datapusher Documentation

Release 1.0

Open Knowledge International

October 26, 2018

Contents

This application is a service that adds automatic CSV/Excel file loading to CKAN.

You should have a CKAN instance with the DataStore installed before using this. Head to the CKAN documentation for information on how to install CKAN and set up the DataStore.

CHAPTER 1

Development installation

Install the required packages:

```
sudo apt-get install python-dev python-virtualenv build-essential libxslt1-dev libxml2-dev zlib1g-dev
```

Get the code:

```
git clone https://github.com/ckan/datapusher
cd datapusher
```

Install the dependencies:

```
pip install -r requirements.txt
pip install -e .
```

Run the DataPusher:

```
python datapusher/main.py deployment/datapusher_settings.py
```

By default DataPusher should be running at the following port:

> http://localhost:8800/

If you need to change the host or port, copy *deployment/datapusher_settings.py* to *deployment/datapusher_local_settings.py* and modify the file.

CHAPTER 2

Production installation and Setup

Download and Install (All CKAN Versions)

Note: Starting from CKAN 2.2, if you installed CKAN via a package install, the DataPusher has already been installed and deployed for you. You can skip directly to *CKAN Configuration*.

This assumes you already have CKAN installed on this server in the default location described in the CKAN install documentation (`/usr/lib/ckan/default`). If this is correct you should be able to run the following commands directly, if not you will need to adapt the previous path to your needs.

These instructions set up the DataPusher webservice on Apache running on port 8800.

```
#install requirements for the DataPusher
sudo apt-get install python-dev python-virtualenv build-essential libxslt1-dev libxml2-

#create a virtualenv for datapusher
sudo virtualenv /usr/lib/ckan/datapusher

#create a source directory and switch to it
sudo mkdir /usr/lib/ckan/datapusher/src
cd /usr/lib/ckan/datapusher/src

#clone the source (this should target the latest tagged version)
sudo git clone -b 0.0.14 https://github.com/ckan/datapusher.git

#install the DataPusher and its requirements
cd datapusher
sudo /usr/lib/ckan/datapusher/bin/pip install -r requirements.txt
sudo /usr/lib/ckan/datapusher/bin/python setup.py develop

#copy the standard Apache config file
# (use deployment/datapusher.apache2-4.conf if you are running under Apache 2.4)
sudo cp deployment/datapusher.conf /etc/apache2/sites-available/datapusher.conf

#copy the standard DataPusher wsgi file
#(see note below if you are not using the default CKAN install location)
sudo cp deployment/datapusher.wsgi /etc/ckan/

#copy the standard DataPusher settings.
sudo cp deployment/datapusher_settings.py /etc/ckan/
```

```
#open up port 8800 on Apache where the DataPusher accepts connections.
#make sure you only run these 2 functions once otherwise you will need
#to manually edit /etc/apache2/ports.conf.
sudo sh -c `echo ``NameVirtualHost *:8800'' >> /etc/apache2/ports.conf'
sudo sh -c `echo ``Listen 8800'' >> /etc/apache2/ports.conf'

#enable DataPusher Apache site
sudo a2ensite datapusher
```

Note: If you are installing the DataPusher on a different location than the default one you need to adapt the following line in the `datapusher.wsgi` file to point to the virtualenv you are using:

```
activate_this = os.path.join('/usr/lib/ckan/datapusher/bin/activate_this.py')
```

Deployment with Gunicorn

The *wsgi.py* file provided lets you run datapusher with gunicorn if required. You will need to run gunicorn under supervisor and configure Nginx or Apache to proxy requests to gunicorn.

```
#install requirements for the DataPusher
sudo apt-get install python-dev python-virtualenv build-essential libxslt1-dev libxml2-

#create a virtualenv for datapusher
sudo virtualenv /usr/lib/ckan/datapusher

#create a source directory and switch to it
sudo mkdir /usr/lib/ckan/datapusher/src
cd /usr/lib/ckan/datapusher/src

#clone the source (this should target the latest tagged version)
sudo git clone -b 0.0.14 https://github.com/ckan/datapusher.git

#install the DataPusher and its requirements
cd datapusher
sudo /usr/lib/ckan/datapusher/bin/pip install -r requirements.txt
sudo /usr/lib/ckan/datapusher/bin/python setup.py develop

#install gunicorn
pip install gunicorn

#run datapusher with gunicorn
JOB_CONFIG='/usr/lib/ckan/datapusher/src/datapusher/deployment/datapusher_settings.py'
```

CKAN Configuration

In order to tell CKAN where this webservice is located, the following must be added to the `[app:main]` section of your CKAN configuration file (generally located at `/etc/ckan/default/production.ini`):

```
ckan.datapusher.url = http://0.0.0.0:8800/
```

The DataPusher also requires the `ckan.site_url` configuration option to be set on your configuration file:

```
ckan.site_url = http://your.ckan.instance.com
```

CKAN 2.2 and above

If you are using at least CKAN 2.2, you just need to add `datapusher` to the plugins in your CKAN configuration file:

```
ckan.plugins = <other plugins> datapusher
```

Restart apache:

```
sudo service apache2 restart
```

CKAN 2.1

If you are using CKAN 2.1, the logic for interacting with the DataPusher is located in a separate extension, ckanext-datapusherext.

