

Developing a Python Flask Web App and deploying with App Engine Flexible

Presented by Andrew Barnes

Google Account

Make sure you are signed up for the google account free trial.

Overview

What you will learn

- How to write a small flask app in Python.
- How to deploy a simple web application to the App Engine Flexible Environment

Python and Flask



Interpreted high-level Programming language.



A microframework for python based on Werkzeug and Jinja2

So, why use Python for this project?

The app engine standard environment has containers that are pre configured with Java 7 & 8, Python 2.7 & 3, Go, PHP, Node.js runtimes.

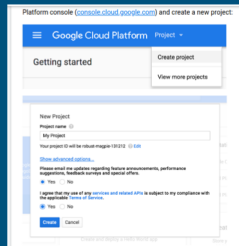
Python comes already loaded on the google cloud shell along with pip which is used for installing flask, and other packages

Let's make a new project to start

Once you have a new project, open Google Cloud shell and make a new directory.

`Mkdir flaskapp`

`Cd flaskpp`



Python 3 Virtual Environment

The cloud shell comes preloaded with virtual environments.

This allows us to setup a "fake" environment to have dependencies and packages that will not affect other projects we might later make.

`Virtualenv -p python3 env`

Enter the virtual environment with the command

`Source env/bin/activate`

Test Virtual environment

You can type `deactivate` to release from the virtual environment.

I suggest leaving the virtual environment and typing `python`.

`Ctrl d` allows you to leave python console

Then enter your virtual environment again with `source env/bin/activate`

Again type `python`.

See that the versions are different. Outside should be default 2.7 env is 3

Let's first make a requirements.txt file

Feel free to use whatever editor you feel comfortable with. I use Visual Studio Code

This file allows us to list the dependencies of our project. What and which version we want.

Right now we will just add Flask since that is our web framework.

Add `Flask==0.12.2`

Let's make the main.py file

There is a lot involved so I am gonna type out our file and do some explaining as I type. Please follow along and if you have questions feel free to ask.

Html files

We first will need to make a templates folder to house our html templates.
Follow along with coding the html files.

When you upload them to the console they will be in the root directory so you will have to move them to the templates folder we made.

Files done let's test locally

We can try running our project locally in google console. We want to make sure it is working properly before we really deploy it.

First we will need to install our dependencies so type

`pip install -r requirements.txt`

Then `python main.py` and now you should be able to test on port 8080

I will now walk around and double check with everyone if locally it is working.

Let's begin to deploy

You will want to first type `gcloud app create`

I believe the regional time zone we want will be # 3

With that created next we will need one more file a `app.yaml` file

Yaml files

Upon looking into this lab I found out that YAML stands for "Yet Another Markup Language".

It's meaning now is Yaml Ain't Markup Language to distinguish its purpose as data-orient, rather than document markup.

Follow along and I will type out our `yaml` file

I also found a nice tool website called `yamllint.com` which verifies if your YAML text works correctly and will clean it up as well.

Finishing up with gunicorn

We need to make one edit to our `requirements.txt`. We will add to it `gunicorn==19.7.1`.

Gunicorn standing for 'Green Unicorn' is a python WSGI HTTP for unix.

You can find more about it at `gunicorn.org`. A lot of examples I found used it.

Finally ready to deploy

When ready just type `gcloud app deploy`. This step might take a while.

I will be walking around to make sure everyone is has theirs working. I ran into some problems last night doing it and had to make sure I named everything exactly.

Once you have it running and can show it, you can stop the project.

Go to `app engine` in the side bar, near the bottom `settings`, and then `disable app`. This should prevent from being charged. If you want you can delete the project as well.

Additional quests or concerns?

Home work is to now look at and do the google code lab "Deploying a Python Flask Web Application to App Engine Flexible"

I want a picture of joy likelihood for a picture. And a picture of the app instance being deployed.

Take a look at the python code they implement and the google apis they use as well.

Resources

Qwik lab: App Engine: Qwik Start - Python
Code lab: Deploy a Python Flask Web Application to App Engine Flexible

<http://gunicorn.org/>

Flask quick start
<http://flask.pocoo.org/docs/0.12/quickstart/>
