django-calaccess-lobbying-activity Documentation
Release 0.2

California Civic Data Coalition

August 13, 2014

Contents

A simple Django app to build lobbying activity data from the cal access database. It is reliant on django-calaccess-parser

Documentation

1.1 Django Cal-Access Lobbying Activity

django-calacces-lobbying-activity is a simple Django app to lobbying activity data from the cal access database. It is reliant on django-calaccess-parser.

Detailed documentation is in the "docs" directory. *(coming soon)*

1.1.1 Requirements

- Django 1.6
- django.contrib.humanize
- MySQL 5.5
- django-calaccess-parser
- Patience

1.1.2 Installation

- Install django-calaccess-browser with pip

  ```
  $ pip install django-calaccess-lobbying-activity
  ```

- Add `lobbying` to your INSTALLED_APPS setting like this:

  ```
  INSTALLED_APPS = (
      ...
      'lobbying',
  )
  ```

  In your project `urls.py`:

  ```
  ...
  urlpatterns = patterns('',
      url(r'^browser/', include('lobbying.urls')),
  )
  ```

- Next, sync the database, create a Django admin user, and run the management command to extract campaign finance data from from the raw calaccess data dump.

```
$ python manage.py syncdb
$ python manage.py build_lobbying
```

> **warning** This'll take a while. Go grab some coffee or do something

else productive with your life.

1.1.3 Building

The JavaScript and CSS for the project is managed with Grunt and Bower. Currently, the JavaScript and SCSS dependencies are not included so you'll need to build them yourself;

1. Install Node.js (this will also include NPM)

2. Install Grunt and bower globally with `npm install -g bower grunt-cli`

3. Go to the `static` directory and install the required dependencies with `npm install && bower install`

4. Generate the `main.css` file and watch for HTML, CSS and JavaScript changes by executing `grunt`

1.1.4 Explore data

Start the development server and visit http://127.0.0.1:8000/browser/ to inspect the Cal-access data.

1.1.5 API

django-calaccess-browser uses django-tastypie to expose the data as an API. Add `tastypie` to the project `INSTALLED_APPS` and make sure you included `lobbying.urls` in your project's `urls.py`.

From there visit 127.0.0.1:8000/browser/api/v1/filer/?format=json to explore the JSON representation of the data.

1.1.6 Export

You can also export the data into easily queryable flat files.

```
$ python manage.py export_lobbying
```

1.1.7 Update the data

When you are ready to get new data, just blow away both the parser and the campaign finance browser app. Then reload them.

You can do something like this, making sure you fill in your database and user names correctly. `bash $ python manage.py sqlclear calaccess | mysql -u user_name -p database_name $ python manage.py sqlclear lobbying | mysql -u user_name -p database_name $ pytnon manage.py syncdb $ python manage.py downloadcalaccess && python manage.py build_lobbying`

1.1.8 Authors

- Agustin Armendariz
- Ben Welsh
- Aaron Williams

1.2 Management commands

1.2.1 build_lobbying

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-lobbying-activity
```

Jump in.

```
$ cd django-calaccess-lobbying-activity
$ . bin/activate
```

Clone the repository from GitHub.

Make sure you have the django-calaccess-parser installed otherwise NONE OF THIS WILL WORK.

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

Move into it 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 `lobbying`.

```
mysqladmin -h localhost -u root -p create lobbying
```

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

Then 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': 'lobbying',
        'USER': 'yourusername', # <-- This
        'PASSWORD': 'yourpassword', # <-- And this
        'HOST': 'localhost',
        'PORT': '3306',
        'OPTIONS': {
            'local_infile': 1,
        }
    }
}
```

Finally create your database and get to work.

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

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

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

Then you can build the lobbying activity models

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

Other resources

- Documentation: http://django-calaccess-lobbying-activity.rtfd.org

- Issues: https://github.com/california-civic-data-coalition/django-calaccess-lobbying-activity/issues

- Packaging: https://pypi.python.org/pypi/django-calaccess-lobbying-activity

- Testing: https://travis-ci.org/california-civic-data-coalition/django-calaccess-lobbying-activity

- Coverage: https://coveralls.io/r/california-civic-data-coalition/django-calaccess-lobbying-activity