To install it, follow the following steps

```
#go to the ckan source directory
cd /usr/lib/ckan/default/src

#clone the DataPusher CKAN extension
sudo git clone https://github.com/ckan/ckanext-datapusherext.git

#install datapusherext
cd ckanext-datapusherext
sudo /usr/lib/ckan/default/bin/python setup.py develop
```

Add `datapusherext` to the plugins line in `/etc/ckan/default/production.ini`:

```
ckan.plugins = <other plugins> datapusherext
```

Restart apache:

```
sudo service apache2 restart
```

CHAPTER 4

Using the DataPusher

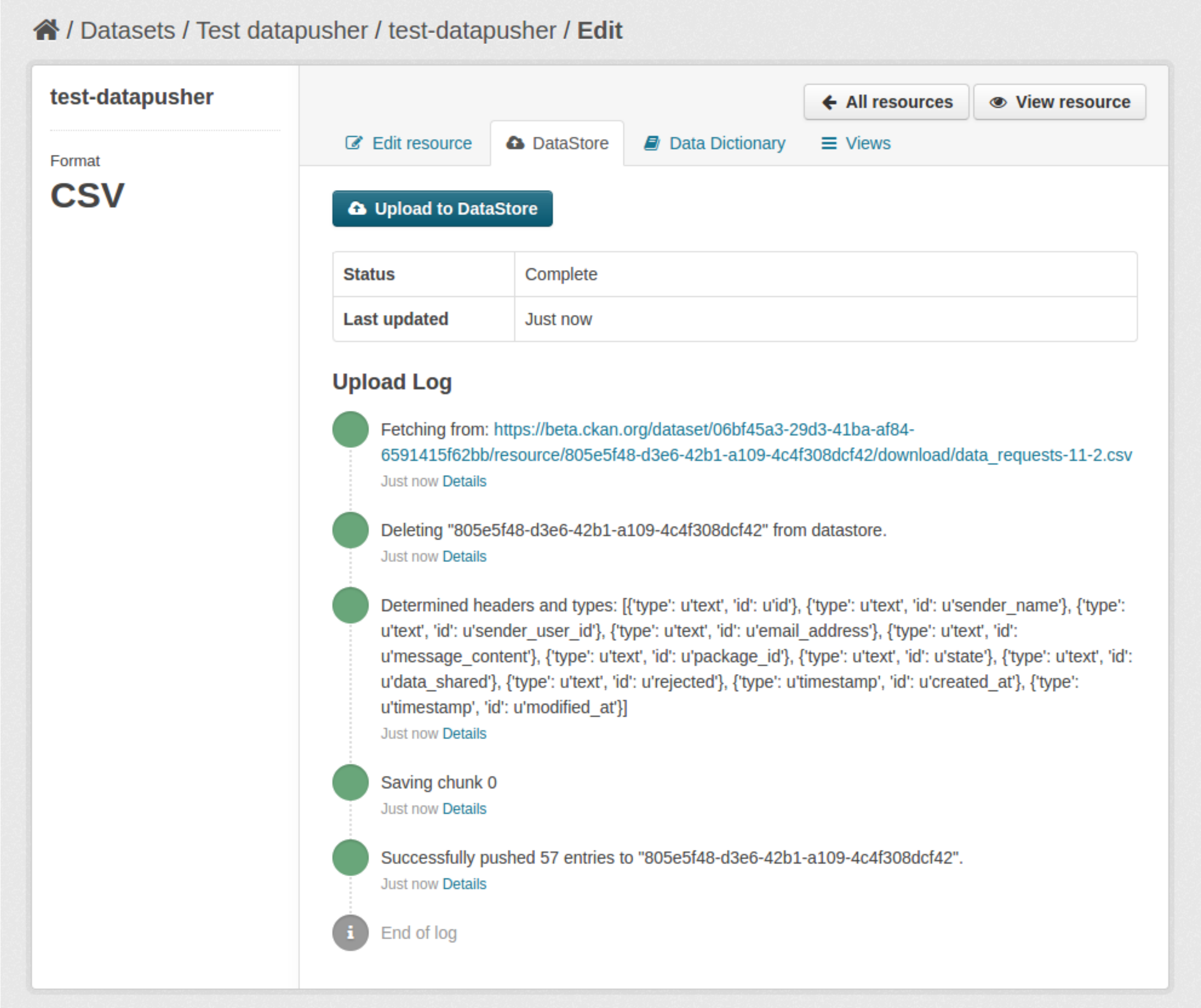
The DataPusher will work without any more configuration as long as the `datapusher` (or `datapusherext` for version 2.1) plugin is installed and added to the ckan config file.

