# How to Run a Flask Application

Last Updated : 21 Aug, 2024

The backend server Flask was created fully in Python. It is a framework made up of Python modules and packages. With its characteristics, it is a lightweight Flask application that speeds up the development of backend apps. We will learn how to execute a Flask application in this tutorial.

## Run Flask application Syntax

We can run the Flask application using the below command.

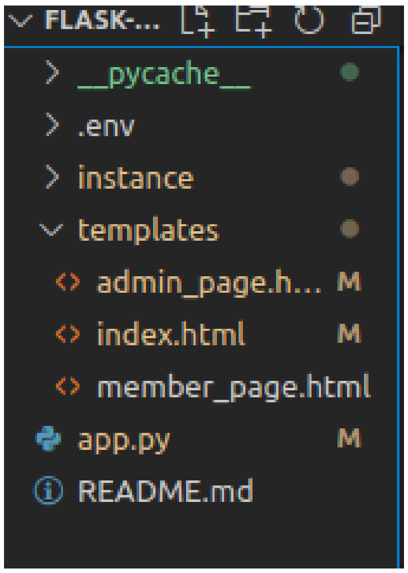
*flask –app <hello> run*

*flask run*

*python <app_name>.py*

## File Structure

Here, we are using the following folder and file.



```
FLASK-...
  > __pycache__
  > .env
  > instance
  v templates
      admin_page.h...  M
      index.html       M
      member_page.html
    app.py             M
    README.md
```

## Run a Flask Application

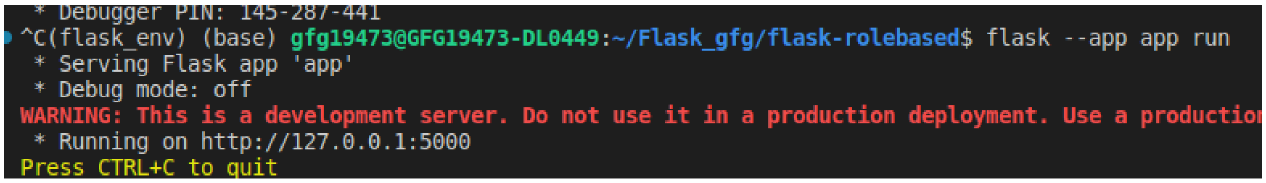
In this example, we have an application called **helloworld.py** below is the basic code for Flask.

Python

```python
# import flast module
from flask import Flask

# instance of flask application
app = Flask(__name__)

# home route that returns below text when root url is accessed
@app.route("/")
def hello_world():
    return "<p>Hello, World!</p>"

if __name__ == '__main__':
   app.run()
```

