Guide

May 25, 2020

1 Flask Set Up Summary

Starting any Flask app will follow the same general flow for a simple application. The steps below are the same steps as taken in the screencast and can be referenced if you get stuck during the exercise.

1.1 Directory & Virtualenv Set Up

We have created the directory `intro_project`, along with the subdirectory `flaskr` and the `__init__.py` file. You can edit `__init__.py` in the editor to the right.

At this point you're ready to start working on your app!

1.2 Create_app

In our setup, we will configure the basic Flask app. In these notes, I'll also include how to set up the application to handle specialized configuration.

1.2.1 Basic App

1. Import your dependencies

- `from flask import Flask, jsonify`

2. Define the `create_app` function with parameter `test_config` initially set to `None`. Then within the function:
3. Define the application. Ensure you include the first parameter. `__name__`is the name of the current Python module.
   - `app = Flask(__name__)`
4. Return the app instance.
   - `return app`

1.3 First Endpoint with JSON

Before you return the app, use the @app.route decorator to create an endpoint to path / and define a function to handle that route.

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

return app "'

Instead of returning text, use jsonify to send an object containing the message

```
 return jsonify({'message':'Hello, World!'})
```

1.4 Run your application

In the command line, you'll run three lines of code. The first two lines tell the terminal where to find your application and to run it in development mode, which allows you to keep it running while it hotloads any modifications. The third actually starts the application.

```
export FLASK_APP=flaskr
export FLASK_ENV=development
flask run
```

After you start the application you'll see output like will look like the following:

```
 * Serving Flask app "flaskr" (lazy loading)
 * Environment: development
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 192-247-084
```

In the second terminal window copy and past the following com'mand: `{'message':'Hello, World!'}` displayed. You can try adding more endpoints or changing the response before moving forward.