Triggering pushes

Any file that has a format of csv or xls will be attempted to be loaded into to DataStore. This is triggered when a new URL is added to a dataset (resource). You can also manually trigger resources to be resubmitted.

CKAN 2.2 and above

When editing a resource in CKAN (clicking the “Manage” button on a resource page), a new tab will appear named “DataStore”. This will contain a log of the last attempted upload and an opportunity to retry to upload.



Status	Complete
Last updated	Just now

Upload Log

```
Fetching from: https://beta.ckan.org/dataset/06bf45a3-29d3-41ba-af84-
6591415f62bb/resource/805e5f48-d3e6-42b1-a109-4c4f308dcf42/download/data_requests-11-2.csv
Just now Details

Deleting "805e5f48-d3e6-42b1-a109-4c4f308dcf42" from datastore.
Just now Details

Determined headers and types: [{'type': u'text', 'id': u'id'}, {'type': u'text', 'id': u'sender_name'}, {'type':
u'text', 'id': u'sender_user_id'}, {'type': u'text', 'id': u'email_address'}, {'type': u'text', 'id':
u'message_content'}, {'type': u'text', 'id': u'package_id'}, {'type': u'text', 'id': u'state'}, {'type': u'text', 'id':
u'data_shared'}, {'type': u'text', 'id': u'rejected'}, {'type': u'timestamp', 'id': u'created_at'}, {'type':
u'timestamp', 'id': u'modified_at'}]
Just now Details

Saving chunk 0
Just now Details

Successfully pushed 57 entries to "805e5f48-d3e6-42b1-a109-4c4f308dcf42".
Just now Details

End of log
```

CKAN 2.1

If you want to retry an upload go into the resource edit form in CKAN and just click the "Update" button to resubmit the resource metadata. This will retrigger an upload.

From the command-line

Resubmit all resources to datapusher, although it will skip files whose hash of the data file has not changed:

```
paster --plugin=ckan datapusher resubmit -c /etc/ckan/default/ckan.ini
```

Resubmit a specific resource, whether or not the hash of the data file has changed:

```
paster --plugin=ckan datapusher submit <pkgname> -c /etc/ckan/default/ckan.ini
```

Configuring the maximum upload size

By default the datapusher will only attempt to process files less than 10Mb in size. To change this value you can specify the MAX_CONTENT_LENGTH setting in datapusher_settings.py

> MAX_CONTENT_LENGTH = 1024 # 1Kb maximum size

Configuring the guessing of types

The datapusher uses Messytables in order to infer data types. A default configuration is provided which is sufficient in many cases. Depending on your data however, you may need to implement your own Messytables types.

You can specify the types to use with the following settings in your datapusher_settings.py:

```
TYPES = [messytables.StringType, messytables.DecimalType, YourCustomType...]
TYPE_MAPPING = {'String': 'text', 'Decimal': 'numeric', 'YourCustom': 'timestamp'... }
```

Configuring SSL verification

By default datapusher will verify that requests to CKAN and other servers with HTTPS are with a valid SSL/TLS certificate. However the default list of root certificates is usually held by the operating system, and often gets out of date, causing SSL verification errors. (Browsers usually have their own list and update it frequently.)

The suggested fix is to use the latest version of requests and its ‘security’ addition:

```
pip install -U requests[security]
```

There are no known compatibility issues with CKAN or common extensions by using a more recent version of requests. (However requirements.txt still pins the version, as per the ckan policy.)

If you still have problems verifying certificates, or maybe for test purposes, you can switch the verification off in datapusher_settings.py:

```
SSL_VERIFY = False
```

CHAPTER 5

Debugging

Test the configuration

To test if it is DataPusher service is working or not run:

```
curl 0.0.0.0:8800
```

The result should look something like:

```
{
"help": "\n         Get help at:\n         http://ckan-service-provider.readthedocs.org/."
}
```

Error and logs

If there are any issues you should look in `/var/log/apache2/datapusher.error.log`. All log output will be put in there.

Debugging Gunicorn

Gunicorn doesn't print error logs to the console by default. Use the option *–log-file*=- to print logs to the console for debugging.

CHAPTER 6

License