État Civil Documentation

Release '0.5.0'

King's Digital Lab

Sep 01, 2020

CONTENTS

Table of Contents:

CHAPTER

ONE

ÉTAT CIVIL

For the period 1790-1890, France was the only country to keep systematic (tabular) track of their expatriate citizens making it possible to study international mobility at scale and in novel ways. The Archives of the French Ministry of Foreign Affairs hold 120,000 digitised microfilm images of the records from 215 consulates around the world.

The "État Civil" – standing for civil registration of births, deaths and marriages – is an exploratory project led by Dr David Todd in collaboration with King's Digital Lab supported by the Faculty of Arts & Humanities, the Department of History at King's College London and the Harvard and Cambridge Centre for History and Economics. The project processes data from the Egyptian consulate of the "État Civil" to visualise mobility on a continental or global scale and offer insights on patterns of migration and social history more in general.

1.1 Settings

See detailed cookiecutter-django settings documentation.

1.2 Running

1.2.1 Getting Started on a Mac

Install Developer Tools, in a Terminal window:

```
$ xcode-select --install
```

Install and start Docker.

Clone the repository, https://github.com/kingsdigitallab/etat-civil-django, in a Terminal window, for example:

```
$ cd Documents
$ git clone https://github.com/kingsdigitallab/etat-civil-django.git
$ cd etat-civil-django
```

Start the project:

```
$ ./bake.py up --build
```

Create a superuser, in a different Terminal window, inside the Etat Civil project:

```
$ ./bake.py manage createsuperuser
```

To import data via the browser the data processing worker needs to be running, start it with:

```
$ ./bake.py rqworker default
```

The project should be available at http://localhost:8000/. Go to http://localhost:8000/admin/deeds/data/ to import a new Excel file. The import process might take a few minutes to import all the data in the spreadsheet.

See also the more detailed cookiecutter-django development with Docker documentation.

1.3 Basic Commands

1.3.1 Setting Up Your Users

- To create a **normal user account**, just go to Sign Up and fill out the form. Once you submit it, you'll see a "Verify Your E-mail Address" page. Go to your console to see a simulated email verification message. Copy the link into your browser. Now the user's email should be verified and ready to go.
- To create an **superuser account**, use this command:

  ```
  $ ./bake.py manage createsuperuser
  ```

For convenience, you can keep your normal user logged in on Chrome and your superuser logged in on Firefox (or similar), so that you can see how the site behaves for both kinds of users.

1.3.2 Type checks

Running type checks with mypy:

```
$ mypy {{cookiecutter.project_slug}}
```

1.3.3 Test coverage

To run the tests, check your test coverage, and generate an HTML coverage report:

```
$ coverage run -m pytest
$ coverage html
$ open htmlcov/index.html
```

1.3.4 Running tests with py.test

```
$ pytest
```

CHAPTER

TWO

TECHNICAL OVERVIEW

2.1 Team

```
Organisation: King's Digital Lab
Site: https://kdl.kcl.ac.uk
Email: kdl-info [at] kcl.ac.uk
Twitter: @kingsdigitallab
GitHub: https://github.com/kingsdigitallab
Location: London WC2B 5LE, United Kingdom

Research Software Analyst: Arianna Ciula
Site: https://www.kdl.kcl.ac.uk/who-we-are/arianna-ciula/

Research Software Engineer: Miguel Vieira
Site: https://www.kdl.kcl.ac.uk/who-we-are/miguel-vieira/
```

2.2 Technologies and Processes

2.2.1 Development

For more information see development and development with docker.

2.2.2 Data model

The data model graph was generated with the django-extensions graph_models command:

```
$ ./bake.py manage graph_models deeds -X TimeStampedModel  --disable-fields --disable-
↪abstract-fields -o models.png
```

Below is a very simplified example of how the model is used to record information on one deed (État Civil Ismailia 1872-1882).

