Flask: Getting Started

Flask is a small, light-weighted web framework. It provides you with the basic building blocks for a powerful Python web application.

We cannot teach you all of its functions, but you can always refer to their website and documentation for any related information.

- Flask official website: http://flask.pocoo.org/
- Flask documents: http://flask.pocoo.org/docs/1.0/

Installing

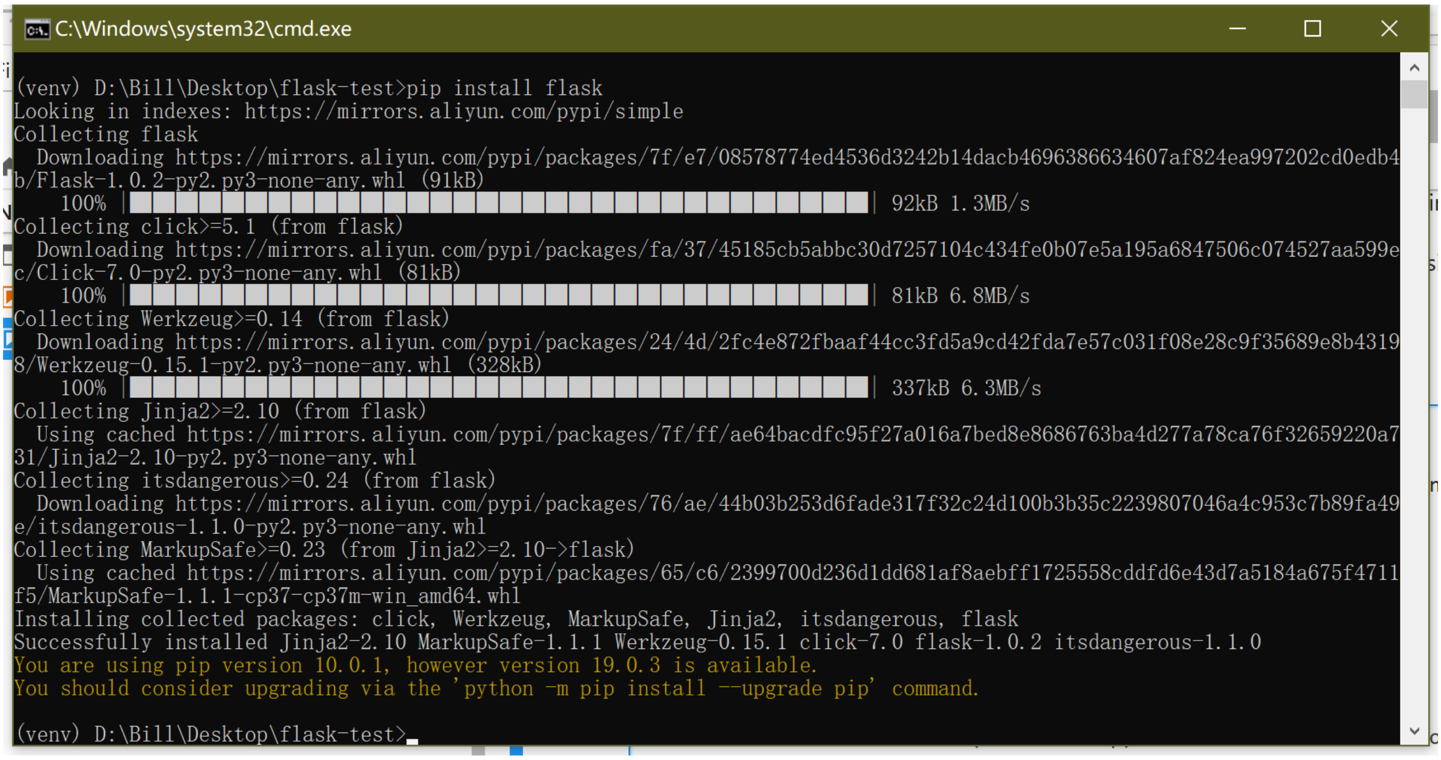
Make sure you have installed Python environment and Pypi package management system (normally come with your Python interpreter).

Install Flask

We will use `pip` package manager to install Flask. Run the following command in your command prompt or terminal.

```
pip install flask
```

