django-calaccess-campaign-finance Documentation

Release 0.1.1

California Civic Data Coalition

January 26, 2016

Contents

1 Documentation **3**

1.1 How to use it 3

1.2 Management commands 5

1.3 Models 10

1.4 How to contribute 10

1.5 Exporting the data 11

1.6 Importing to Microsoft SQL Server 11

1.7 Changelog 13

2 Open-source resources **15**

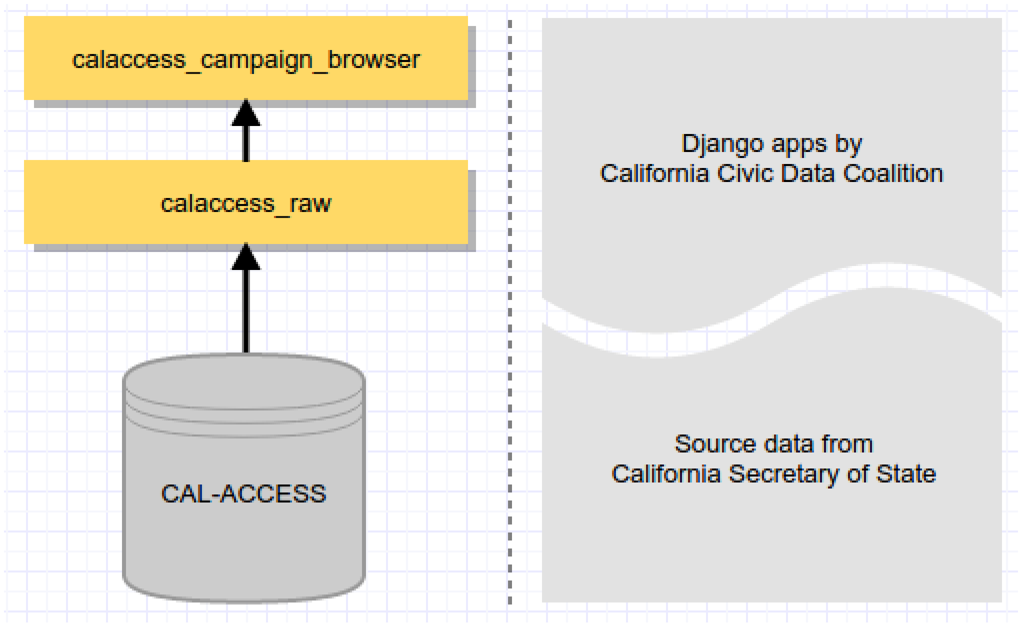
3 Read more **17**

4 Events **19**

5 Sponsors **21**

A Django app to refine and investigate campaign finance data drawn from the California Secretary of State's CAL-ACCESS database.

Intended as a second layer atop django-calaccess-raw-data that transforms the source data and loads it into simplified models that serve as a platform for investigative analysis.


calaccess_campaign_browser
calaccess_raw
CAL-ACCESS
Django apps by California Civic Data Coalition
Source data from California Secretary of State

Warning: This is a work in progress. Its analysis should be considered as provisional until it is further tested and debugged.

Documentation

1.1 How to use it

This guide will walk users through the process of installing the latest official release from the Python Package Index. If you want to install the raw source code or contribute as a developer refer to the "How to contribute" page.

Warning: This library is intended to be plugged into a project created with the Django web framework. Before you can begin, you'll need to have one up and running. If you don't know how, check out the official Django documentation.

1.1.1 Installing the Django app

The latest official release of the application can be installed from the Python Package Index using `pip`.

```
$ pip install django-calaccess-campaign-browser
```

Add this `calaccess_campaign_browser` app as well as the underlying `calaccess_raw` app that contains the raw government database to your Django project's `settings.py` file.

```
INSTALLED_APPS = (
    ...
    'calaccess_raw',
    'calaccess_campaign_browser',
)
```

1.1.2 Connecting to a local database

Also in the `settings.py` file, you will need to configure Django so it can connect to a database.

Unlike a typical Django project, this application only supports the MySQL database backend. This is because we enlist specialized tools to load the immense amount of source data more quickly than Django typically allows. We haven't worked out those routines for PostgreSQL, SQLite and the other Django backends yet, but we're working on it.

Preparing MySQL

Before you begin, make sure you have a MySQL server installed. If you don't, now is the time to hit Google and figure out how. If you're using Apple's OSX operating system, you can install via Homebrew. Here is the official MySQL documentation for how to get it done.

Once that is handled, create a new database named `calaccess`.

```
$ mysqladmin -h localhost -u your-username-here -p create calaccess
```

Also in the Django `settings.py`, configure a database connection.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'calaccess',
        'USER': 'your-username-here',
        'PASSWORD': 'your-password-here',
        'HOST': 'localhost',
        'PORT': '3306',
        # You'll need this to use our data loading tricks
        'OPTIONS': {
            'local_infile': 1,
        }
    }
}
```

1.1.3 Loading the data

Now you're ready to create the database tables with Django using its `manage.py` utility belt.

