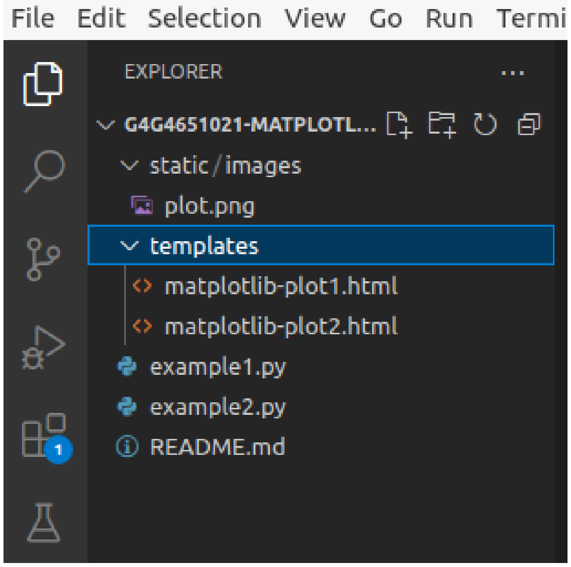
Create Scatter Charts in Matplotlib using Flask

Last Updated : 05 Feb, 2023

In this article, we will see how to create charts in Matplotlib with Flask. We will discuss two different ways how we can create Matplotlib charts in Flask and present it on an HTML webpage with or without saving the plot using Python.

File structure



```
G4G4651021-MATPLOTL...
  static/images
    plot.png
  templates
    matplotlib-plot1.html
    matplotlib-plot2.html
  example1.py
  example2.py
  README.md
```

Create and Save the Plot in the Static Directory

Here, We first created a get_plot() function which generates the Matplotlib plot and returns the plot object. Python's Numpy library generates random data for this plot. It is not necessary to import if you are defining your own data. The root URL ('/') first calls this function to get the plot object. It then saves this plot object as 'plot.png' under the images folder present inside the static directory. This is the default location defined by Flask for static files like images, CSS, JS, etc. The final step is to render the below HTML script which reads the image file from the directory and renders it to the web browser as shown in the output image.

Python3

```python
# Importing required functions
import numpy as np
import matplotlib.pyplot as plt
from flask import Flask, render_template


# Flask constructor
app = Flask(__name__)

# Generate a scatter plot and returns the figure
def get_plot():
    data = {
        'a': np.arange(50),
        'c': np.random.randint(0, 50, 50),
        'd': np.random.randn(50)
    }
    data['b'] = data['a'] + 10 * np.random.randn(50)
    data['d'] = np.abs(data['d']) * 100

    plt.scatter('a', 'b', c='c', s='d', data=data)
    plt.xlabel('X label')
    plt.ylabel('Y label')
    return plt

# Root URL
@app.get('/')
def single_converter():
    # Get the matplotlib plot
    plot = get_plot()

    # Save the figure in the static directory
    plot.savefig(os.path.join('static', 'images', 'plot.png'))

    return render_template('matplotlib-plot1.html')

# Main Driver Function
if __name__ == '__main__':
    # Run the application on the local development server
    app.run(debug=True)
```

Save the HTML file as **‘matplotlib-plot1.html’** under the templates folder in the root directory.

HTML

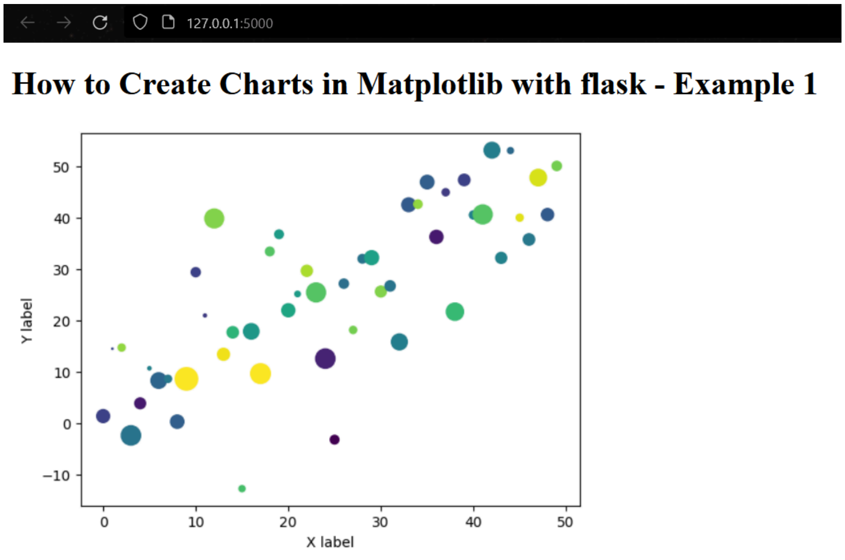
```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Matplotlib Plot</title>
</head>

<body>
    <h1>How to Create Charts in Matplotlib with flask - Example 1</h1>
    <img src="{{ url_for('static', filename='images/plot.png') }}">
</body>

</html>
```

Output:



How to Create Charts in Matplotlib with flask - Example 1

Example 1 – Output

Generating a Base64 I/O String of the Plot

We will generate the base64 I/O) string format for the image and pass this to the template to render the plot. Here, We first created a get_plot() function which generates the Matplotlib plot and returns the plot object. Python’s

Numpy library generates random data for this plot. It is not necessary to import if you are defining your own data. The root URL ('/') first calls this function to get the plot object. It then creates a Base64 I/O equivalent format of the string using python's built-in 'io' and 'base64' modules. The final step is to render the below HTML script and also pass this string to the template. The template reads the image file using the string that is passed to it.

