Integrating Bokeh Visualizations in Django Application

Last Updated : 15 Aug, 2021

Bokeh is an interactive visualization library that helps us to create a visual representation of data set and interact with it. You can create various types of visualizations such as bar charts, horizontal plots,time-series, etc. There are various methods to include Bokeh apps and widgets into web apps and pages.

In this tutorial, we are going to create a basic bokeh graph and embed it into our Django web app. For that, we will be importing components from *bokeh.embed* which returns the individual components. The function *bokeh.embed.components()* returns a script that contains that data for your plot with a *<div> tag* in which the plot view is loaded. We will look in detail at the step-by-step procedure.

Step 1: Setting up a basic Django project

For this project, we are using PyCharm IDE. PyCharm is one of the most popular IDE used for the python Scripting language.

- Open PyCharm and create a new project and save it as BokehProject

```
pip install django
```

- In the same way, we will install bokeh in our project as:

```
pip install bokeh
```

Step 2: Create the Django project

- Create a Django project using the following command:

```
django-admin startproject BokehDjango
```

- Change the project folder using the below command:

```
cd BokehDjango
```

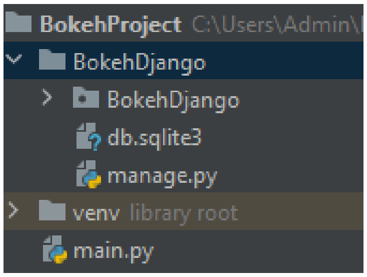
- Run manage.py to initially migrate data changes to our project by using migrate as below

```
python manage.py migrate
```

- Create a superuser using the following command to create a **superuser** account

```
python manage.py createsuperuser
```

- Add the name, email, and password.
- At this stage, the directory structure is as shown below:

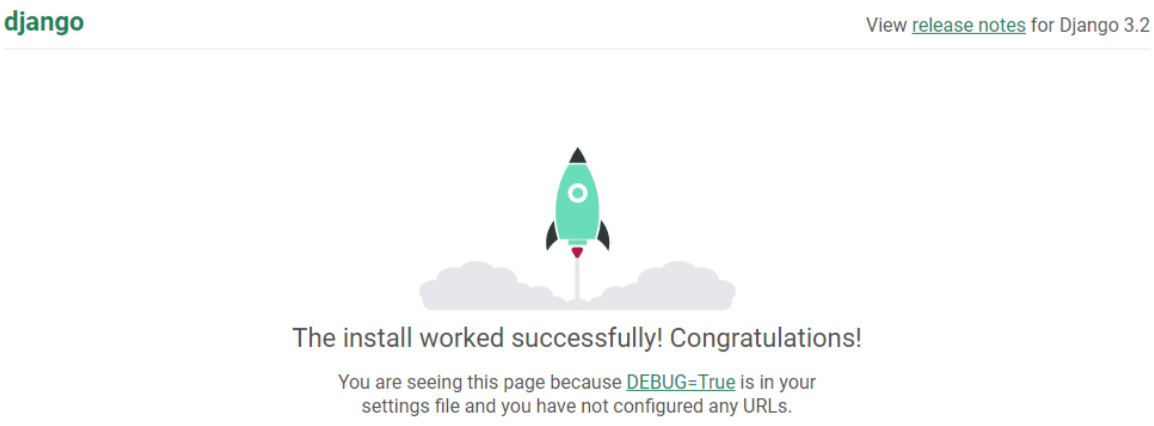


```
BokehProject C:\Users\Admin\
  BokehDjango
    BokehDjango
    db.sqlite3
    manage.py
  venv library root
  main.py
```

- Now let us run the command below command to check if Django is installed successfully.

```
python manage.py runserver
```

- Navigate to the address http://127.0.0.1:8000/ and you will see something like this.



django

View release notes for Django 3.2

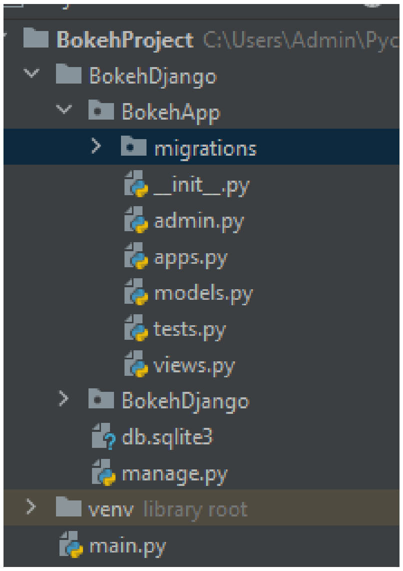
The install worked successfully! Congratulations!

