2017-04-18

- #1863 Update restws module to v2.7
- #1859 Fixed update hooks to correctly set jquery version and remove old modified source date field
- #1864 Update media to 2.0 and remove patch 2534724.
- #1829 Fixed missing properties on warning message during datajson harvest cache.
- #1802 Better support for Issued and Updated dataset properties from harvested sources.
- #1821 Remove redundant CSS load in dkan_dataset.
- #1726 Fixed broken links in search results to nodes without aliases
- #1792 Remove Windows reserved characters from default content filenames for better Windows support
- #1815 Support importing Temporal Coverage during POD Harvest.
- #1785 Fix error messages when installing DKAN with the UI.
- #1786 Remove horizontal tab field from the group content type, fix margin on group block.
- #1786 Add hook to auto-assign editors the og administrator role.
- #1752 Added option for using quote delimiters when fast import is enabled for dkan datastore. Fixed error message when an import fails on DKAN Datastore Fast Import.
- #1703 Refactor Datastore API module, fixing some caching issues and improving joins
- #1804 Add project = dkan to dkan_sitewide.info to fix errors on update screen.
- #1811 Apply patches to the rules module that can prevent unnecessary DB lockouts.

8.2.42 7.x-1.13.2 2017-03-16

- #1803 Fix broken access to featured groups sort order view.
- #1796 Fix Harvest support for contact name and contact email.
- #1795 Update front page test on topics.feature with @customizable.
- #1783 Update services to 3.19
- #1794 Update visualization_entity to 1.1
- #1781 Update views to 3.15 and remove patch 1388684.
- #1751 Add POD validation link to command center menu for site manager access.
- #1762 Re-number update hooks that were mixed up during release integration.
- #1709 Changed function dkan sitewide conversion homepage to fix problem with link entity attribute. This attribute need to be a boolean.
- #1742 Fix home page HTML <head><title> so that it's just the site name (not the node or panel title)
- #1747 Update DKAN API link to use the RTD documentation page.
- #1730 Fix logic error for front page in theme causing error messages on homepage
- #1728 Added a tag @defaultHomepage to a topic test which relies on homepage content.
- #5807 Fix panelizer permissions to hide 'Customize Display' button
- #1744 Grant Site Manager role Harvest Dashboard actions.
- #1776 Fix 500 errors when linking or uploading geojson files in resources.
- #1767 Better handling of empty values by datastore_api. Also see NuCivic/feeds_flatstore_processor#9

8.2.43 7.x-1.13 2017-02-28

- #1719 Added site details to settings nuboot_radix to allow change site name, slogan, e-amail address for site manager.
- #1717 Upgrading Drupal to 7.54
- #1705 Added css for chart visualizaitons, fixes issue in IE.
- #1695 Upgrade better_exposed_filters to v3.4.
- #1702 Update branding.
- #1369 Added ReadTheDocs integrations and docs folder for centralized documentation
- #1701 Renamed test files.
- #1691 #1682 Update resource tests to work with client site variations.
- #1694 Moved functions from FeatureContext to DKANExtension.
- #1687 Added a function in dkan_sitewide to check if a specific page is the front page.
- #1693 Improve module and user cleanup inside dkan workflow context after run tests
- #1669 Update leaflet library to v1.0.2 and leaflet markercluster to v1.0.0. Unift leaflet libraries (was using a separate version for recline module)
- #1684 Security update for autocomplete_deluxe

- #1670 Upgraded to new 1.0 release of Visualization Entity module
- #1663 Add install hook to DKAN Workflow to force a Features revert of dkan_sitewide_menu, to make sure Workflow links added correctly
- #1348 Remove Panels IPE from dataset and search pages; in DKAN, we only want to show IPE for panelizer layouts, not templates or other "sitewide" pages.
- #1046 Update user profile page search to be consistant with the rest of the site and moves user info to sidebar block.
- #1096 Fixes typo in "add data story" link in command center menu
- #1069 Add topic icons to the drop down menu
- #1085 Renamed the default HTML text format to 'Markdown HTML'
- #1440 Improved markdown text editor toolbar
- #1110 URLs on dataset's additional info are now properly displayed as links.
- #1130 Removed warnings about undefined permissions when visualization_entity_choropleth_bundle module is enabled.
- #1130 Removed warnings about undefined permissions when visualization_entity_geojson_bundle module is enabled.
- #1130 Removed specific visualization entity charts permissions from dkan_extension.
- #1130 Added test feature to validate that users with Editor, Content Creator and Site Manager roles are not able to access to admin pages.
- #1259 Default content was improved and now it's generated based on fixtures.
- #1152 Removed omega theme.
- #1152 Removed delta module.
- #1152 Moved Nuboot Radix theme into core build.
- #1152 Change admin theme setting to use default theme rather than nuboot radix
- #1221 Page content type added to panelizer and now defined in features, with in-place editor enabled for page content.
- #1222 Page link added to the command center menu.
- #1301 Added Open Data Federal Extras module into DKAN Core.
- #1297 Fix for relative paths to ensure links still work when a site is installed into a subdirectory.
- #1358 Removed field mapping warnings during import of default content.
- #1376 Removed conditional_fields from dkan_topics.
- #1387 Upgrade Features to 7.x-2.9
- #1387 Make date facet use "day" granularity.
- #1387 Remove ARC2 library.
- #1439 Disable pathauto for content created using dkan_fixtures.
- #1403 Removed warning message when try edit resource without dataset.
- #1468 Added PHP Unit tests configurations to make them run on DKAN.
- #1463 Added hook_uninstall to federal extra module for remove fields after uninstall module.