You will see some output like,



```
(venv) D:\Bill\Desktop\flask-test>pip install flask
Looking in indexes: https://mirrors.aliyun.com/pypi/simple
Collecting flask
  Downloading https://mirrors.aliyun.com/pypi/packages/7f/e7/08578774ed4536d3242b14dacb4696386634607af824ea997202cd0edb4
b/Flask-1.0.2-py2.py3-none-any.whl (91kB)
    100% |████████████████████████████████| 92kB 1.3MB/s
Collecting click>=5.1 (from flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a6847506c074527aa599e
c/Click-7.0-py2.py3-none-any.whl (81kB)
    100% |████████████████████████████████| 81kB 6.8MB/s
Collecting Werkzeug>=0.14 (from flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/24/4d/2fc4e872fbaaf44cc3fd5a9cd42fda7e57c031f08e28c9f35689e8b4319
8/Werkzeug-0.15.1-py2.py3-none-any.whl (328kB)
    100% |████████████████████████████████| 337kB 6.3MB/s
Collecting Jinja2>=2.10 (from flask)
  Using cached https://mirrors.aliyun.com/pypi/packages/7f/ff/ae64bacdfc95f27a016a7bed8e8686763ba4d277a78ca76f32659220a7
31/Jinja2-2.10-py2.py3-none-any.whl
Collecting itsdangerous>=0.24 (from flask)
  Downloading https://mirrors.aliyun.com/pypi/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49
e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10->flask)
  Using cached https://mirrors.aliyun.com/pypi/packages/65/c6/2399700d236d1dd681af8aebff1725558cddfd6e43d7a5184a675f4711
f5/MarkupSafe-1.1.1-cp37-cp37m-win_amd64.whl
Installing collected packages: click, Werkzeug, MarkupSafe, Jinja2, itsdangerous, flask
Successfully installed Jinja2-2.10 MarkupSafe-1.1.1 Werkzeug-0.15.1 click-7.0 flask-1.0.2 itsdangerous-1.1.0
You are using pip version 10.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

(venv) D:\Bill\Desktop\flask-test>
```

If you can see the "Successfully installed Flask", you are ready to go.

The First Piece of Code

Now we can start to write the first piece of code.

On your desktop, create a folder call `flask`, then enter this folder. In this folder, create a file call `hello.py`, and type the following code.

```python
# hello.py

from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

The above code is the minimal but a complete application. Now please copy the above code to a new Python file in your project folder. For example, `hello.py`.

Then, set the environment variable in your command prompt or terminal. In this case, right click the blank area of the folder contains the file `hello.py`, click the `Open with Command Prompt` or `Open with Windows PowerShell` button in the context menu, type one of the following command accordingly on the command prompt of PowerShell window.

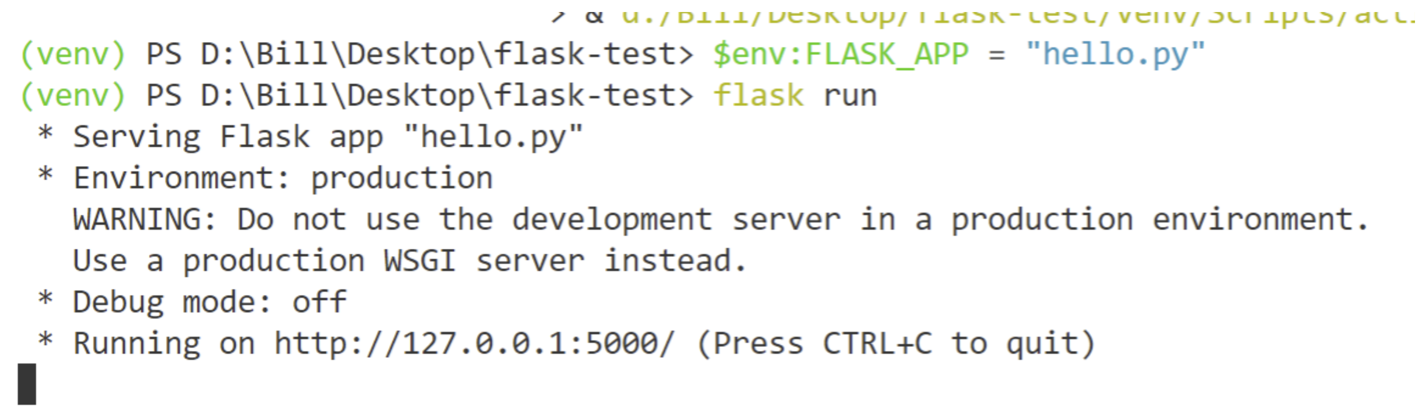
- Windows command prompt: `set FLASK_APP=hello.py`
- Windows PowerShell: `$env:FLASK_APP = "hello.py"`

- macOS/Linux: `export FLASK_APP=hello.py`

Finally, run your code with the following command,

```
flask run
```

if you can see the following output, your code is running on port `5000` now.



```
> & d:/Bill/Desktop/flask-test/venv/Scripts/act
(venv) PS D:\Bill\Desktop\flask-test> $env:FLASK_APP = "hello.py"
(venv) PS D:\Bill\Desktop\flask-test> flask run
 * Serving Flask app "hello.py"
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Open your browser, type `http://localhost:5000`, you should see the sentence you just typed, "Hello World".

Program Structure

This is the code you just ran.

```
# hello.py

from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

What does the code do?

- `from flask import Flask`: in this statement, a class `Flask` was imported.
- `app = Flask(__name__)`: at here, an instance of the class was created.
    - The argument here indicates the name of the module. If you are working on a multiple-module project, you should do your design a fit a suitable name. Otherwise, set this as `__name__` like the example here if you are writing a single-module project.
- `@app.route('/')`: this statement is a decorator. `route()` decorator indicates the URL that triggers this function.
- The `hello_world()` function: this function generates URLs for the particular function, and returns the message you just saw.

Your following Flask programs will look like this, but you will have more functions and routes defined.

Now you have seen a simple Flask application, then we will begin with an actual project to practice URL handling, templates, and database connection.

References

- http://flask.pocoo.org/docs/1.0/quickstart/#a-minimal-application