- Deed:
  - ID: 1; acttype: 1; n: 21; date: 24 déc. 1872; place: 1; source: 1
- Person:
  - ID: JMS; name: Joseph Marius Silvy; year of birth: 1842
  - ID: TB; name: Thérèse Blaum; year of birth: 1848

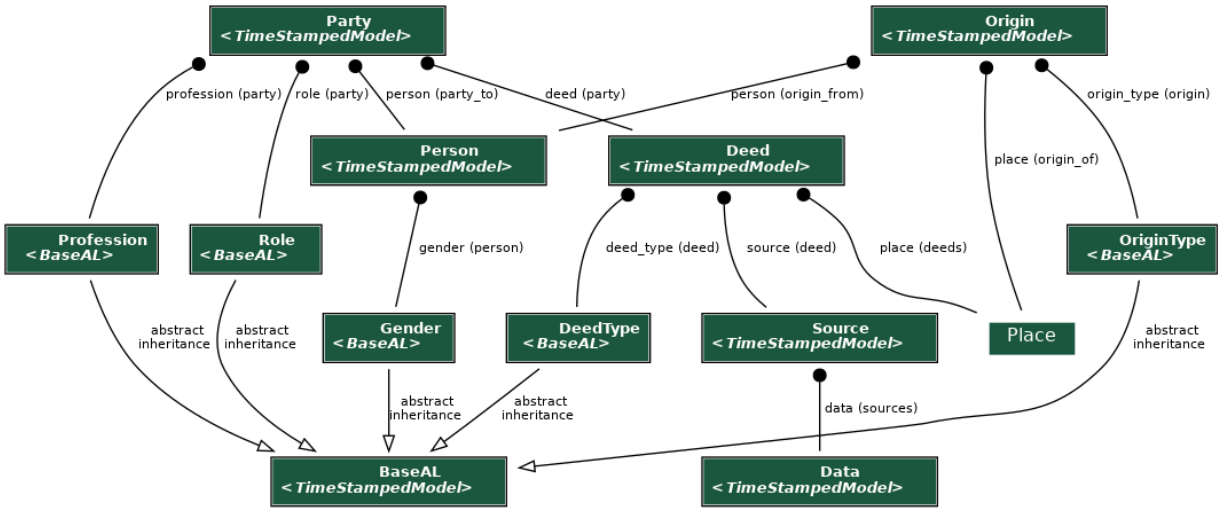


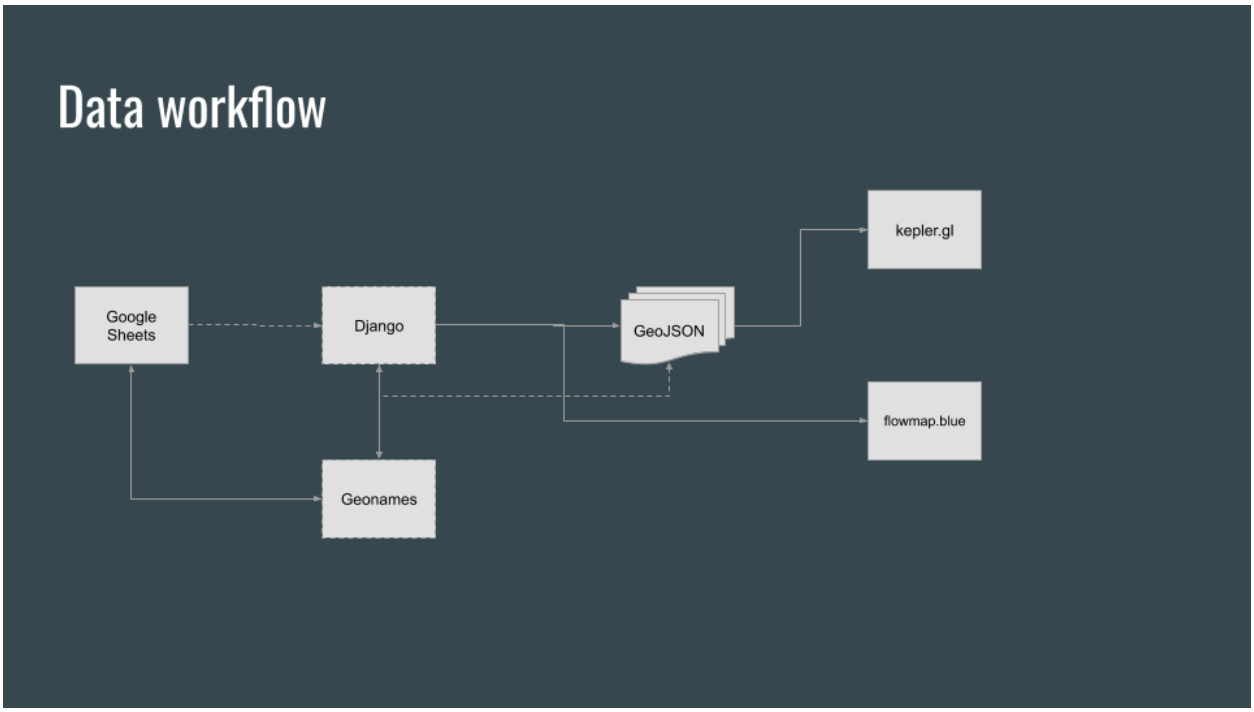
Fig. 1: Django models

- Party:
  - Deed: 1; person: JMS; role: 2; profession: 2
  - Deed: 1; person: TB; role: 1; profession: 1
- Origin:
  - Person: JMS; place: 2; originype: 1
  - Person: TB; place: 3; origintype: 2
- DeedType:
  - 1: Birth
  - 2: Death
  - 3: Marriage
- Place:
  - 1: Ismailia; Egypt
  - 2: Département de la Haute-Garonne; Saint-Marcel; France
  - 3: Département des Pyrénées-Orientales; Collioure; France
- OriginType:
  - 1: Birth
  - 2: Origin
- Profession: - 1: Sans profession - 2: Mécanicien
- Role:
  - 1: Mother
  - 2: Father
- Source:
  - 1: Classmark: Ismaïlia 1; microfilm: P 07070

- 2: Classmark: Le Caire 1; microfilm: P 06505

2.2.3 Workflows

The Django app imports data from a spreadsheet (exported from the project Google Sheet), adds geographic locations to places, and provides an admin/editing interface to manage the data (with simple filtering). After the curation and cleaning up is done in Django admin, the applicaition can export data into GeoJSON and CSV formats to support map visualisations.



2.2.4 Architecture

KDL built the resource using Django, an Open Source web publishing framework with which KDL has extensive experience of and has found to be stable, powerful and scalable. The Django based web platform provided functionality to upload the project dataset. With respect to the Proof of Concept, data was exported from the Django database in GeoJSON and CSV formats so as to generate maps visualisations in the tools mentioned below. The Django based web platform and related tools provides the functionality to record metadata about the archival sources under examination via an administrative interface to the resource.

To support advanced queries on the project Django database (e.g. in checking, processing, analysing and visualising the data), KDL set up Metabase (a free, stand-alone Java application that can connect to various types of databases, build queries via a user interface, browse, filter and export the results, make simple charts and visualisation and share them as cards into collections) which can be used without advanced coding knowledge (apart from SQL, if needed) and includes a REST API to export JSON/CSV results from the cards. However, this additional application has not been used as yet to analyse the data systematically.