- #1459 Removed redundant definition for CKAN package_list endpoint from open_data_schema_map_dkan, allowing that feature to be in default state after install.
- #1456 Update ctools to version 1.10, which fixes some PHP7 incompatibilities
- #1440 Added links to ODSM page and DCAT validation under Site Configuration in the command center menu
- #1367 Update open data schema module to render a valid output passing validation (dcat and rdf).
- Added upgrade hook to clean DB after removal of 'conditional_fields', 'entity_rdf', 'rdfui' and 'rdfx'.
- #1527 Added upgrade hook to remove deprecated test and theme directories.
- #1537 Fixed popular tags view.
- #1685 Moved front page search block from dkan_sitewide_demo_front to dkan_sitewide.
- #1562 Fixed link to group page on group node teaser when site has clean urls disabled.
- #1534 Added update hook to disable dkan_default_content on upgrades.
- #1565 Updated visualization_entity.
- #1556 Added fix to display the 'Request membership' link only to users that are logged in.
- #1582 Remove content padding on data extent and social blocks.
- #1606 Added DKAN Extension.
- #1508 Upgrade workbench_moderation to v3.0.
- #1595 Upgrade entity to 1.8
- #1598 Upgrade manualcrop to 1.6
- #1598 Upgrade markdown to 1.5
- #1598 Upgrade panels to 3.8
- #1592 Upgrade ctools to 1.12
- #1592 Upgrade entityreference to 1.2
- #1592 Upgrade libraries to 2.3
- #1592 Upgrade services to 3.17
- #1592 Upgrade simple_gmap to 1.3
- #1592 Upgrade tablefield to 2.5
- #1592 Upgrade entityreference_filter to 1.7
- #1622 Upgrade fieldable_panels_panes to 1.11
- #1596 Upgrade radix to 3.5
- #1598 Upgrade search_api to 1.20
- #1603 Upgrade media to 2.0-beta13
- #1603 Upgrade panopoly_widgets to 1.41
- #1603 Upgrade panopoly_images to 1.41
- #1604 Upgrade file_entity to 2.0-beta3
- #1605 Upgrade link_iframe_formatter to 1.1
- #1654 Assign workflow supervisor role to site manager users when dkan workflow is enabled.

- #1645 Switch to forked version of the spectrum library to address accessibility issues.
- #1642 Removed eva module.
- Fixed CSV column validation on visualization entity.
- Fixed retrieve of version information on chroma.js.
- #1678 Add patch to fix panopoly_widgets overrides OOB.

DKAN Datastore:

- #1387 Added DKAN Datastore module into DKAN Core.
- #1599 Greatly expand the datastore API to support aggregation functions, better joins, multiple queries. See dkan_datastore_api's README file.
- #1599 Fixed bugs when running the datastore via cron.

DKAN Harvest:

- #1676 Harvested content is published even if dkan_workflow is enabled.
- #1287 Added DKAN Harvest module into DKAN Core.
- Added UI to add harvest sources
- Added UI to manage harvest sources and datasets
- #1446 Fix dkan harvest feature overridden after install.
- #1440 Add link for site managers to create harvest source in command center menu.
- #1472 Added Batch API on 'Harvest now' action.
- #1482 Added Batch API on the 'Preview' page.
- #1531 Added 'Add Source' shortcut on Harvest Dashboard pages.
- #1531 Fixed breadcrumb on harvest sources pages: Preview, Manage Datasets, Events, Errors.
- #1434 Fixed harvesting of resources with remote files with redirects.
- #1588 Fix "Manage Datastore" tab leaking to the harvest source node.
- #1588 Add icons for harvest source tabs.
- #1634 Fixed the dataset count displayed on the harvest preview message.
- #1640 Added support for –limit, –instruments, –skiphash and –idlist options on drush commands.
- #1658 Fix 508 compliance errors on admin and node view pages.
- #1650 Removed links to Harvest Source nodes from main menu.
- #1672 Prevent harvest source machine name from containing forward slash character.
- Improved PHP Unit tests.
- #1468 Add support "Compound" fields on Filters/Overrides/Excludes/Default in DKAN Harvest.
- #1488 Replace field notes by field body in harvest source content type.

