

Flask HTTP methods, handle GET & POST requests

Flask has different decorators to handle http requests. *Http protocol* is the basis for data communication in the *World Wide Web*.

Different methods for retrieving data from a specified URL are defined in this protocol. The following table summarizes the different http methods:

Request	Purpose
GET	The most common method. A GET message is send, and the server returns data
POST	Used to send HTML form data to the server. The data received by the POST method is not cached by the server.
HEAD	Same as GET method, but no response body.
PUT	Replace all current representations of the target resource with uploaded content.
DELETE	Deletes all current representations of the target resource given by the URL.

Related course: [Python Flask: Create Web Apps with Flask](#)

Flask HTTP Methods

Form

By default, the Flask route responds to GET requests. However, you can change this preference by providing method parameters for the route () decorator.

To demonstrate the use of a POST method in a URL route, first let us create an HTML form and use the POST method to send form data to the URL.

Save the following script as login.html

```
<html>
  <body>
    <form action = "http://localhost:5000/login" method = "post">
      <p>Enter Name:</p>
      <p><input type = "text" name = "nm" /></p>
      <p><input type = "submit" value = "submit" /></p>
    </form>
  </body>
</html>
```

GET and POST requests

To handle both GET and POST requests, we add that in the decorator `app.route()` method.

Whatever request you want, you can change it in the decorator.

Enter the following script in the Python shell.

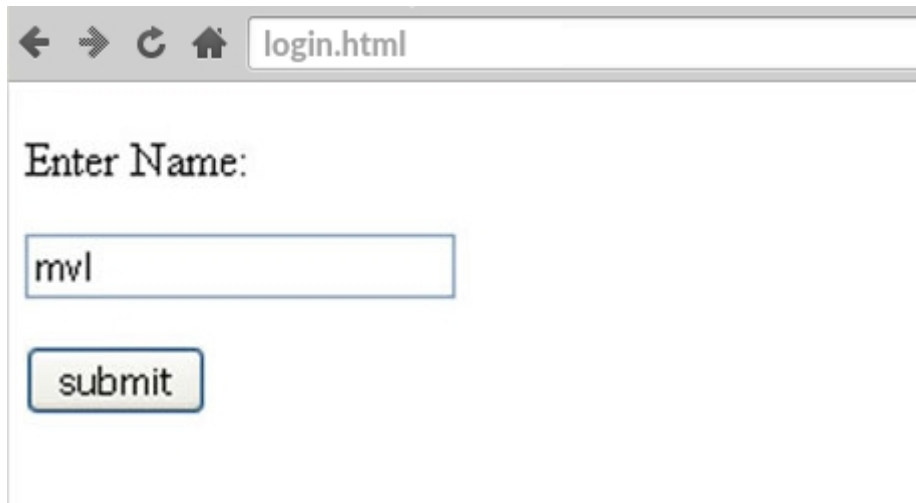
```
from flask import Flask, redirect, url_for, request
app = Flask(__name__)

@app.route('/success/<name>')
def success(name):
    return 'welcome %s' % name

@app.route('/login', methods = ['POST', 'GET'])
def login():
    if request.method == 'POST':
        user = request.form['nm']
        return redirect(url_for('success', name = user))
    else:
        user = request.args.get('nm')
        return redirect(url_for('success', name = user))

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

Once the development server is up and running, open login.html in the browser, enter the name in the text field, and then click Submit.



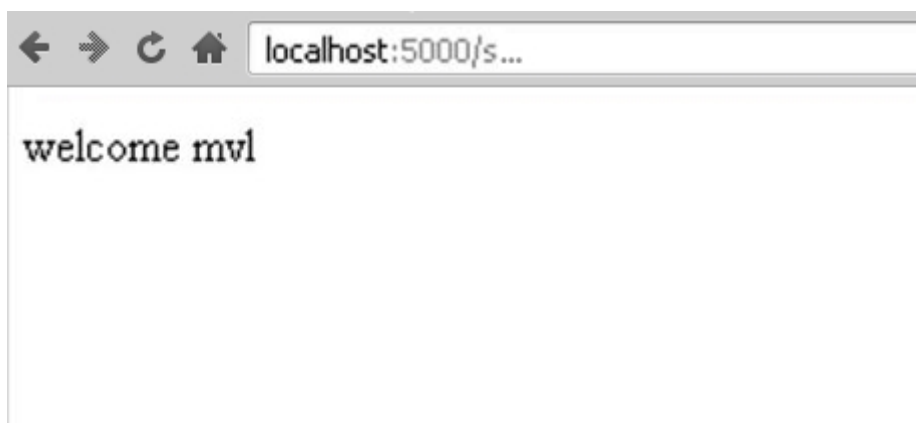
Enter Name:

The form data will POST to the URL in the action clause of the form label.

localhost/login image to the login() function. Because the server receives data through the POST method, the value of the “nm” parameter obtained from the form data is obtained by following these steps:

```
user = request.form['nm']
```

It is passed as part of the variable to the ‘/success’ URL. The browser displays a welcome message in the window.



welcome mvl

Change the method parameter to ‘GET’ in login.html, and then open it again in the browser. The data received on the server is obtained through the GET method. Get the value of the ‘nm’ parameter by:

```
user = request.args.get('nm')
```

Here, args are dictionary objects that contain the pair of form parameters and the list of their corresponding value pairs. The value corresponding to the 'nm' parameter is passed to the '/success' URL as before.

Related course: [Python Flask: Create Web Apps with Flask](#)

[Back](#)

[Next](#)

[Flask Tutorial: Routes](#)

[Python To Web - Your Python App online](#)