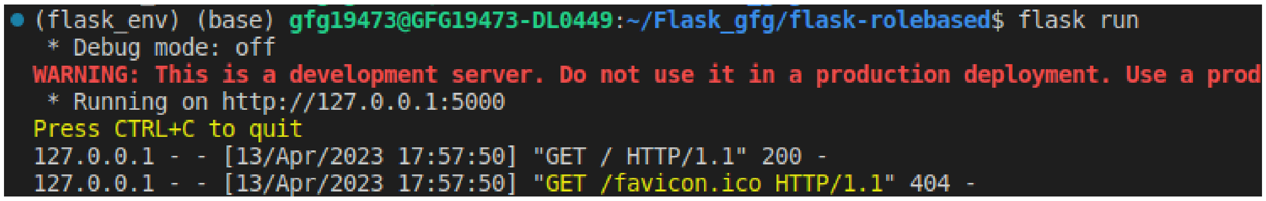
**Output:**

Using **flask –app <app_name> run**



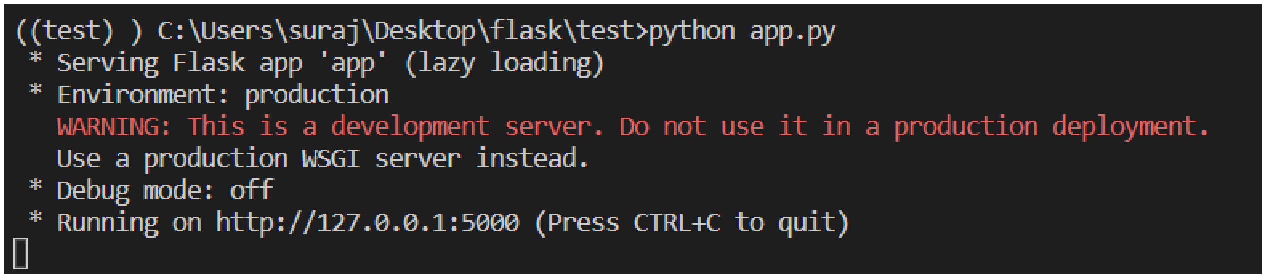
```
  * Debugger PIN: 145-287-441
^C(flask_env) (base) gfg19473@GFG19473-DL0449:~/Flask_gfg/flask-rolebased$ flask --app app run
  * Serving Flask app 'app'
  * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a productio
  * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Using **flask run**

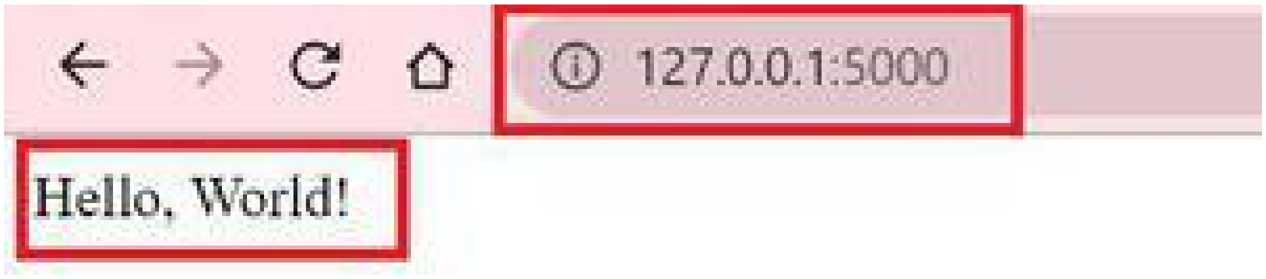


```
 (flask_env) (base) gfg19473@GFG19473-DL0449:~/Flask_gfg/flask-rolebased$ flask run
  * Debug mode: off
 WARNING: This is a development server. Do not use it in a production deployment. Use a prod
  * Running on http://127.0.0.1:5000
 Press CTRL+C to quit
 127.0.0.1 - - [13/Apr/2023 17:57:50] "GET / HTTP/1.1" 200 -
 127.0.0.1 - - [13/Apr/2023 17:57:50] "GET /favicon.ico HTTP/1.1" 404 -
```

Using the **python app_name.py**



```
((test) ) C:\Users\suraj\Desktop\flask\test>python app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
```

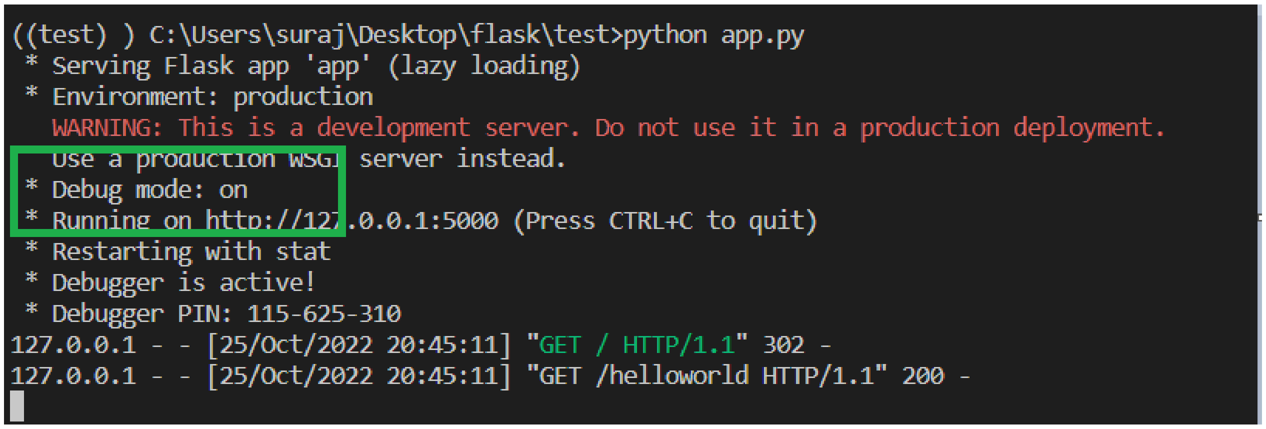


Hello, World!

## Run the app in the debugger

We will use the below command to run the flask application with debug mode as on. When debug mode is turned on, It allows developers to locate any possible error and as well the location of the error, by logging a traceback of the error.

> *if __name__ == '__main__':*
> *app.run(debug = True)*