DKAN Migrate Base:

- #1387 Added DKAN Migrate Base module into DKAN Core.
- #1462 Removed row from migration map table when a dataset is deleted.

DKAN Workflow:

- Add patch for workbench_moderation to avoid php shutdown function errors.
- #1690 Added patch for workbench_moderation: Invalid argument supplied for foreach() 2360973.
- #1712 Added a filter in the workflow.feature to avoid issues for the amount of already existing nodes.
- #1707 Isolate dkan workflow tests related to emails.#1069
- #1715 Moved the update of the roleassign_roles variable from dkan_workflow_permissions to dkan_workflow.
- #1677 Fixed panels-related bug where, if a dataset had both a "published" and "draft" version, the published would show in the draft tab.
- #1438 Updated DKAN workflow vbo customizations to not affect other vbo forms.
- #1481 Updated workbench_email from 3.9 to 3.11
- #1667 Fixed admin menu source when dkan_workflow is enabled.
- #1663 Moved workbench links from dkan_sitewide_menu to dkan_workflow.

DKAN Dataset:

- #1696 Disable/uninstall dkan_dataset_api if enabled because it is deprecated in favor open_data_schema_map.
- Leaflet draw widget usability improvements
- #1636 Fix validation on resource forms when multiple resource type fields are populated
- #1626 Fix 508 compliance issues for leafleat draw widget
- #1387 Added DKAN Dataset module into DKAN Core.
- #1589 Fix geofield map button toggle function.
- #1345 Added default image for groups.
- #1377 Add preview support for resources of many more formats, outside of recline.js. Support added for JSON, geojson, XML, ArcGIS REST, WMS, images, PDF, ZIP
- #1377 Fix bugs in resource mimetypes and previews.
- Add more frequency update options to the dataset creation form https://project-open-data.cio.gov/iso8601_guidance/#accrualperiodicity
- #1301 Groups field on Dataset form was modified to be a single select with all group options and it was moved from the tabs section to the main section of the form.
- #1301 Groups field was hidden on the resource form since groups are assigned automatically based on the parent datasets.
- #1301 Moved and renamed the following fields from ODFE to DKAN Core: field_is_part_of, field_data_dictionary_type, field_landing_page, field_pod_theme, field_conforms_to, field_rights, field_language.
- #1301 Added hook updates to handle the renaming of the old ODFE fields removing the 'odfe' namespacing.
- #1301 Added hook update to migrate the content from field_odfe_category to field_pod_theme on Datasets.
- #1301 Added new option on DKAN Dataset Forms to enable POD based validation on Dataset form.
- #1301 Added new option on DKAN Dataset Forms to enable Groups field validation on Dataset form.

- #1280 Fixes to groups UX: Group body field now labeled "Description," longer group descriptions do not get cut off after 200 characters on group page, and extraneous "about" tab on Group node edit form removed
- Add new "modified date" field, hidden in form, to datasets. This is for dkan_harvest compatibility; saves sources dates into a separate field so they aren't affected by node changes. ODSM mappings also updated.
- Fix keywords/tags creating broken links when containing spaces, and add missing keywords to default content.
- #1477 Fix ODSM permissions for non-admin roles.
- #1532 Fixed text on 'Download All' button to not display HTML.
- #1570 Removed 'Back to dataset' button on standalone resources.
- Better support of downloading remote files from the resource view page "Download" button.
- #1553 Removed warning when a resource is created without title.
- #1591 Add limit to proxy resources.
- #1576 Improved download of remote files.
- #1670 Fix 508 compliance issues for visualization_entity_charts.
- #1669 Fix 508 compliance issues for leaflet draw widget.
- #1434 Add patch to remote_stream_wrapper to fix memory exhausted errors.
- #1652 Renamed 'filefield_remotefile' as 'filefield_dkan_remotefile'.
- #1671 Add patch to field_group to avoid DOM-based cross-site scripting vulnerabilities.

DKAN Topics:

- #1159 Added a test for creating a topic term.
- #1486 Make the icon field required if the icon type is set to 'font'.
- #1656 Fix topics icon selector functionality.

8.2.44 7.x-1.12.13 2017-01-04

- Fix broken recline (resource CSV) preview embeds caused by how recline module loads bootstrap
- Fix publisher token in open_data_schema_map_dkan - was showing only URL rather than publisher name
- Add patch to remote_stream_wrapper to avoid memory exhausted on big files.