You are seeing this page because DEBUG=True is in your settings file and you have not configured any URLs.

- Now we create a Django application using the following command

```
python manage.py startapp BokehApp
```

- The directory structure at this stage will be as shown below:



```
BokehProject C:\Users\Admin\Pyc
  BokehDjango
    BokehApp
      migrations
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      views.py
    BokehDjango
    db.sqlite3
    manage.py
  venv library root
  main.py
```

- Since we have created an app we need to add it to settings. Open settings.py and add the following in installed apps:

```
INSTALLED_APPS = [
   'django.contrib.admin',
   'django.contrib.auth',
   'django.contrib.contenttypes',
   'django.contrib.sessions',
   'django.contrib.messages',
   'django.contrib.staticfiles',
    'BokehApp',
]
```

- Update urls.py file and add URL patterns. Open *urls.py* from our project folder i.e.,BokehDjango, and add the include function in the import statement. Also, add the path to include the URLs of our new application as shown below:

Python

```
from django.contrib import admin
from django.urls import path, include


urlpatterns = [
    path("admin/", admin.site.urls),
    path("", include("BokehApp.urls")),
]
```

- Now create a new file in our app folder i.e., BokehApp, and save it as *urls.py.*
- Open the file and add the path to route for your home page as below and also don't forget to import path and views.

Python

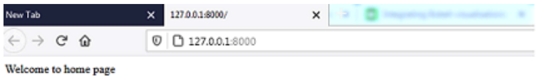
```
from django.urls import path
from . import views

urlpatterns = [path("", views.home, name="home")]
```

- Next, we create the view for our home page that will render our first Bokeh Graph. Open views.py and create a new method called *home()* and before that we import HttpResponse..HttpResponse is most frequently used as a return object from a Django view.
- As of now, we are simply displaying a welcome message as below:

Python

```python
from django.shortcuts import render
from django.http import HttpResponse

# Create your views here.
def home(request):
    return HttpResponse("Welcome to home page")
```

- Let us run the server using python manage.py runserver and look at the result:



Welcome to home page

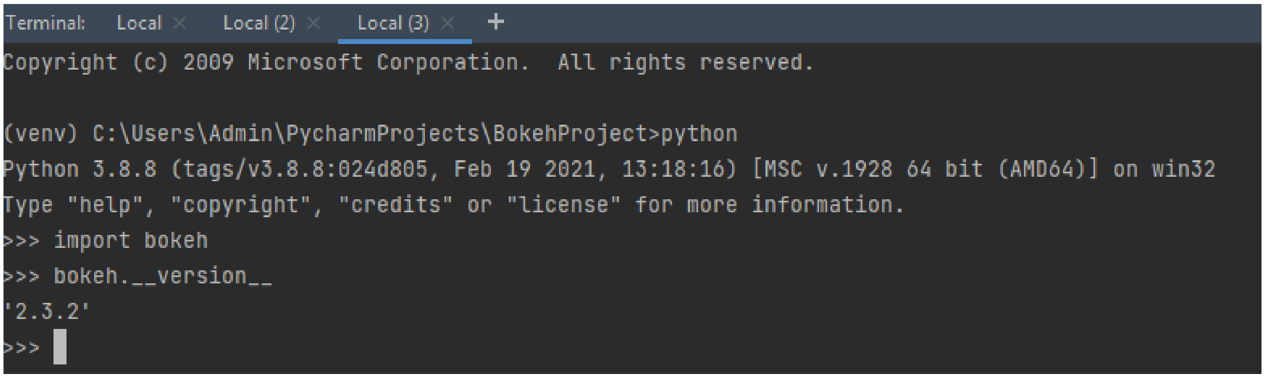
Great! So this was all about setting our Django website.

Step 3:Complete Bokeh Setup into our project:

- Go to your python shell and check the version of Bokeh as:

```
bokeh.__version__
```

- As shown in the image below:



```
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

(venv) C:\Users\Admin\PycharmProjects\BokehProject>python
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import bokeh
>>> bokeh.__version__
'2.3.2'
>>>
```

- Now let’s create a template folder in our BokehApp directory and save it as templates. Create a new file in the templates directory and save it as base.html.
- Add the following links of CSS in your *base.html* file in the head tag and replace the version of your bokeh at the place *bokeh-x.y.z.min (underlined place x.y.z.)*

HTML

```
<link href=”http://cdn.pydata.org/bokeh/release/bokeh-2.3.2.min.css
   " rel=”stylesheet” type=”text/css”>
<link href=”http://cdn.pydata.org/bokeh/release/bokeh-widgets-2.3.2.min.css"
   rel=”stylesheet” type=”text/css”>
```

- And the JavaScript links below the ending body tag i.e., after </body> and similarly replace your bokeh version at x.y.z

HTML

```
<script src="https://cdn.bokeh.org/bokeh/release/bokeh-2.3.2.min.js"></script>
<script src="https://cdn.bokeh.org/bokeh/release/bokeh-widgets-2.3.2.min.js"></scr
<script src="https://cdn.bokeh.org/bokeh/release/bokeh-tables-2.3.2.min.js"></scri
<script src="https://cdn.bokeh.org/bokeh/release/bokeh-api-2.3.2.min.js"></script>
```

- The *base.html* file looks like

HTML

```
<html>
   <head>
      <link href=”http://cdn.pydata.org/bokeh/release/bokeh-2.3.2.min.css
         " rel=”stylesheet” type=”text/css”>
      <link href=”http://cdn.pydata.org/bokeh/release/bokeh-widgets-2.3.2.min.css"
         rel=”stylesheet” type=”text/css”>
   </head>
   <body>
      <h1>Our first Bokeh Graph</h1>
      {{div| safe}}
   </body>
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-2.3.2.min.js"></script>
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-widgets-2.3.2.min.js"></
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-tables-2.3.2.min.js"></s
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-api-2.3.2.min.js"></scri
   {{script| safe}}
</html>
```

- Let us now replace the view function home so that it renders our first graph. Add the below code which creates basic circle scatter marks on our plot:

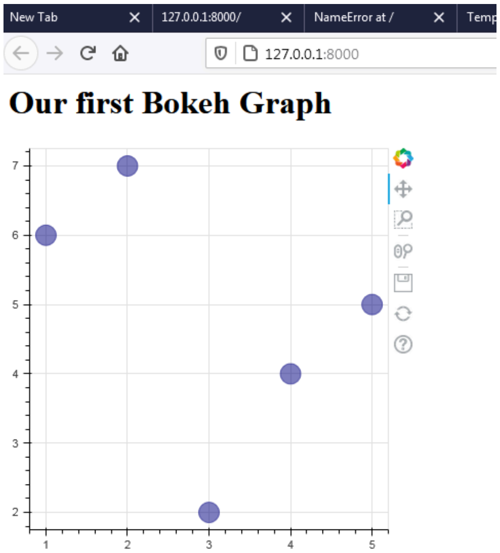
Python

```
from django.shortcuts import render
from django.http import HttpResponse
from bokeh.plotting import figure
from bokeh.embed import components

# Create your views here.

def home(request):

   #create a plot
    plot = figure(plot_width=400, plot_height=400)

   # add a circle renderer with a size, color, and alpha

   plot.circle([1, 2, 3, 4, 5], [6, 7, 2, 4, 5], size=20, color="navy", alpha=0.5)

   script, div = components(plot)
```

```
    return render(request, 'base.html', {'script': script, 'div': div})
```

- The **components** method returns a script that contains the data for your plot and provides a <div>tag to display the plot view. These two elements can be inserted into the HTML text and the <script> when executed will replace the div with the plot.
- The **circle** method is a glyph method which is a method of the **figure** object. Glyphs are the basic visual building blocks of Bokeh plots. This includes elements such as lines, rectangles, squares, wedges, or the circles of a scatter plot
- The **plot** variable enables us to create a plot that holds all the various objects such as glyphs, annotations, etc. of a visualization.

