BACK TO TRACKS

Python / Django CS 221 - Flask v1.1 Flask Fundamentals **Basic Routes**

Basic Routes

Routes are the essential part of any web application.

Every route has two parts:

1. HTTP Verb (GET, POST, PUT, PATCH, DELETE)
2. URL

For the next couple of examples, we are going to focus only on GET routes. **But what is a route?** HTTP request. **When an HTTP request hits a server the server checks to see whether it can han route then runs and sends an appropriate HTTP response.**

We are going to add a templates folder alongside our hello.py file. Inside the templates folder, ac

/hello_flask/templates/index.html

```
<html>
  <body>
    <p>Hello Flask!</p>
  </body>
</html>
```

/hello_flask/hello.py

note the addition of render_template -- that allows us to return the rendered HTML that we crea

```
from flask import Flask, render_template  # Import Flask to allow us to create our app, and import
                                            # render_template to allow us to render index.html.
app = Flask(__name__)                      # Global variable __name__ tells Flask whether or not we
                                            # are running the file directly or importing it as a module.
@app.route('/')                          # The "@" symbol designates a "decorator" which attaches the
                                            # following function to the '/' route. This means that
                                            # whenever we send a request to localhost:5000/ we will ru
                                            # the following "hello_world" function.
def hello_world():
  return render_template('index.html')    # Render the template and return it!
app.run(debug=True)                         # Run the app in debug mode.
```

Here you can see that we are handling the '/' route with the hello_world function which renders HTTP verb is "GET".

ALL ROUTES ARE DEFAULT TO GET!

If no route is explicitly mentioned, it is a GET route. GET routes will mostly be used for serving p learn more about POST routes in the later chapters.

Now let's create a route that serves another page!

Add the following route to your hello.py file:

```
@app.route('/success')
def success():
  return render_template('success.html')
```

Now create the success.html file.

/hello_flask/templates/success.html

```
<html>
  <body>
    <p>Yay you successfully created another GET route that serves a page!</p>
  </body>
</html>
```

Make sure you understand how to add routes that serve pages before moving on to the assignm

Route Structure

When creating generic routes we write something like this:

☰ BACK TO TRACKS

Flask Fundamentals • 17% (/m/7/3677)

Overview (/m/

virtualenv (/m/

Flask Installation (/m/

○ Hello Flask (/m/

Basic Routes (/m/

○ Views

○ Flask Templates

○ Static Files

Landing Page

HTTP Methods

More Routing

Dojo Survey

Sessions

Counter

Hidden Inputs

Great Number Game

Ninja Gold

Basic Form Validation

Dojo Survey with Validation

More Form Validation

Registration Form

Advanced Routing

Disappearing Ninja

Flask + MySQL • 0% (/m/7/3678)

```
@app.route('/some_route')
def some_function_name():
  // your code here
```

The above route would be triggered when a user requests:

```
localhost:5000/some_route
```

In other words, our route rules define what comes after the initial forward-slash. When a request received by the server, the function that immediately follows the @app.route('/some_route') is ir

‹ PREVIOUS (/M/7/3677/24582)