```
$ python manage.py migrate
```

Once everything is set up, this management command will download the latest bulk data release from the state and load it in the database. It'll take a while. Go grab some coffee.

```
$ python manage.py downloadcalaccessrawdata
```

Next, run the management command that extracts and refines campaign finance data from from the raw CAL-ACCESS data dump.

```
$ python manage.py buildcalaccesscampaignbrowser
```

1.1.4 Exploring the data

In your project `urls.py` file, add this app's URLs:

```
urlpatterns = patterns('',
    url(r'^', include('calaccess_campaign_browser.urls')),
)
```

Finally start the development server and visit localhost:8000 in your browser to start using the app.

```
$ python manage.py runserver
```

1.2 Management commands

1.2.1 Loaders

buildcalaccesscampaignbrowser

The master command that runs all of the other commands below necessary to completely update the database.

```
Usage: example/manage.py buildcalaccesscampaignbrowser [options]

Transforms and loads refined data from raw CAL-ACCESS source files

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

loadcalaccesscampaigncontributions

```
Usage: example/manage.py loadcalaccesscampaigncontributions [options]

Load refined campaign contributions from CAL-ACCESS raw data

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

loadcalaccesscampaignexpenditures

```
Usage: example/manage.py loadcalaccesscampaignexpenditures [options]
```

```
Load refined campaign expenditures from CAL-ACCESS raw data

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

loadcalaccesscampaignfilers

```
Usage: example/manage.py loadcalaccesscampaignfilers [options]

Load refined CAL-ACCESS campaign filers and committees

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

loadcalaccesscampaignfilings

```
Usage: example/manage.py loadcalaccesscampaignfilings [options]

Load refined CAL-ACCESS campaign filings

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
```

```
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --flush               Flush table before loading data
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

loadcalaccesscampaignsummaries

```
Usage: example/manage.py loadcalaccesscampaignsummaries [options]

Load refined CAL-ACCESS campaign filing summaries

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

1.2.2 Exporters

exportcalaccesscampaignbrowser

```
Usage: example/manage.py exportcalaccesscampaignbrowser [options]

Export refined CAL-ACCESS campaign browser data as CSV files

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --skip-contributions  Skip contributions export
  --skip-expenditures   Skip expenditures export
  --skip-summary        Skip summary export
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

1.2.3 Scrapers

scrapecalaccesscampaigncandidates

```
Usage: example/manage.py scrapecalaccesscampaigncandidates [options]

Scrape links between filers and elections from the CAL-ACCESS site

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

scrapecalaccesscampaignpropositions

```
Usage: example/manage.py scrapecalaccesscampaignpropositions [options]

Scrape links between filers and propositions from the CAL-ACCESS site

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

scrapecalaccesscampaignelectiondates

```
Usage: example/manage.py scrapecalaccesscampaignelectiondates [options]
```

```
Scrape election dates from the Secretary of State's site

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

1.2.4 Other

dropcalaccesscampaignbrowser

```
Usage: example/manage.py dropcalaccesscampaignbrowser [options]

Drops all CAL-ACCESS campaign browser database tables

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

flushcalaccesscampaignbrowser

```
Usage: example/manage.py flushcalaccesscampaignbrowser [options]

Flush CAL-ACCESS campaign browser database tables

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output,
                        2=verbose output, 3=very verbose output
  --settings=SETTINGS   The Python path to a settings module, e.g.
                        "myproject.settings.main". If this isn't provided, the
                        DJANGO_SETTINGS_MODULE environment variable will be
                        used.
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g.
                        "/home/djangoprojects/myproject".
  --traceback           Raise on exception
  --no-color            Don't colorize the command output.
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

1.3 Models

coming soon

1.4 How to contribute

This walkthrough will show you how to install the source code of this application to fix bugs and develop new features.

First create a new virtualenv.

```
$ virtualenv django-calaccess-campaign-browser
```

Jump in.

```
$ cd django-calaccess-campaign-browser
$ . bin/activate
```

Clone the repository from GitHub.

```
$ git clone https://github.com/california-civic-data-coalition/django-calaccess-campaign-browser.git
```

1.4.1 Installing the database

Move into the repository and install the Python dependencies.

```
$ cd repo
$ pip install -r requirements_dev.txt
```

Make sure you have MySQL installed. If you don't, now is the time to hit Google and figure out how. If you're using Apple's OSX operating system, you can install via Homebrew. If you need to clean up after a previous MySQL installation, this might help.

Then create a new database named `calaccess`.

```
$ mysqladmin -h localhost -u root -p create calaccess
```

If you have a different username, substitute it above. You'll be prompted for that user's mysql password.

Create a file at `example/project/settings_local.py` to save your custom database credentials. That might look something like this.

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'calaccess',
        'USER': 'your-username-here',
        'PASSWORD': 'your-password-here',
        'HOST': 'localhost',
        'PORT': '3306',
        'OPTIONS': {
            'local_infile': 1,
        }
    }
}
```

Finally create your database and get to work.

```
$ python example/manage.py migrate
```

1.4.2 Loading the data

You might start by loading the data dump from the web.

```
$ python example/manage.py downloadcalaccessrawdata
```

Then you can build the campaign finance models

```
$ python example/manage.py buildcalaccesscampaignbrowser
```

And fire up the Django test server to use the browser

```
$ python example/manage.py collectstatic
$ python example/manage.py runserver
```

1.5 Exporting the data

This walkthrough will show you how to export the database tables for the `Contribution` and `Expenditure` models as CSV files.

