Thursday: Deploying to Heroku

Deploying To Heroku.

Let us now learn how we can deploy our apps to Heroku.

Requirements

1. gunicorn.
2. Heroku account.
3. Heroku Command line Interface.

Step 1 . Gather all Requirements.

- Install **gunicorn** **(http://gunicorn.org/)**

This is a unix based server that will run our application on Heroku

```
(virtual)$ python3.6 -m  pip install gunicorn
```

- Create a free Heroku **account** **(https://moringacore-python.herokuapp.com/heroku.com)**

This will be our hosting environment for our application.

- Install the Heroku **CLI** **(https://devcenter.heroku.com/articles/heroku-cli)**

Step 2. List all the Python dependencies.

We need to list all the dependencies required by the Heroku environment.

To do that we just type the command:

```
(virtual)$ pip freeze

click==6.7
dominate==2.3.1
Flask==0.12.2
Flask-Bootstrap==3.3.7.1
Flask-Script==2.0.5
Flask-WTF==0.14.2
gunicorn==19.7.1
itsdangerous==0.24
Jinja2==2.9.6
MarkupSafe==1.0
visitor==0.1.3
Werkzeug==0.12.2
WTForms==2.1
pkg-resources==0.0.0
```

We will put all those dependencies in a file called _requirements.txt_ that will be in our root folder.

```
(virtual)$ pip freeze > requirements.txt
```

Go into your _requirements.txt_ file and remove the `pkg-resources==0.0.0` dependency. This is a small bug that will prevent us from deploying our applications.

Step 3. Creating a Procfile

A Procfile is a mechanism for declaring what commands are run by the Heroku environment We will create a file in our root folder and name it `Procfile`

In our case we want to run our Flask application using gunicorn.

Procfile

```
web: gunicorn manage:app
```

Step 4. Create a new Heroku application

If you haven't done so, set up your Heroku account and follow the given steps.

```
(virtual)$ heroku login
```

Then let us create a new application

```
(virtual)$ heroku create <name-of-app>
```

Replace the `<name-of-app>` with what you want to call your application.

Step 5 Adding configurations

We can now add our environment variables as configuration variables to our Heroku project

```
(virtual)$ heroku config:set MOVIE_API_KEY=<YOUR MOVIE API>
(virtual)$ heroku config:set SECRET_KEY=<YOUR SECRET KEY>
```

This will enable us to access the environment configurations in our Heroku application.

Step 6. Deployment.

```
$ git add .
$ git commit -m "deployment to heroku"
$ git push heroku master
```

Heroku goes through your *requirements.txt* file,installs all our dependencies and builds our application.

Projects

Lost and Afraid

Create a text adventure game where a user is lost in a forest. The game would give the user a set of instructions and observations The user has to make decisions based on the options the game provides. You can also supplement the text with images.

Have the user create a profile before he starts playing. YOu can collect the user's name,their weapon of choice and 3 resources that they have.

Book Worm.

Using the **Google Books   (https://developers.google.com/books/docs/v1/getting_started)** API create an application that allows users to search for books and get information about the books.

Did you enjoy this content?

Yes   No

Send Feedback

Did you enjoy this content?