8.2.45 7.x-1.12.12 2016-12-15

- Start caching recline "embed" pages (previews rendered with no headers/sidebars for iframe) (recline module)
- Upgrade Drupal core to 7.52
- Update ctools to 1.11
- Fix link to group page on group node teaser when site has clean urls disabled
- Fix "format" facet collapsed even when selected on the search page.
- Fix resources not synced with datasets when upgrade from 1.11
- Update media to 2.0-beta13

- Fix hidden body field on page content type
- Improve access check/security on datastore pages - unauthorized users could perform certain datastore functions
- Use dashes in links to tags to avoid "page not found" errors

DKAN Datastore:

- Add limit to proxy resources.
- Better support of downloading remote files from the resource view page "Download" button.

8.2.46 7.x-1.12.11 2016-10-20

- Fixed a bug in Recline regarding file objects and node forms that caused errors when using the "view changes" button on Dataset or Resource edit form
- Center group images in "Groups" page and group node page sidebar
- Add "2x" "3x" etc to datset teasers when more than one resource of a particular format present.
- Update the default jquery library setting from 1.7 to 1.10
- Fix topics menu and facet links if special characters are used in topic terms.
- Fixed dataset form redirect when validation fails, was sending user to node/add/dataset rather than node/%/edit
- Patch fontyourface to remove from the "standard text" selector, to prevent unpredictable results from this option
- Fixed issue in open_data_schema_map with "Data Dictionary" field not displaying URL in data.json file

8.2.47 7.x-1.12.10 2016-09-07

- Fix JS error on IE browsers preventing previews from appearing
- Provide upgrade paths for older sites upgrading to newer BU editor config

8.2.48 7.x-1-12.9 2016-08-31

- Fix data dictionary field to use html input format and improvements to the field's display and help text.
- Update dkan_workflow content type legend to include all content types with correct icons.
- Fix to new bug introduced in recline module in 1.12.7; previews would not display if file smaller than max preview size, even if loading from datastore.
- Fix a problem with paths for image icons for topics. Uploaded image icons will now display correctly in the Topics menu.

8.2.49 7.x-1.12.8 2016-08-19

- Patch panelizer module to correct bug introduced in previous release. See release notes for details.
- Add validation to Group form to prevent duplicate groups from being created
- Update and patch panels and panelizer to fix critical security issue. See release notes.
- Patch panelizer module to correct bug introduced in previous release. See release notes.

8.2.50 7.x-1-12.7 2016-08-09

- Update contrib modules uuid, services, manualcrop, markdown, panelizer, panopoly_widgets, panopoly_images
- Fix bug that produced a "Cannot use a scalar value" error when trying to add a content pane to a data story node in panels.
- Change theming on resource to remove iframe to "API or link URL" if remote file field also populated
- Update options in Dataset "frequency" field to reflect standards for DCAT and Project Open Data
- Provide "data proxy" to serve remote CSVs through local domain and resolve cross-origin issues with previews n these resources
- Some changes to the Resource node form to improve UX of links and file attachments, especially by replacing the label "Link to an API" to "API or Website URL"
- Fix bug on relative paths for links in theme template files.
- Fix minor typo in DKAN topics menu link

8.2.51 7.x-1.12.6 2016-07-26

- Sanitize theme output for facets to avoid any security issue from search input
- Fix a potential XSS vulnerability in search facets by adding some validation hooks for facet input
- Theme updates back ported from dev branch, including topics icons in topics drop-down menu
- Add RoleAssign module, update DKAN Permissions to give site managers permission to assign roles.
- Make links and emails in metadata pages clickable

8.2.52 7.x-1.12.5 2016-07-13

- Update restws to 7.x-2.6 (critical security update)
- Update to latest version of recline.js to fix map tiles (Mapquest discontinued open access)
- Upgrade Drupal core to 7.50

8.2.53 7.x-1.12.4 2016-07-12

- Update permissions in DKAN Permissions module to allow anonymous users to access RDF endpoints for individual datasets.

