Language Support (/categories/language-support) › Python (/categories/python-support) › Working w...

Configuring Django Apps for Heroku

Last updated 18 September 2018

The basics

First, and most importantly, Heroku web applications require a `Procfile` .

This file is used to explicitly declare your application's process types and entry points. It is located in the root of your repository.

Procfile

```
web: gunicorn myproject.wsgi
```

This Procfile requires Gunicorn, the production web server that we recommend for Django applications. For more information, see Deploying Python Applications with Gunicorn (https://devcenter.heroku.com/articles/python-gunicorn).

Installing gunicorn

```
$ pip install gunicorn
```

Be sure to add `gunicorn` to your `requirements.txt` file as well.

settings.py changes

On Heroku, sensitive credentials are stored in the environment as config vars (https://devcenter.heroku.com/articles/config-vars). This includes database connection information (named `DATABASE_URL`), which is traditionally hardcoded in Django applications.

The django-heroku (https://github.com/heroku/django-heroku) package automatically configures your Django application to work on Heroku. It is compatible with **Django 2.0** applications.

It provides many niceties, including the reading of `DATABASE_URL` , logging configuration, a Heroku CI (https://devcenter.heroku.com/articles/heroku-ci)–compatible `TestRunner` , and automatically configures 'staticfiles' to "just work".

Installing django-heroku:

```
$ pip install django-heroku
```

Be sure to add `django-heroku` to your `requirements.txt` file as well.

Add the following `import` statement to the top of `settings.py` :

```
import django_heroku
```

Then add the following to the bottom of `settings.py` :

```
# Activate Django-Heroku.
django_heroku.settings(locals())
```

Deploy, and you should be good to go!