Waves Resource Center

Release 0.4.0

Harison Gachuru

May 11, 2021

CONTENTS

CHAPTER

ONE

INSTALLATION

1.1 Required Software

1. Git
2. Python
3. Firefox web browser
4. Geckodriver
5. MySQL or PostgreSQL

1.2 Local Setup

1. Clone the repository
2. Set the following **environment variables**:
   - `DATABASE_URL`
   - `DJANGO_EMAIL_HOST_USER` - a gmail account
   - `DJANGO_EMAIL_HOST_PASSWORD` - password to the gmail account (prefarably an app password)
3. Create and activate a virtual environment using `pipenv` by running `$ pipenv shell`
4. Install development dependencies by running `$ pipenv install --dev`
5. Run the tests using `$ python manage.py test --settings=resource_center.settings.test`.
   - Make sure you have `geckodriver` installed and in your `PATH` before attempting to run the test. Read selenium python docs for more information on how to do this.
   - You could also add this flag `--exclude-tag=functional` to run unit tests only.

CHAPTER TWO

DEPLOYMENT

2.1 Google Cloud Environment Setup

1. Create a Google Cloud project
   - `GCP_PROJECT_ID`
2. Create a Cloud Storage bucket
   - `GS_BUCKET_NAME` - Google Cloud Storage bucket name
3. Create a Cloud SQL MySQL 2nd generation instance
   - Note the `DATABASE_INSTANCE_CONNECTION_NAME`
4. Create a database user
5. Create a database
6. Create 2 service accounts, create keys for them and save them in your local machine:
   - `GOOGLE_APPLICATION_CREDENTIALS` - a json file containing credentials for a Google Cloud service account with the following roles:
     - Storage Object Creator
     - Storage Object Viewer
   - `APP_ENGINE_DEPLOYER_SERVICE_ACCOUNT_FILE`- a json file containing credentials for a Google Cloud service account with the following roles:
     - App Engine Deployer
     - App Engine Service Admin
     - Cloud Build Editor
     - Storage Object Creator
     - Storage Object Viewer
7. Create an App Engine app

2.2 App Engine Deployment

1. Set the required **environment variables**
2. Run `./scripts/deploy_to_app_engine.sh` in a Linux terminal.
    - Use Git bash or WSL if using Windows OS.

2.3 Notes

- You need the Google Cloud SDK installed on your machine.
- App Engine currently doesn't support `Pipfile`. Instead of doing the deployment manually, we recommend you use the utility script for deployment: `deploy_to_app_engine.sh` stored in the `scripts` directory. It does set up operations before deployment and clean up after deployment.

CHAPTER

THREE

RESOURCE-CENTER

3.1 accounts package

3.1.1 Submodules

3.1.2 accounts.admin module

3.1.3 accounts.apps module

3.1.4 accounts.forms module

3.1.5 accounts.managers module

3.1.6 accounts.models module

3.1.7 accounts.urls module

3.1.8 accounts.views module

3.1.9 Module contents

3.2 books package

3.2.1 Submodules

3.2.2 books.admin module

3.2.3 books.apps module

3.2.4 books.models module

3.2.5 books.urls module

3.2.6 books.views module

3.2.7 Module contents

3.3 config package

3.3.1 Subpackages

config.settings package

Submodules

config.settings.base module

config.settings.local module

config.settings.production module

config.settings.test module

Module contents

3.3.2 Submodules

3.3.3 config.asgi module

3.3.4 config.urls module

3.3.5 config.wsgi module

3.3.6 Module contents

3.4 core package

3.4.1 Submodules

3.4.2 core.admin module

3.4.3 core.apps module

3.4.4 core.models module

3.4.5 core.urls module

3.4.6 core.views module

3.4.7 Module contents

3.5 utils package

3.5.1 Submodules

3.5.2 utils.config module

`utils.config.`**`list_of_tuples`**`(`*str*`)`

3.5.3 utils.storages module

3.5.4 utils.test module

3.5.5 Module contents

3.6 videos package

3.6.1 Submodules

3.6.2 videos.admin module

3.6.3 videos.apps module

3.6.4 videos.models module

3.6.5 videos.urls module

3.6.6 videos.views module

3.6.7 Module contents

CHAPTER

FOUR

GOOGLE CLOUD COMMAND REFERENCE

- Create a project: `$ gcloud projects create [PROJECT_ID] --name=[PROJECT_NAME]`
- Create/set a billing account for the project
  - Only done via Cloud Shell
- Create a service account:

```
$ gcloud iam service-accounts create [SERVICE_ACCOUNT_ID] \
  > --description="DESCRIPTION" \
  > --display-name="DISPLAY_NAME"
```

- Add an IAM policy to a service account:

```
$ gcloud projects add-iam-policy-binding [PROJECT_ID] \
  > --member="serviceAccount:SERVICE_ACCOUNT_ID@PROJECT_ID.iam.gserviceaccount.com
↪" \
  > --role="ROLE_NAME"
```

- List all service accounts: `$ gcloud iam service-accounts list`
- List all Google Cloud regions: `$ gcloud compute regions list`
- Set a default region/zone for the project: `$ gcloud config set compute/region [REGION]`
- Enable the Cloud Storage service: `$ gcloud services enable storage-component.googleapis.com`
- Create a bucket: `$ gsutil mb gs://[BUCKET_NAME]`
- Create a Cloud SQL instance:

```
$ gcloud sql instances create [INSTANCE_NAME] \
  > --database-version [DATABASE_VERSION] --region=[REGION] --tier=[TIER] \
  > --backup-start-time=[BACKUP_START_TIME] \
  > --storage-auto-increase
```

- Enable the SQL Admin API (to use the Cloud SQL proxy): `$ gcloud services enable sqladmin.googleapis.com`
- List App Engine regions: `$ gcloud app regions list`
- Create an app: `$ gcloud app create --region=[REGION]`
- Enable the App Engine Admin API: `$ gcloud services enable appengine.googleapis.com`
- Enable the Cloud Datastore API: `$ gcloud services enable datastore.googleapis.com`

CHAPTER

FIVE

DOCUMENTATION

5.1 Building Documentation

1. Activate the development virtual environment
2. Run `$ docs/make clean` to remove any docs previously built
3. Run `$ docs/make html` to build the docs in HTML format
4. Change the current directory to the location of the built docs by running `$ cd docs/_build/html`
5. Start the Python static files server by running `$ python -m http.server`
6. Visit `localhost:8000` in your browser to view the docs

5.2 Generating the Documentation

1. Activate the development virtual environment
2. Generate the docs by running `$ sphinx-apidoc -f -o docs/_sources . main.py manage.py *migrations* *tests*`

CHAPTER SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX