Python Flask Quickstart

medium.com/netron/python-flask-quickstart-2cf258256619

Dheeraj Shah

September 23, 2020

Machine learning Medium Publication, Special Focus on Deep learning and Computer Vision. Soon gonna have the great series of END TO END implementation on Data Science and AI projects.

Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.

Before you start proceeding with this tutorial, I am assuming that you have hands-on experience on HTML and Python. If you are not well aware of these concepts, then I will suggest you to go through on short tutorials on HTML and Python.

Installation

Python Version

I'll be using the latest version of Python 3. Flask supports Python 3.5 and newer, Python 2.7, and PyPy.

Virtual environments

Use a virtual environment to manage the dependencies for your project, both in development and in production.

What problem does a virtual environment solve? The more Python projects you have, the more likely it is that you need to work with different versions of Python libraries, or even Python itself. Newer versions of libraries for one project can break compatibility in another project.

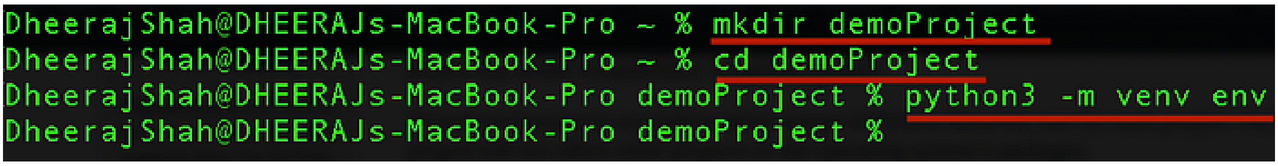
Virtual environments are independent groups of Python libraries, one for each project. Packages installed for one project will not affect other projects or the operating system's packages.

Python 3 comes bundled with the **venv** module to create virtual environments.

Create an environment

Create a project folder and a env folder within, here I have my project named as demoProject and env is just folder within in which installing of flask will be done.

On Mac and Linux:



```
DheerajShah@DHEERAJs-MacBook-Pro ~ % mkdir demoProject
DheerajShah@DHEERAJs-MacBook-Pro ~ % cd demoProject
DheerajShah@DHEERAJs-MacBook-Pro demoProject % python3 -m venv env
DheerajShah@DHEERAJs-MacBook-Pro demoProject %
```

On Windows:

```
$ mkdir demoProject$ cd demoProject$ py -3 -m venv env
```

Activate the environment

Before you work on your project, activate the corresponding environment:

```
$ . env/bin/activate
```

On Windows:

```
 venv\Scripts\activate
```

Your shell prompt will change to show the name of the activated environment as shown below.

Install Flask

Within the activated environment, use the following command to install Flask:

```
$ pip install Flask
```

Flask is now installed.

Create and run a minimal Flask app

In Sublime Text IDE , create a new file in your project folder named `app.py`

In `app.py`, add code to import Flask and create an instance of the Flask object. If you type the code below (instead of using copy-paste)

```
from flask import Flaskapp = Flask(__name__)
```

Also in `app.py`, add a function that returns content, in this case a simple string, and use Flask's `app.route` decorator to map the URL route `/` to that function:

```
@app.route("/")def home():    return "Hello, Flask!"
```

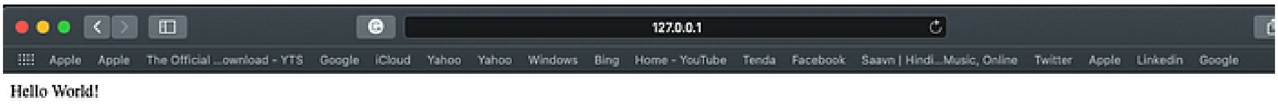
> Tip: You can use multiple decorators on the same function, one per line, depending on how many different routes you want to map to the same function.

Save the `app.py` file

In the terminal, run the app by entering `python3 -m flask run` (macOS/Linux) or `python -m flask run` (Windows), which runs the Flask development server. The development server looks for `app.py` by default. When you run Flask, you should see output similar to the following:

Also, if you want to run the development server on a different IP address or port, use the host and port command-line arguments, as with `--host=0.0.0.0 --port=80`.

To open your default browser to the rendered page, Ctrl+click the `http://127.0.0.1:5000/`URL in the terminal.



Hello World!

Another ways to run the server

1.In this way you need to tell your terminal the application to work with by exporting the `FLASK_APP` environment variable, But one problem can exist i.e when you change in the python file and reload the page you will not be able to see that change which you made while in presence of running server.

To see the change that you have to restart the server. Hence you will see the change.

```
export FLASH_APP=app.py#set FLASK_APP=app.py (for windows)flask run
```

2. In this way you will be able to make change in your files while running of your server.

```
export FLASH_DEBUG=1#set FLASK_DEBUG=1 (for windows)flask run
```

3. Add below lines in the your app.py file and run by the command`python app.py`.

```
if __name: == '__main__':  app.run(debug =True)
```

> This was the first part of the **Get your hands on Flask**.Whole series will be coming soon.

Thank You.

Reference

https://flask.palletsprojects.com/en/1.1.x/quickstart/

https://flask.palletsprojects.com/en/1.1.x/installation/#install-create-env