OCDS Kingfisher Tool

Jun 12, 2019

Contents

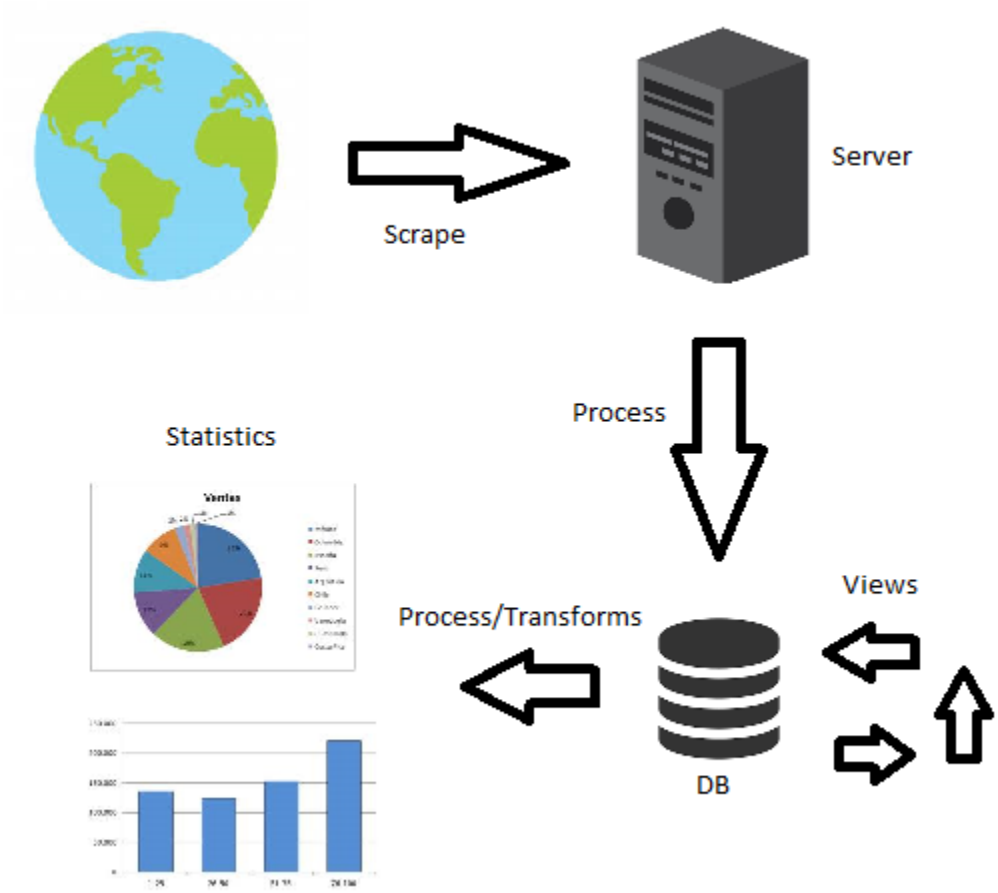
OCDS Kingfisher is a suite of tools for downloading, storing and analysing data from publishers of the Open Contracting Data Standard.

CHAPTER 1

Overview

Kingfisher comprises three tools, which integrate to form an OCDS data pipeline:

- kingfisher-scrape , for downloading data from public sources of OCDS data
- kingfisher-process, for storing OCDS data, and performing operations on it, including transformation and validation.
- kingfisher-views, A suite of queries that can be used to meaningfully interpret OCDS data sets



In the model you can see “Kingfisher Scrape” retrieves the data and stores it in a server, then “Kingfisher Process” puts them in the database in JSON format and also makes transformations, and “Kingfisher views” transform the jsons into their tabulated format to finally use them for generate the graphics.

CHAPTER 2

Hosted Kingfisher

Kingfisher has been developed primarily for internal use by Open Contracting Partnership, and OCP maintains a hosted instance. If you work or OCP or are part of the OCDS Team, see *Hosted Kingfisher* for details of the server, and how to access it.

CHAPTER 3

Previous Versions

In Febrary 2019, Kingfisher was completely rewritten. You can find details of previous versions on the *Old OCDS Kingfisher tool* page.

CHAPTER 4

Typical Uses of Kingfisher in the OCDS Helpdesk

4.1 I'm helping a publisher who's just published for the first time understand the quality of their data

First, write a scraper for the new source, or ask one of the developers to do it. Once it's merged into master on kingfisher-scrape, deploy the updated code to the server (using salt), and then log into the server and run the new scraper After a few minutes, check on the status of the scraper. Once it's started downloading, you should be able to open a connection to the Postgres database and see the data coming in.