8.2.54 7.x-1.12.3 2016-06-30

- Open Data Schema Map update: improves performance of package_show endpoint
- Point to new release of visualization entity, which cleans up some make files and brings visualization_entity_charts into the same project repository.
- Fix editor permissions to allow access to visualizations list from admin menu
- Upgrade Views to 7.14
- Upgrade Drupal core to 7.44

- Add
<h2><h3> <iframe> to allowed html tags
- Renamed the default HTML text format to 'Markdown HTML'
- Improved markdown text editor toolbar
- Stories main menu link removed from stories view
- Groups main menu link removed from page manager groups page config
- Main menu links added on install function rather than through features
- Add body field to data_dashboard content types
- Front page page manager config moved to the dkan_sitewide_demo_front feature
- Front page group views moved to the dkan_dataset_groups feature
- Add dkan_ipe feature to simplify the in-place editor interface, includes addition of the panels_curator module

8.2.55 7.x-1.12.2 2016-06-09

- Fix problems in visualization entity charts creation due to version of CSV.js referenced in recline.make

8.2.56 7.x-1.12.1 2016-06-07

- New version of Visualization Entity Charts fixes a number of UI bugs and restores support for Google Sheets and Data Proxy
- Fix XSS vulnerability by adding sanitization for titles on workbench view
- Upgrade of: Colorizer, Admin Menu Source, Honey Pot, Panopoly Widgets, Panopoly Images, Pathauto, Rules, Restws, Manual Crop, Image Cache Actions, Features, Search API, Organic Groups, Chosen, Date, Entity, Facet API, Facet API Bonus, Facet API Pretty Paths, UUID, Views, Select or other, Remote Stream Wrapper, Link, Libraries, Beauty Tips, Gravatar, OG Extras, Services and Workbench Email.
- Upgrade of Panopoly Widgets, Panopoly Images and Fieldable Panels Panes.
- Patch colorfield module to avoid incorrect status error. See https://www.drupal.org/node/2696505

8.2.57 7.x-1.12 2016-04-20

- Fixed regex used to rewrite font file paths on Dkan Topics, which was breaking filepaths in data.json
- Rename dkan_featured_topics to dkan_topics before release
- Add "data dashboard" creation link to user admin menu (in dkan_sitewide_menu)
- Patch DKAN core file module to fix managed file problem https://www.drupal.org/node/1903010#comment-10118508
- Standardize facet titles between user, search and group pages
- Add dkan_featured_topics module and relevant updates to dkan_dataset and nuboot_radix
- Upgrade Drupal core to 7.43
- Enable pretty paths for search page
- Change search page path from /dataset to /search and redirect /dataset to /search for backward compatibility
- Add big file upload support through file_resup module

- Add support to import big files using mysql statement load data infile
- Added Dkan Workflow module
- Upgrade Fieldable Panels Panes to 1.8
- Added hook_update to remove 'Add Dataset' and 'Datasets' links from main menu. The 'Datasets' link is now added by the dkan_sitewide_search_db feature.
- Removed 'taxonomy_menu_vocab_parent_dkan_topics' variable from info file to get the 'DKAN Featured Topics' feature back into 'Default' state.
- Upgrade of Panopoly Widgets, Panopoly Images and Fieldable Panels Panes.

8.2.58 7.x-1.12 2016-04-01

DKAN_Datastore:

- Add update function enable field_hidden module

NuBoot Radix Theme:

- Add managed file support to logo and hero image
- Add check on filenames to remove spaces and special characters that can break functionality.
- Make date facets consistent with other facets styling
- Upgraded to use Radix 3.3.
- Fix node–search-result.tpl.php file to check the $group_list variable before printing markup

8.2.59 7.x-1.11 2016-02-02

DKAN Dataset:

- Fix some bugs breaking the resource links on dataset pages
- Re-arranged group links on group landing pages; membership links now in sidebar rather than tabs
- For long fields in the "additional info" custom metadata table on dataset pages, limit the row height for fields displaying extremely long text values
- Add a list of related visualizations (if using the [Visualization Entity](https://github.com/NuCivic/visualization_entity) module) to resource pages
- Move group-level permission export into new dkan_dataset_groups_perms module
- Fixed counting of datasets, broken when a dataset belongs to more than one group
- Hide "add resource" button on datasets from users who do not have permission to do so
- Numerous small improvements and bug fixes

NuBoot Radix Theme:

- Theming improvements to support new search page design
- Fix the Additional Info field in datasets to hide extra-long rows
- Fix a bug that prevented sub-themes of Nuboot Radix from retrieving their own settings
- Styling for re-arranging of group links on group pages DKAN Dataset
- Numerous small styling and code fixes

8.2.60 7.x-1.11 2016-02-01

- Re-designed dataset/search page. "Datasets" link on default menu bar now goes to a page that lets you browse and search all site content, not just datasets, but does filter by dataset. Search box will search all content by default. See note below.
- Add new dkan_permissions module, refactoring default roles and permissions and using new export method. See note below.
- Moved group permissions from old dkan_sitewide_roles_perms into new dkan_dataset_groups_perms
- Patch fontyourface module to make font changes happen instantly
- Fix adminrole implementation to avoid warning on install
- Fix (again) a bug that would make colorizer styles disappear after running "poor man's cron"
- Add to resource node page a list of visualizations built with that resource
- Number of contrib module updates
- Change the view of groups to show 12 instead of 10 nodes per page, to fit with the layout of three per line
- Patch the FacetAPI module to avoid a cross-site scripting vulnerability
- Add integration with ProboCI (http://probo.ci/) for QA builds
- Fix a bug that showed an extra pager on the group page
- Major refactor of Behat tests, including introduction of new [DKAN extension](https://github.com/NuCivic/dkanextension).
- Improve layout/ordering of blocks in the sidebar for dataset pages
- Improvement in the UX of user pictures, including Gravatar integration
- Numerous other small improvements and bugfixes

Notice: Avoiding PHP errors after upgrade

In some cases, changes to the codebase may cause PHP errors when the Drupal bootstrap process looks for a file that no longer exists. As always, backing up your db before upgrading is recommended. In addition, if you get to a state in which Drupal will not bootstrap successfully due to an issue with views_autocomplete_filters, try some combination of:

$ drush sql-query 'DELETE FROM registry WHERE filename LIKE "%views_autocomplete_filters_handler_filter_string.inc%"'; $ drush cc all $ drush rr

Search page redesign

This release of DKAN includes a redesigned search page, which is also the page used to browse datasets. This redesign required some changes to the included search indexes. Use the following drush commands to rebuild your search indexes, or go to admin/config/search/search_api/index/datasets and disable, re-enable and re-index the _datasets_ search index.

$ drush search-api-disable datasets -y $ drush search-api-enable datasets -y $ drush sapi-r -y $ drush sapi-i -y

Permissions/roles upgrade

As mentioned above, this release includes a new module called DKAN Permissions (dkan_permissions), which is meant to replace the old DKAN Sitewide Roles and Permissions (dkan_sitewide_roles_perms). New installs of DKAN will enable this module and ignore the old one, while existing sites will see no change (but are recommended to upgrade). See more information in the module's README file.

The command line method, including deleting the now-deprecated "storyteller" role, would look like this:

$ drush dis dkan_sitewide_roles_perms -y $ drush rdel 'storyteller' $ drush en dkan_permissions -y $ drush fra -y $ drush fr dkan_permissions -y

Group-level permissions have been moved to DKAN Dataset Groups Permissions. To start using them, enable the new module, and revert it using Features. Do this via the UI or on the command line with Drush:

$ drush en dkan_dataset_groups_perms -y $ drush fr dkan_dataset_groups_perms -y

DKAN Dataset:

- See the DKAN Dataset release notes for 7.x-1.11 for notes specific to the DKAN Dataset module.

8.2.61 7.x-1.10 2015-11-10

DKAN Distribution:

- A number of improvements to our test infrastructure
- Improved user experience for user photos and Gravatar image fallback
- Improvements to views on group pages
- Enabled and improved UX of tools for adding existing visualizations directly into panels layouts
- Fixed an extremely annoying bug in the Colorizer module that blew away colorizer CSS every time Drupal cron ran (sites using external/system cron were unaffected)
- Added better HTTPS by loading certain external images over HTTPS
- Upgraded to Drupal version 7.x-1.41
- Added a CONTRIBUTING.md file to provide community contribution guidelines for DKAN project.

DKAN Dataset Module:

- NOTE: 7.x-1.10 Was re-released on 2015-11-18 to address bugs in the teaser preview links.
- Various improvements to dataset teaser displays.
- Support for external previews (opening catalogued resources instantly in third-party visualization services, * including ArcGIS and CartoDB)
- Support for Flaticon at module level, so vector icons work on any theme
- Minor improvements and bugfixes

DKAN Datastore:

- Upgrade Feeds module to latest dev version

NuBoot Radix Theme:

- Style improvements for "open with" button
- Fix default logo path when svg not available

8.2.62 7.x-1.9 2015-09-17

- Added "Data Dashboards" and "Data Stories" content types with customizable layouts
- Added Panopoly Widgets for use in Dashboards, Stories, the front page and panel pages
- Added new custom DKAN widgets for use on panel pages
- The Visualization Entity module, along with the additional Charts bundle, are now included in DKAN core

- Added a "command center" to the user's own profile page to more easily find common functions
- @font-your-face module added w/configuration for easier font admin
- Multiple security updates and other contrib module and theme updates.
- Drupal now on version 7.39.
- Many bug fixes and code cleanups
- See release notes for nuboot_radix theme and individual DKAN modules for additional release notes

NuBoot Radix Theme:

- Add support for SVG logos
- New option to use solid color for background/hero graphic instead of image
- Improved Colorizer support for different page elements
- Numerous bug fixes and small styling improvements

DKAN Datastore:

- Better UX/behavior for the "add to datastore" button: https://github.com/NuCivic/dkan_datastore/pull/39
- Small bugfixes and improvements

DKAN Dataset:

- Fixed issues around the group/publisher field in search indexes and facets
- When multiple resources are available for a dataset, new option to download all as zip file
- Improvements for resource display in dataset pages and teasers
- New hooks to allow additional license options https://github.com/NuCivic/dkan/issues/447
- Numerous small improvements and bugfixes

Search API Database upgrade issue

This release updates the Search API DB module, which, if you are using the database backend, may break your search indexes, resulting in either a) nothing, or b) all nodes regardless of type on your search/datasets pages.