```
((test) ) C:\Users\suraj\Desktop\flask\test>python app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 115-625-310
127.0.0.1 - - [25/Oct/2022 20:45:11] "GET / HTTP/1.1" 302 -
127.0.0.1 - - [25/Oct/2022 20:45:11] "GET /helloworld HTTP/1.1" 200 -
```

## How to Run a Flask Application – FAQs

### How Do I Run a Production Flask App?

> *Running a Flask application in production involves setting up a more robust and secure server than the built-in development server provided*

*by Flask. Here are the steps to deploy a Flask app for production:*

1. ***Choose a WSGI Server**: For production, it's recommended to use a WSGI server like Gunicorn or uWSGI. These servers are designed to manage and handle requests in a production environment efficiently.**Example with Gunicorn (on Linux/Mac):***
    - `-w 4` *specifies four worker processes. Replace* `app:app` *with your Flask app's module and variable names.*

2. ***Use a Reverse Proxy**: Setting up a reverse proxy like Nginx or Apache in front of your WSGI server can enhance security and performance. The reverse proxy handles HTTP requests, serving static files, and directing other requests to the WSGI server.*
3. ***Configuration and Security**: Configure your server for security, such as setting up HTTPS using SSL/TLS, configuring firewalls, and ensuring all dependencies are up to date.*
4. ***Environment Variables**: Use environment variables to manage configuration settings and sensitive information, keeping them out of the codebase.*

```
pip install gunicorn
gunicorn -w 4 app:app
```

## How Do I Run Flask App in Windows Terminal?

*To run a Flask application in the Windows Terminal:*

1. ***Set Environment Variables**:*
2. ***Run the Flask Application**:*
    - *This command will start the Flask development server. Ensure your Flask script is named* `app.py`*, or change* `FLASK_APP` *to your script's name.*

```
set FLASK_APP=app.py
set FLASK_ENV=development
```

```
flask run
```

## How Do I Make My Flask App Executable?

*To make your Flask app executable, meaning creating a standalone application that can be run directly without explicitly running Python, you can use tools like PyInstaller:*

1. ***Install PyInstaller**:*
2. ***Create an Executable**:*
   - *`--onefile` tells PyInstaller to bundle your app and all its dependencies into a single executable. Replace `app.py` with the path to your Flask script.*
3. ***Run the Executable**: After PyInstaller completes, find your executable in the `dist` folder. You can run it directly, and it will start your Flask application.*

```
pip install pyinstaller
```

```
pyinstaller --onefile app.py
```

## How Do I Run Flask REST API?

*Running a Flask REST API is similar to running any Flask app but tailored towards handling API requests:*

1. ***Define Your Flask App**:*
2. ***Run Your Flask App**: In your command line (e.g., Windows Terminal), set the environment variables and start the server.*
   - *This starts your Flask REST API which can now respond to HTTP requests at the defined routes.*

```
from flask import Flask, jsonify
```

```
app = Flask(__name__)

@app.route('/api/data', methods=['GET'])
def get_data():
    data = {'key': 'value'}
    return jsonify(data)

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

```
set FLASK_APP=app.py
set FLASK_ENV=development
flask run
```

*These steps will help you manage both development and production setups for Flask applications, making your web applications more robust, secure, and efficient.*

Y yash... 2

**Previous Article** Flask - (Creating first simple application)

**Next Article** Flask App Routing

## Similar Reads

Documenting Flask Endpoint using Flask-Autodoc

Documentation of endpoints is an essential task in web development and being able to apply it in different frameworks is always a utility. This article discusses...

4 min read

### How to use Flask-Session in Python Flask ?

Flask Session - Flask-Session is an extension for Flask that supports Server-side Session to your application.The Session is the time between the client logs in to...

4 min read

### How to Integrate Flask-Admin and Flask-Login

In order to merge the admin and login pages, we can utilize a short form or any other login method that only requires the username and password. This is know...

8 min read

### Minify HTML in Flask using Flask-Minify

Flask offers HTML rendering as output, it is usually desired that the output HTML should be concise and it serves the purpose as well. In this article, we would...

12 min read

### Flask URL Helper Function - Flask url_for()

In this article, we are going to learn about the flask url_for() function of the flask URL helper in Python. Flask is a straightforward, speedy, scalable library, used f...

11 min read

### How to Run Python Flask App Online using Ngrok?

Python Flask is a popular web framework for developing web applications, APIs, etc. Running flask apps on the local machine is very simple, but when it comes t...

2 min read

### How to run Flask App on Google Colab?

Flask is a web framework in python language. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This...

2 min read

### Profile Application using Python Flask and MySQL

A framework is a code library that makes a developer's life easier when building web applications by providing reusable code for common operations. There are ...

10 min read

**Create Postman collection from Flask application using flask2postman**

Prerequisite: Introduction to Postman, First App using Flask Since Postman is gaining popularity in the development domain, this article explains a way in...

3 min read

**Build a Flask application to validate CAPTCHA**

In this article, we are going a build a web application that can have a CAPTCHA. CAPTCHA is a tool you can use to differentiate between real users and...

5 min read

**Article Tags :** Python Technical Scripter python Technical Scripter 2022

**Practice Tags :** python python