This assumes that you have installed the project and ran the `buildcalaccesscampaignbrowser` command.

```
$ python manage.py buildcalaccesscampaignbrowser
```

From here, you can now run the export command to get a dump of the data from the database.

```
$ python manage.py exportcalaccesscampaignbrowser
```

This will export the CSVs as `YYYY-MM-DD-model.csv` to `os.path.join(settings.BASE_DIR, 'data')`

1.6 Importing to Microsoft SQL Server

This walkthough will show you how to import the exported CSV data to Microsoft SQL Server. This guide assumes you already have SQL server installed and running.

First build the browser and export the tables:

```
$ python manage.py buildcalaccesscampaignbrowser
$ python manage.py exportcalaccesscampaignbrowser
```

1.6.1 Setting up pypyodbc

In order to connect to SQL Server from the application, we'll need to install pypyodbc, an *open database connectivity* library that allows Python to communicate with multiple databases.

Installation is simple:

```
$ pip install pypyodbc
```

Configure ODBC and Free TDS

You'll need to install ODBC and configure its database drivers.

Mac OS X users can install ODBC and Free TDS with homebrew:

'Note:' This has not been tested and should be considered provisional

```
$ brew install unixodbc
$ brew install freetds
```

Linux users can follow these instructions are from the project wiki:

1. Install ODBC and Free TDS

```
$ sudo apt-get install tdsodbc unixodbc
```

2. Modify `/etc/odbcinst.ini`

From the wiki: "If `odbcinst.ini` doesn't exist under /etc, create the file. Find the path to `libtdsodbc.so`. If the path to the file is `/usr/lib/x86_64-linux-gnu/libtdsodbc.so`, make sure you have the below content in the file:"

```
[FreeTDS]
Driver = /usr/lib/x86_64-linux-gnu/odbc/libtdsodbc.so
```

3. Modify /etc/freetds/freetds.conf

From the wiki: "Make sure there are following lines under the "Global" section in the file:"

```
[Global]
TDS_Version = 8.0
client charset = UTF-8
```

1.6.2 Connecting to the server

Next, you'll need to add the SQL server variables to your `settings.py`. Since this will contain sensitive information, we suggest storing these variables in a `local_settings.py` file that's not tracked by version control and importing it.

```
SQL_SERVER_DRIVER = ''  # Use 'FreeTDS' if you followed instructions above
SQL_SERVER_ADDRESS = ''  # Your SQL Server IP address
SQL_SERVER_PORT = ''  # Your SQL Server port number
SQL_SERVER_USER = ''  # Your SQL server username
SQL_SERVER_PASSWORD = ''  # Your SQL server password
SQL_SERVER_DATABASE = '' # Your SQL server database name
```

1.6.3 Import data into SQL Server

With the above configuration, you should now be able to import the exported CSV data from your local folder to SQL server:

```
$ python manage.py importtosqlserver
```

1.7 Changelog

1.7.1 0.1.1 (February 2015)

- Scrapers that collect elections and propositions

1.7.2 0.1.0 (2014)

- Initial release

Open-source resources

- Code: github.com/california-civic-data-coalition/django-calaccess-campaign-browser
- Issues: github.com/california-civic-data-coalition/django-calaccess-campaign-browser/issues
- Packaging: pypi.python.org/pypi/django-calaccess-campaign-browser
- Testing: travis-ci.org/california-civic-data-coalition/django-calaccess-campaign-browser
- Coverage: coveralls.io/r/california-civic-data-coalition/django-calaccess-campaign-browser

CHAPTER 3

Read more

- ‘CAL-ACCESS Dreaming’, the kickoff presentation from the initial code convening (Aug. 13, 2014)
- ‘Introducing the California Civic Data Coalition’, a blog post annoucing the first public code release (Sept. 24, 2014)
- ‘Package data like software, and the stories will flow like wine’, a polemic explaining this project’s software design philosophy (Sept. 24, 2014)
- ‘Light everywhere: The California Civic Data Coalition wants to make public datasets easier to crunch’, a story about the creators by Nieman Journalism Lab (Oct. 20, 2014)
- ‘Once a crusader against big money, Gov. Brown is collecting millions’, a Los Angeles Times story that utilized this application (Oct. 31, 2014)

CHAPTER 4

Events

Development has been advanced by a series of sprints supported by Knight-Mozilla OpenNews.

- August 13-15, 2014, at Mozilla's offices in San Francisco
- January 14-15, 2015, at USC's Wallis Annenberg Hall
- March 4-8, 2015, the California Code Rush at NICAR 2015 in Atlanta

Sponsors