The graphs were generated by the docker-compose-viz tool:

```
$ docker run --rm -it --name dcv -v $(pwd):/input pmsipilot/docker-compose-viz render␣
↪-m image local.yml
```

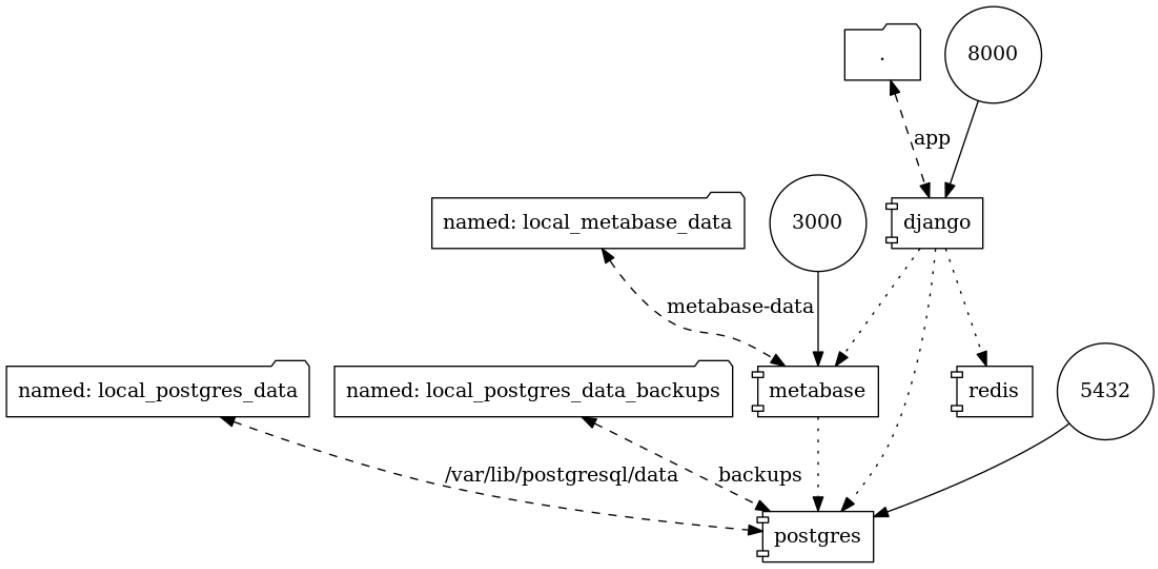


Fig. 2: Local Docker Stack

2.3 Design process

The project makes uses of two existing tools to generate map visualisations:

- Kepler.gl is an open source geospatial analysis tool for large-scale data sets
- flowmap is a more minimal solution to create geographic flow maps from data published (not private) in Google Sheets. It allows to visualize numbers of movements between locations (origin-destination data) and explore the data interactively; however on a limited number of parameters (from, to and count). See the flowmap version based on the État Civil data.

Description of this proof of concept with historical introduction are available on the Harvard and Cambridge Centre for History and Economics Visualizing Historical Networks project website.

CHAPTER

THREE

CHANGELOG

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

3.1 [Unreleased] - yyyy-mm-dd

3.1.1 Added

- Kepler map.

3.2 [0.5.0] - 2020-07-02

3.2.1 Added

- Text under data model, workflow, architecture and design process.
- Image under workflow.

3.3 [0.4.1] - 2020-05-13

3.3.1 Fixed

- Export to GeoJSON.

3.4 [0.4.0] - 2020-04-20

3.4.1 Added

- Latest version of the data

3.4.2 Changed

- Use the place names from the data collection spreasheet rather than the geonames names

3.4.3 Fixed

- Recover deleted test data
- Increase gunicorn timeout for data exports

3.5 [0.3.1] - 2020-04-06

3.5.1 Fixed

- Add missing dependencies for production
- Database set up on production

3.6 [0.3.0] - 2020-04-06

3.6.1 Added

- LDAP configuration for production

3.6.2 Changed

- Production settings for deployment

3.7 [0.2.0] - 2020-03-27

3.7.1 Added

- Team information to the docs
- humans.txt (http://humanstxt.org/)
- Docker script to stop the containers
- Data to source control

3.8 [0.1.0] - 2020-02-13

3.8.1 Added

- Data model and data collection template
- Django admin interface
- Commands to import/export data into CSV and GeoJSON
- Geocoding workflow
- Map visualisations (wip)

CHAPTER

FOUR

INDICES & TABLES

- genindex
- modindex
- search