So let us refresh our page after saving all the files and the output will be as shown below.



Our first Bokeh Graph

To enhancing the look of the page we are adding bootstrap to our base.html file. We have added a few of the components and the final HTML will be as shown below:

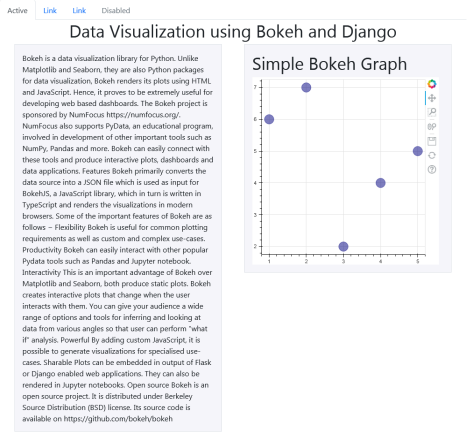
HTML

```
<html>
    <head>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.
          rel="stylesheet"
```

```
      integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7
      crossorigin="anonymous">
    <link href=”http://cdn.pydata.org/bokeh/release/bokeh-2.3.2.min.css
      " rel=”stylesheet” type=”text/css”>
    <link href=”http://cdn.pydata.org/bokeh/release/bokeh-widgets-2.3.2.min.css"
      rel=”stylesheet” type=”text/css”>
  </head>
  <body>
    <ul class="nav nav-tabs">
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="#">Active</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#" tabindex="-1"
          aria-disabled="true">Disabled</a>
      </li>
    </ul>
    <h1 align="center">Data Visualization using Bokeh and Django</h1>
    <div class="container overflow-hidden">
      <div class="row gx-5">
        <div class="col">
          <div class="p-3 border bg-light">Bokeh is a data
            visualization library for Python. Unlike Matplotlib and
            Seaborn, they are also Python packages for data visualization,
            Bokeh renders its plots using HTML and
            JavaScript. Hence, it proves to be extremely useful
            for developing web based dashboards.
            The Bokeh project is sponsored by NumFocus
            https://numfocus.org/. NumFocus also supports PyData, an
            educational program, involved in development of
            important tools such as NumPy, Pandas and more.
            Bokeh can easily connect with these tools and
            produce interactive plots, dashboards and data applications.
            Features
            Bokeh primarily converts the data source into a JSON file
            which is used as input for BokehJS, a JavaScript library,
            which in turn is written in TypeScript and renders the
            visualizations in modern browsers.
            Some of the important features of Bokeh are as follows −
            Flexibility
            Bokeh is useful for common plotting requirements as
            well as custom and complex use-cases.
            Productivity
```

```
                  Bokeh can easily interact with other popular Pydata
                  tools such as Pandas and Jupyter notebook.
                  Interactivity
                  This is an important advantage of Bokeh over Matplotlib and
                  Seaborn, both produce static plots. Bokeh
                  creates interactive plots that change when the user
                  interacts with them. You can give your audience a
                  wide range of options and tools for inferring and
                  looking at data from various angles so that user can
                  perform “what if” analysis.
                  Powerful
                  By adding custom JavaScript, it is possible to generate
                  visualizations for specialised use-cases.
                  Sharable
                  Plots can be embedded in output of Flask or Django
                  enabled web applications. They can also be rendered in
                  Jupyter notebooks.
                  Open source
                  Bokeh is an open source project. It is distributed under
                  Berkeley Source Distribution (BSD) license. Its
                  source code is available on https://github.com/bokeh/bokeh
               </div>
            </div>
            <div class="col">
               <div class="p-3 border bg-light">
                  <h1>Simple Bokeh Graph</h1>
                  {{ div| safe}}
               </div>
            </div>
         </div>
      </div>
      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.
         integrity="sha384-gtEjrD/SeCtmISkJkNUaaKMoLD0//ElJ19smozuHV6z3Iehds+3Ulb9
         crossorigin="anonymous"></script>
   </body>
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-2.3.2.min.js"></script>
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-widgets-2.3.2.min.js"></
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-tables-2.3.2.min.js"></s
   <script src="https://cdn.bokeh.org/bokeh/release/bokeh-api-2.3.2.min.js"></scri
   {{script| safe}}
</html>
```