To fix, follow these steps after updating your database:

1. Go to /admin/config/search/search_api/index/datasets/edit to edit the database node index
2. Set the Server to "< No Server >" and Save. Your index will now show as "disabled."
3. Edit it again (by clicking "Edit" next to "Database Node Index") and set the Server back to "Database Server" and Save.
4. Click the "enable" link while viewing the resulting page, or edit the index again and check the "Enabled" box.
5. Once the index is enabled, you should see "0/[total] Indexed" on the index page. Click "Index Now" at the bottom of the page to re-index all your datasets.
6. Repeat for any other indexes you have running on a database server.

This issue has been identified and documented for other Drupal distributions using Search API. See release notes for the Panopoly (https://www.drupal.org/node/2425263) and OpenAtrium (https://www.drupal.org/node/2443025) distros on Drupal.org.

8.2.63 7.x-1.8 2015-04-02

- Drupal core update to 7.36
- Some tweaks to display properly drupal admin pages using nuboot_radix
- Removed unnecessary drupal warning messages (colorizer, updates, etc)
- Tweaks to behat tests
- Security updates for contrib modules
- DKAN_Datastore: Adding CRUD functions to Datastore.inc class
- DKAN_Datastore: Adding drush interface to CRUD functions in order to create instances from the cli

DKAN Dataset:

- Several recline.js tweaks
- Updated open_data_schema_map to include module that allows to hook your own custom output formatters (xml example included)
- Security updates for contrib modules

8.2.64 7.x-1.7 2015-02-20

- Adds Panels to DKAN for Drag and Drop layouts: https://docs.getdkan.com/dkan-documentation/dkan-users-guide/customize-dkan-pages-layouts-and-components-using-panels
- Adds Open Data Schema to DKAN https://github.com/NuCivic/open_data_schema_map#open-data-schema-map
- Adds 'saved states' to Recline data preview
- DKAN_Datastore: Remove schema patch

DKAN Dataset:

- Adds Project Open Data v1.1 schema through open_data_schema_map update
- Adds dkan data.json v1.1 complaint implementation through open_data_schema_map_dkan update
- Adds Panels implementation for dkan_dataset_groups: https://docs.getdkan.com/dkan-documentation/dkan-users-guide/customize-dkan-pages-layouts-and-components-using-panels
- Updates to dkan_dataset stylesheet to make them radix compatible
- Security updates for several modules

8.2.65 7.x-1.6 2014-11-20

- core update to 7.34
- fix issues with javascript in behat tests
- freezing behat drupal extension to 1.0.2
- dkan_sitewide modules now report to github

DKAN_Datastore:

- Fix services module reference at dkan_datastore.make file

DKAN Dataset:

- Creating taxonomies during setup test
- Updating ref_field patch
- Replacing dkan_dataset_api with open_data_schema_map

8.2.66 7.x-1.5 2014-10-15

- Drupal core security update. Upgrades to 7.32

8.2.67 7.x-1.4 2014-10-10

- Replaced tid hardcoding during profile install with proper taxonomy_vocabulary_machine_name_load
- Removed duplicates in dkan.make already present in dkan_dataset.make
- Footer branding updated
- Editable dkan blocks
- Sort on search
- New rebuild script to keep dkan updated on custom installs

DKAN_Datastore:

- Many warnings fixed with proper isset calls
- Added dkan_datastore_api_count implementation
- Optimisations on GET parameters processing

DKAN Dataset:

- Many warnings fixed with proper isset calls
- Changes in make file (fixed versions on some components)
- Additional Info block on dataset page
- Added docx as allowed format for field_upload

8.2.68 7.x-1.3 2014-09-25

- Issue #192 drupal security update

8.2.69 7.x-1.2 2014-07-18

- Issue #184 drupal security update
- Updating makefile back to 7.x-1.x state
- – 1.1 Release commit + changing git urls with https urls in order to travis to not fail during make
- Changing git protocol to https for dkan_dataset, dkan_datastore and nuboot in order for travis build to success
- Issue #166 updating nuboot for iframe link theme
- Update dataset.inc fixed #150
- Update README.md