CHAPTER 5

Typical Uses of Kingfisher outside of the OCDS Helpdesk

5.1 I'm a journalist interested in my country's OCDS data, and I want to be able to download the whole data set

First, check that kingfisher-scrape has a scraper for your country. If not, email data@open-contracting.org to let us know about the data. Then, install kingfisher-scrape, and run it to download the data. If you already have your own database, you can import the downloaded files to it. Or, if you'd prefer, you can use kingfisher-process to store the data in a Postgres database directly from the scraper.

5.2 I'm helping a government publish OCDS data, and I want to check that the published data is correct

If the data is already online, then use kingfisher-scrape to download it from the public portal. Email data@open-contracting.org if you require assistance with this. If the data isn't online, but you have it as JSON files on disk, you can use kingfisher-process to import it into a Postgres database, and then use the provided queries to start to understand the data.

CHAPTER 6

Other Information

6.1 Hosted Kingfisher

Open Contracting Partnership operates a hosted instance of the Kingfisher tool suite for the use of OCP staff and OCDS Team members. If you think you should have access to this, contact mailto:code@opendataservices.coop.

6.1.1 Access

You can connect to the server via SSH to run commands, or via a Postgres client to run queries on the database. Details of this are below.

From time to time, we create development servers to try things out before deploying them. If you're involved in the development of new Kingfisher components, you may be asked to log into the dev server to try things out. If so, substitiute the address of the dev server you've been provided with for the address of the live server in the examples below.

Over the course of a typical use of Hosted Kingfisher, you'll need to log in to run scrapers, then log out and back in as a different user to run process operations, and potentially connect to Postgres to carry out database operations.

6.1.2 Access for kingfisher-scrape

If you're running scrapers, SSH in as the *ocdskfs* user:

```
ssh ocdskfs@scrape.kingfisher.open-contracting.org
```

Once logged in, you can run scrapers as per the kingfisher-scrape documentation

6.1.3 Access for kingfisher-process

If you're running process operations, SSH in as the *ocdskfp* user:

```
ssh ocdskfp@process.kingfisher.open-contracting.org
```

Once logged in, you can run process operations as per the kingfisher-process documentation

You can browse the information in the web UI.

6.1.4 Access for analysis

If you're running analysis operations, SSH in as the *analysis* user:

```
ssh analysis@analyse.kingfisher.open-contracting.org
```

Once logged in, you can take advantage of the powerful server to carry out analysis operations, such as using flatten-tool on files, more quickly than on your local machine. The analysis user has read-only access to the files downloaded by the scrapers. Please remember to delete your files when you're done!

6.1.5 Access to the archives

There is an archive server which contains files that have been downloaded previously but are no longer held on the main server. In some cases, the data from them is still in the process database, but they are retained for reference. If you would like to access them, SSH in as the *archive* user:

```
ssh archive@archive.kingfisher.open-contracting.org
```

6.1.6 Access for Postgres Database queries

In order to access the database, you'll need a Postgres client such as *pgadmin <https://www.pgadmin.org/>*, and some details that are stored on the server.

To obtain the details, SSH into the server as the *ocdskfp* user (as above), and then run::: cat ~/.pgpass

The output contains two lines, for the two different database users that are available. It is recommended that you use the read-only user unless you're making changes to the database. The lines have the format:

localhost:*port*:*database*:*username*:*password*

These should be all the details you need to connect with a Postgres client.

6.1.7 Access for Redash

A Redash server is available. Contact Open Data Services for access.

6.1.8 Access for development

If you're developing new scrapers or working on any of the code for Kingfisher, then you may access a development server in order to try things out. The development server is low-spec, so intensive operations may fail. Don't rely on this server address, as it may change, or additional servers may be provided for your use. Please ensure that any development is co-ordinated with Open Data Services Co-Op.

The development server is at:

```
ocdskingfisher-dev.default.opendataservices.uk0.bigv.io
```

The development server is largely configured in the same way as the production server, so logins and commands should still work in the same way.

To see the scrapyd web interface, visit: http://scrape.ocdskingfisher-dev.default.opendataservices.uk0.bigv.io/ The username and password can be supplied by Open Data Services Co-op on request.

To see the Process app web interface, visit: http://process.ocdskingfisher-dev.default.opendataservices.uk0.bigv.io/app

6.2 Old OCDS Kingfisher tool

An older version of this tool is still available.

For more see:

- https://github.com/open-contracting/kingfisher/tree/archive
- https://ocdskingfisher.readthedocs.io/en/archive/

6.3 Vagrant for developers

A Vagrant box is included in this repository. This is one box that includes Scrape and Process for any development work.