Output:



Data Visualization using Bokeh and Django

Bokeh is a data visualization library for Python. Unlike Matplotlib and Seaborn, they are also Python packages for data visualization, Bokeh renders its plots using HTML and JavaScript. Hence, it proves to be extremely useful for developing web based dashboards. The Bokeh project is sponsored by NumFocus https://numfocus.org/. NumFocus also supports PyData, an educational program, involved in development of other important tools such as NumPy, Pandas and more. Bokeh can easily connect with these tools and produce interactive plots, dashboards and data applications. Features Bokeh primarily converts the data source into a JSON file which is used as input for BokehJS, a JavaScript library, which in turn is written in TypeScript and renders the visualizations in modern browsers. Some of the important features of Bokeh are as follows – Flexibility Bokeh is useful for common plotting requirements as well as custom and complex use-cases. Productivity Bokeh can easily interact with other popular Pydata tools such as Pandas and Jupyter notebook. Interactivity This is an important advantage of Bokeh over Matplotlib and Seaborn, both produce static plots. Bokeh creates interactive plots that change when the user interacts with them. You can give your audience a wide range of options and tools for inferring and looking at data from various angles so that user can perform “what if” analysis. Powerful By adding custom JavaScript, it is possible to generate visualizations for specialised use-cases. Sharable Plots can be embedded in output of Flask or Django enabled web applications. They can also be rendered in Jupyter notebooks. Open source Bokeh is an open source project. It is distributed under Berkeley Source Distribution (BSD) license. Its source code is available on https://github.com/bokeh/bokeh

Simple Bokeh Graph

N nama... 1

Previous Article **Next Article**

Setting Up a Virtual Environment in Django

Similar Reads

Integrating Django with Reactjs using Django REST Framework

In this article, we will learn the process of communicating between the Django Backend and React js frontend using the Django REST Framework. For the sake...

15+ min read

Integrating TinyMCE with Django

TinyMCE is a online rich text editor which is fully flexible and provides customisation. mostly used to get dynamic data such as articles in GFG and muc...

2 min read

Integrating Facebook Comments Plugin in Django Project

Django is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because...

2 min read

Integrating Facebook Like & Share Plugin Using Django

Django is a Python-based web framework that allows you to quickly create efficient web applications. It is also called batteries included framework because...

2 min read

Integrating Deep Learning model With Django

In this project, we will be integrating a deep learning model with the Django framework. This deep learning model converts greyscale images to colorful...

12 min read

Creating Interactive 3D Visualizations using PyVista

In this article, we will first explore the benefits of using PyVista in Python. Next, we will guide you through the installation of this library on your system. Then, ...

5 min read

Integrating Facebook Like, Comments and Share Plugin in Flask Project

Flask is a Framework of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask’s framework is more explicit than...

2 min read

Integrating HAL in Flask

Hypertext Application Language (HAL) is a standard that is used to establish conventions for expressing hypermedia controls, such as links, with JSON. It is...

2 min read

Integrating Java with Python

While programming in a language, a developer may feel the need to use some functionality that may have better support in another language. For example,...

4 min read

Integrating Poetry in Python with version control systems

Version control systems, like Git, are crucial components in today's software application development processes. They assist in controlling modification to th...

7 min read

Article Tags : Python Django-Projects Python Django Python-Bokeh

Practice Tags : python

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company
About Us
Legal
In Media
Contact Us
Advertise with us

Languages
Python
Java
C++
PHP
GoLang

GFG Corporate Solution

Placement Training Program

GeeksforGeeks Community

SQL

R Language

Android Tutorial

Tutorials Archive

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

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning

ML Maths

Data Visualisation

Pandas

NumPy

NLP

Deep Learning

Web Technologies

HTML

CSS

JavaScript

TypeScript

ReactJS

NextJS

Bootstrap

Web Design

Python Tutorial

Python Programming Examples

Python Projects

Python Tkinter

Web Scraping

OpenCV Tutorial

Python Interview Question

Django

Computer Science

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Software Development

Software Testing

DevOps

Git

Linux

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

Inteview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

School Subjects

Mathematics

Physics

GeeksforGeeks Videos

DSA

Python

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

Java

C++

Web Development

Data Science

CS Subjects