- Updating nuams references to NuCivic
- Updating nuams references to NuCivic
- Updating nuams references to Nucivic
- Updating nuams references to NuCivic
- Updating nuams references to NuCivic
- Updating drupal-org.make to point to Nucivic
- Updating drupal-org-dev.make to point to NuCivic
- Moving to using URLs for make file instead of copying locally
- Fixes issue #112: "downloald" typo in drupal-org.make
- Changed 'The goal of the project combine...' to 'The goal of the project is to combine...'
- Update .travis.yml
- Adding information about google group
- Adding key to drupal core to play nice with buildmanager
- adding multi channel notifications for dkan
- adding slack integration for travis
- Update to make files

NuBoot Radix Theme:

- Switch to radix as a base theme
- Updates for panels layouts
- Switch to flat icons for files http://www.flaticon.com/packs/file-formats-icons

8.2.70 7.x-1.1 2013-06-27

- Moved dkan development to Nucivic's github: https://github.com/NuCivic/dkan
- Grabbing dkan_dataset.make and dkan_datastore.make fiels from github cdn
- Several bugs fixed.
- Moving to using URLs for make file instead of copying locally #160
- Issue nuams/dkan#112: "downloald" typo in drupal-org.make #159
- Fixing #151: Admin role defaults to "Editor" #158
- Fix typo on About page #156
- Using hook_post_features_revert() for setting user_admin_role. #145
- Issue nuams/dkan#138 update recline in dkan_dataset.make to fix negative lat/lon #135
- Update to make files #131
- POD changes #128
- Issue #108 Add default email address for sample content #126
- Fix license block #123
- Issue #111 Fix facetapi_pretty_paths makefile definition #97

- Update drupal-org.make #84

DKAN_Datastore:

- Moved dkan_datastore development to NuCivic: https://github.com/NuCivic/dkan_datastore
- DKAN_Datastore: Several bugs fixed.

DKAN Dataset:

- Change references to NuCivic organization on github
- Several bug fixes
- Some changes for Project Open Data compliance, including data.json self-reference

nüBoot Theme:

- Styles for resource download button
- Change nuams references to NuCivic to reflect branding changes
- Left floating resource list
- Improve colorizer functionality

8.2.71 7.x-1.0 2014-04-29

- Moved dkan development to github: https://github.com/nuams/dkan
- Moved dkan_dataset development to github: https://github.com/nuams/dkan_dataset
- Moved dkan_datastore development to github: https://github.com/nuams/dkan_datastore
- Added BDD test suite for dkan: https://github.com/nuams/dkan/tree/7.x-1.x/test
- Added WebTestCase test suite for dkan_dataset: https://github.com/nuams/dkan_dataset/tree/7.x-1.x/tests
- Added WebTestCase test suite for dkan_dataset_api: https://github.com/nuams/dkan_dataset/tree/7.x-1.x/modules/dkan_dataset_api/tests
- Added WebTestCase test suite for dkan_datastore and dkan_datastore_api: https://github.com/nuams/dkan_datastore/tree/7.x-1.x/tests
- Travis + Github integration for all test suites on every commit
- Moved issues for all three modules exclusively to dkan's github issues: https://github.com/nuams/dkan/issues/
- Completed CKAN API Read Compliance with the following API methods: package_list, package_show, site_read, package_revision_list, group_list, resource_show, group_package_show and revision_list
- Added nuboot theme (https://drupal.org/project/nuboot) as default dkan theme
- Completed 508 compliance on dkan default theme
- Several bugs fixed.
- DKAN_Datastore: Moved dkan_datastore development to github: https://github.com/nuams/dkan_datastore
- DKAN_Datastore: Added WebTestCase test suite for dkan_datastore and dkan_datastore_api: https://github.com/nuams/dkan_datastore/tree/7.x-1.x/tests
- DKAN_Datastore: Travis + Github integration for all test suites on every commit
- DKAN_Datastore: Several bugs fixed.
- Added nüBoot theme as the main out-of-box theme for DKAN distro.

8.2.72 7.x-1.0-beta 2013-09-30

- First tagged release. Further tags will include a changelog.

8.3 Roadmap

Coming soon!

CHAPTER 9

License

DKAN is licensed on the same terms as Drupal, under GPLv2 or later. If you have any questions about the license a good place to start is to look at the Drupal Licensing FAQ.

The DKAN license also covers the related modules such as recline, open data scheme map, visualization entity feeds flat processor, and the taxonomy features.

CHAPTER 10

Additional resources

- DKAN Home: Central portal for the DKAN community.
- DKAN Starter Documentation: Implementation and deployment tools
- Drupal Documentation
- CKAN: Open-source data portal that is the inspiration for DKAN