6.3.1 Requirements

You will need:

- Vagrant https://www.vagrantup.com/
- VirtualBox https://www.virtualbox.org/

(It may be possible to get this to work with other Virtulisation solutions, but VirtualBox is the one we have tested with. Also check your virtualization options to make the system work, if you are already virtualizing your server you may have problems with starting Vagrant)

6.3.2 Setting up repositories

To use it, check out the main repository somewhere on your disk.

```
git clone git@github.com:open-contracting/kingfisher.git somewhere_you_decide
```

You then need to check out the repositories with the actual code. Note these must be within the location you just created - so change to that folder. Also use the names specified here, as the Vagrantfile will refer to these.

```
cd somewhere_you_decide
git clone git@github.com:open-contracting/kingfisher-scrape.git scrape
git clone git@github.com:open-contracting/kingfisher-process.git process
git clone git@github.com:open-contracting/kingfisher-views.git views
```

6.3.3 Starting and building the Vagrant box

Simply type

```
vagrant up
```

6.3.4 Opening (and exiting) a shell in the Vagrant box

Simply type

```
vagrant ssh
```

To exit this shell, type *exit*.

6.3.5 Working with Scrape

You will find all the files inside the vagrant box in */scrape*

There is a virtual environment at *.ve*

So, after you have opened a shell inside the Vagrant box, try:

Simply type

```
cd /scrape
source .ve/bin/activate
```

You can run standalone Scrapy commands straight away - see https://kingfisher-scrape.readthedocs.io/en/latest/use-standalone.html

6.3.6 Working with Process

You will find all the files inside the vagrant box in */process*

There is a virtual environment at *.ve*

So, after you have opened a shell inside the Vagrant box, try:

```
cd /process
source .ve/bin/activate
```

You can access the database by simply typing *db*.

There is a test database - to run tests in try:

```
KINGFISHER_PROCESS_DB_URI=postgresql://test:test@localhost:5432/test pytest␣
↪tests/
```

To run the app in debug mode on port 9090, try:

```
FLASK_APP=ocdskingfisherprocess.web.app FLASK_ENV=development KINGFISHER_
↪PROCESS_WEB_API_KEYS=cat flask run --host 0 --port 9090
```

When this is running, you should be able to see results in http://localhost:9090/app

You can generate a detailed description of the database Schema with SchemaSpy:

```
java -jar /bin/schemaspy.jar -t pgsql -dp /bin/postgresql.jar   -s public  -
↪db ocdskingfisher  -u ocdskingfisher -p ocdskingfisher -host localhost -o /
↪vagrant/schemaspy
```

6.3.7 Working with Views

You will find all the files inside the vagrant box in */views*

There is a virtual environment at *.ve*

More information will follow soon. TODO

6.3.8 Guide: Running a scraper and seeing it appear in the database

You will need two shells open.

In the first one, we are going to run the process app:

```
cd /process
source .ve/bin/activate
python ocdskingfisher-process-cli upgrade-database
FLASK_APP=ocdskingfisherprocess.web.app FLASK_ENV=development KINGFISHER_
↪PROCESS_WEB_API_KEYS=cat flask run --host 0 --port 9090
```

Leave that running.

Open a second shell and run:

```
cd /scrape
source .ve/bin/activate
source env.sh
scrapy crawl canada_buyandsell -a sample=true
```

Log messages will appear in the shell. While this is happening, you can

- Open a third shell, type *db* and see the data appear in the database
- Open a webbrowser, and see the data appear in http://localhost:9090/app

6.3.9 Finishing work with the Vagrant Box

Simply type

```
vagrant halt
```

6.3.10 If you break the Vagrant Box

If you have tried to change the config of the software, tried to install something else and it's all gone horribly wrong ....

That's totally fine!

The whole point is there should be no data you care about inside the Vagrant box, and thus you should feel free to destroy it and recreate it at any time.

```
vagrant destroy
vagrant up
```

6.3.11 Removing totally the Vagrant Box

Simply type

```
vagrant destroy
```

6.4 The Archive Server

6.4.1 Purpose

The archive server stores old data files that have been downloaded by scrape.

It does this so that if we need to load old files for historical analysis later, we can use the files on the archive server and do so by the local load function in process.

6.4.2 Installation on the Scrape and Process server

Currently the script needs to be on one server that has both scrape and process parts. (This is because it needs to directly access both the database and the source files)

The script is at https://github.com/open-contracting/kingfisher-archive

The user account that runs it needs

- ssh access to the archive server
- access to the database (a .pgpass file)
- sudo permission to delete files from the scrape account

More than one instance of the script should not run at once. This is because they may clash, and try and archive the same collection at the same time.

To ensure this, the script exits after it has found and archived one collection.

The script is started once per day by a cron job.

Output can be piped to a log file, for debugging purposes. On hosted Kingfisher, these are in /home/archive/logs/

6.4.3 Installation on the Archive server

There is no software part to install on this side.

Simply make sure the archive account

- can be accessed over SSH
- has a /home/archive/data/ folder