Python

```python
# Importing required functions
from flask import Flask, render_template
import matplotlib.pyplot as plt
import os
import numpy as np
import matplotlib
matplotlib.use('Agg')


# Flask constructor
app = Flask(__name__)

# Generate a scatter plot and returns the figure
def get_plot():

    data = {
        'a': np.arange(50),
        'c': np.random.randint(0, 50, 50),
        'd': np.random.randn(50)
    }

    data['b'] = data['a'] + 10 * np.random.randn(50)
    data['d'] = np.abs(data['d']) * 100

    plt.scatter('a', 'b', c='c', s='d', data=data)
    plt.xlabel('X label')
    plt.ylabel('Y label')

    return plt

# Root URL
@app.get('/')
def single_converter():
    # Get the matplotlib plot
    plot = get_plot()

    # Save the figure in the static directory
    plot.savefig(os.path.join('static', 'images', 'plot.png'))

    # Close the figure to avoid overwriting
    plot.close()
    return render_template('matplotlib-plot1.html')

# Main Driver Function
if __name__ == '__main__':
    # Run the application on the local development server
    app.run(debug=True)
```

Save the HTML file as **‘matplotlib-plot2.html’** under the templates folder in the root directory.

HTML

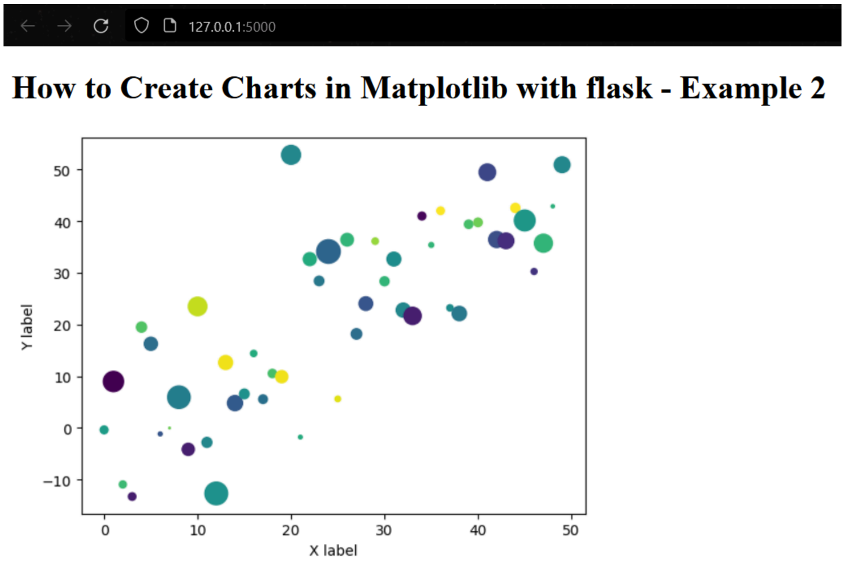
```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Matplotlib Plot</title>
</head>

<body>
    <h1>How to Create Charts in Matplotlib with flask - Example 2</h1>
    <img src="data:image/png;base64,{{ s }}">
</body>

</html>
```

Output:



127.0.0.1:5000

How to Create Charts in Matplotlib with flask - Example 2

Example 2 – Output

Related Articles: [How to Add Graphs to Flask apps]

apath... 1

Previous Article

Create a Bar Chart From a DataFrame with Plotly and Flask

Next Article

Implement ChatGPT in a Flask Application

Similar Reads

Python | Plotting scatter charts in excel sheet using XlsxWriter module

Prerequisite: Create and Write on an excel file. XlsxWriter is a Python library using which one can perform multiple operations on excel files like creating,...

10 min read

How to create charts in bokeh with flask

In this post, we will use the Flask framework to create our web application in Python. The Bokeh library will be used to create interactive bar graphs and we...

5 min read

Pandas Scatter Plot – DataFrame.plot.scatter()

A Scatter plot is a type of data visualization technique that shows the relationship between two numerical variables. For plotting to scatter plot using pandas there...

2 min read

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

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

How to create a Scatter Plot with several colors in Matplotlib?

Matplotlib is a plotting library for creating static, animated, and interactive visualizations in Python. Matplotlib can be used in Python scripts, the Python an...

3 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

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

Create lollipop charts with Pandas and Matplotlib

In this article, we will create Lollipop Charts. They are nothing but a variation of the bar chart in which the thick bar is replaced with just a line and a dot-like “o”...

4 min read

Article Tags : Python Technical Scripter Python Flask Python-matplotlib +1 More

Practice Tags : python

In Media

Contact Us

Advertise with us

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

DSA

Data Structures

Algorithms

DSA for Beginners

Basic DSA Problems

DSA Roadmap

Top 100 DSA Interview Problems

DSA Roadmap by Sandeep Jain

All Cheat Sheets

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

C++

PHP

GoLang

SQL

R Language

Android Tutorial

Tutorials Archive

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects