**Webex Teams Hackathon 2018**



**Lab3 - Creating Endpoints to Serve Static Content**

# Objectives

In this lab, you will complete the following objectives:

* create REST API endpoint to serve static content from a directory
* create REST API endpoint to serve static index.html file

# Background / Scenario

Back end systems often serve not only dynamic but static content too. It is a good practice if the static front end contains Ajax requests\* to get data from REST API endpoints. Because the base URL of the front end is the same as the base URL of the back end therefor there will be no CORS\*\* problems at all.

**Note:**

* \* Ajax (Asynchronous JavaScript and XML) is a method of building interactive applications for the Web that process user requests immediately. Ajax allows content on Web pages to update immediately when a user performs an action, unlike an HTTP request, during which users must wait for a whole new page to load.
* \*\* Cross-origin resource sharing (CORS) is a mechanism that allows restricted resources on a web page to be requested from another domain outside the domain from which the first resource was served. Ajax requests, are forbidden by default by the same-origin security policy.

When completed, the **lab3-serve-static.py** program will serve static content from a specific directory of the local file system. index.html from this directory will be served when root URL of the site requested.

# Required Resources

* Postman application
* Python 3 with IDLE
* Python code files

**Step 1: Import os module**

In this step, you will create your next project based on the Lab2 project.

1. Create your project directory called lab3-serve-static. Copy the final version of your lab2–sqlite.py file from lab3–sqlite directory to lab3-serve-static directory. Rename the new file to lab3-serve-static.py.
2. Edit your lab3-serve-static.py file. Import the os. The os module provides a portable way of using operating system dependent functionality. Add the following text right after the last import command:   
   import os

We need to import send\_from\_directory module from flask package too. Add the module name to the end of flask import line:

from flask import Flask, request, jsonify, send\_from\_directory

**Step 2: Create a directory with static HTML content**

1. Create a new directory in the root folder of your project called static
2. Create a text file in the static directory called index.html
3. Copy the content of Bootstrap 4 starter template into index.html from the following URL: <https://getbootstrap.com/docs/4.2/getting-started/introduction/> (Basic template section).

**Step 3: Create endpoint to serve static content from a directory**

1. Get the appropriate operating system dependent path of the static directory. Before the end point definitions add the following text:

static\_file\_dir = os.path.join(os.path.dirname(os.path.realpath(\_\_file\_\_)), 'static')

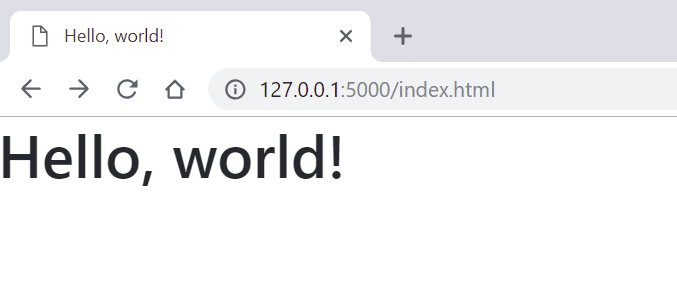
1. Create a REST API endpoint which will serve static content from static directory.

@app.route('/<path:path>', methods=['GET'])

def serve\_static\_dir(path):

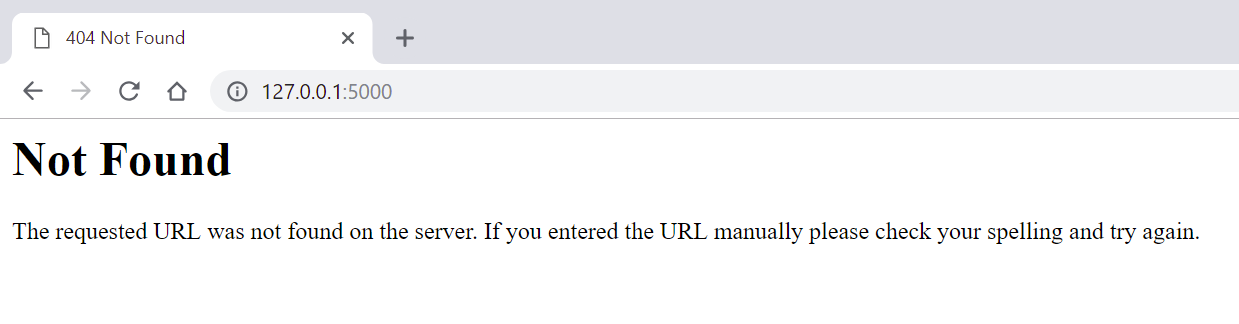
return send\_from\_directory(static\_file\_dir, path)

1. Start your app. In the command line type python lab3-serve-static.py
2. Open your Chrome browser and type the following url: <http://127.0.0.1:5000/index.html>



**Step 4: Create endpoint to serve index.html when root URL of the site requested**

1. In your browser try to get the root URL of the site: <http://127.0.0.1:5000/>. You will get error message:



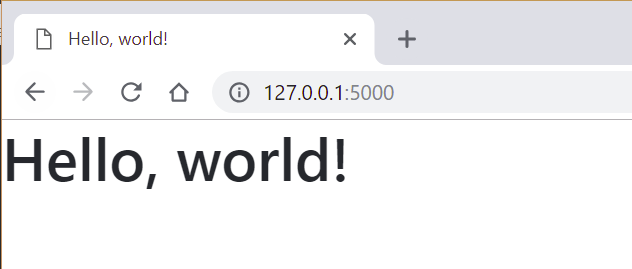
1. To serve index.html when the root URL of the site requested add one more endpoint before the previous endpoint:

@app.route("/")

def index():

return app.send\_static\_file('index.html')

1. Restart your back end app. Press Crtl-C in the command windows then type python lab3-serve-static.py
2. In your browser try the root URL of the host again: <http://127.0.0.1:5000/>:



**Step 4: Create a css file for your static front end site**

1. Create a style.css file in the static directory.
2. Add the following content into your style.css file:

body {

margin: 25px;

background-color: grey;

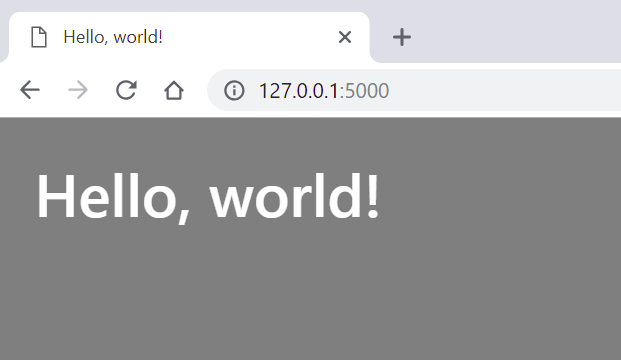
color: white;

}

1. Link the style.css into your index.html. Add the following line after the Bootstrap stylesheet link tag.

<link rel="stylesheet" href="style.css">

1. Reload the page in your browser. Press Ctrl-F5.



**Note:** If you create other files (like JavaScript, images etc.) in static directory they also will be served.

1. Stop your back end app. Press Crtl-C in the command windows.