

Develop Your First Web Application in Python

 webdamn.com/develop-your-first-web-application-in-python

webdamn

May 17, 2022

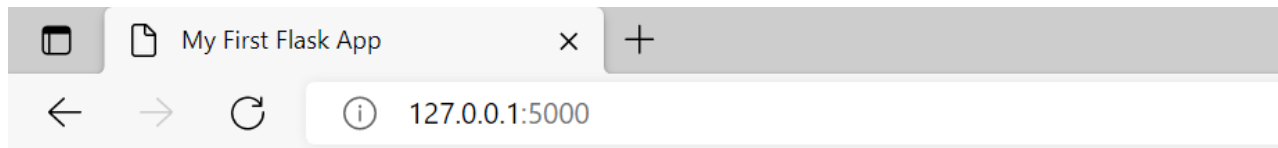
In our previous Python tutorial, we have explained how to [Create REST API in Python](#). In this tutorial, we will explain how to develop your first web application in Python.

In Python, there are many frameworks such as [Django](#), [Flask](#) which allows to easily create web pages and develop web applications.

In this tutorial, we will use [Flask](#) to develop our first web application in Python. The [Flask](#) is a popular micro web framework written in Python that provides useful tools and features to easily develop web applications. It is an easy to use and gives flexibility to new developers.

The [Flask](#) frameworks uses the [Jinja2](#) template engine by default. You can obviously use a different template engine but you still have to install [Jinja2](#) to run Flask itself.

In this tutorial, we will develop a web application that a user form with input field and a submit to submit form and display form submitted values.



First Name:

Last Name:

Email:

So let's proceed with developing web applications in Python.

Create Application Directory

First we will create our application directory. We will move to the Python installation directory to create application directory. Now we will create application directory `web-app-python` using below command.

```
$ mkdir web-app-python
```

We will move to the newly create application directory.

```
$ mkdir web-app-python
```

Install Flask

Now we will install `Flask` framework to use it to develop our web application with HTML form.

```
$ pip install Flask
```

Creating Webpage with Flask

After configuring project and installing required packages, let's get our hands on the code.

Now we will create a Python script `app.py` and use Flask module.

In below script, we will import flask module at top of script. The Flask constructor takes the name of current module (`__name__`) as argument. The `route()` function of the Flask class is a decorator, which tells the application which URL should call the associated function.

The `run()` method of Flask class runs the application on the local development server.

```
#app.py

from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello():
    return 'Hello, My application with Flask in Python!'
if __name__ == '__main__':
    app.run()
```

When we run above script, it will display details like below on command line:

```
* 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)
```

Then go to the url (<http://127.0.0.1:5000>) given above will see our first webpage displaying message there on your local server browser.

Creating Web Application using with Flask

The above application only displays a simple message without any HTML. Now we will create HTML Form page with input element using HTML template file.

The `Flask` provides a `render_template()` helper function that allows the use of the Jinja templates. This will make managing HTML much easier by writing your HTML code in `.html` files as well as using logic in your HTML code.

So now we will create a directory `templates` in our project. Then we will create `index.html` with following HTML to create HTML Form with input.

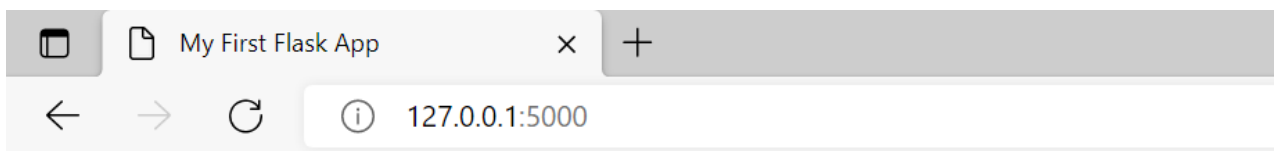
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=<<for>>, initial-scale=1.0">
<title>My First Flask App</title>
</head>
<body>
{% block content %}
<br><br>
<form action="{ { url_for("index") } }" method="post">
<label for="firstName">First Name:</label>
<input type="text" id="firstName" name="firstName" placeholder="First Name"><br>
<br>
<label for="lastName">Last Name:</label>
<input type="text" id="lastName" name="lastName" placeholder="Last Name"><br><br>
<label for="lastName">Email:</label>
<input type="email" id="email" name="email" placeholder="Email"><br><br>
<button class="submit"> submit</button>
{% endblock %}
</body>
</html>
```

Now save above file as `index.html` and render this using Python script to create the server.

```
#app.py
```

```
from flask import Flask, request, render_template
app = Flask(__name__)
@app.route('/', methods =["GET", "POST"])
def index():
    if request.method == "POST":
        firstName = request.form.get("firstName")
        lastName = request.form.get("lastName")
        email = request.form.get("email")
        return "Name: " + firstName + " " + lastName + " <br>Email: "+email
    return render_template("index.html")
if __name__=='__main__':
    app.run()
```

After running above script, the web server starts loading webpages in browser using url: **http://127.0.0.1:5000/** and display following HTML form with input and submit button.

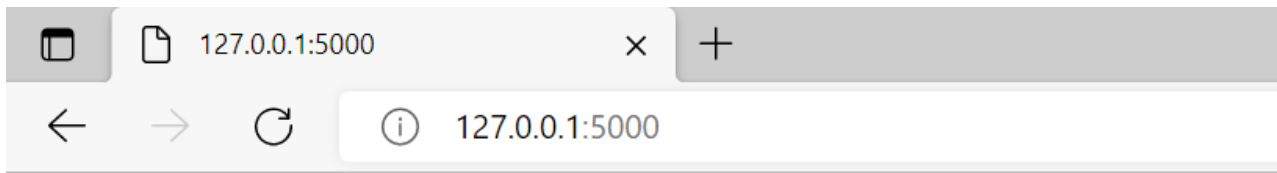


First Name:

Last Name:

Email:

The below page will display submit page result when above form submitted:



Name: web damn

Email: admin@webdamn.com