



[Django](#) [Views](#) [Model](#) [Template](#) [Forms](#) [Jinja](#) [Python SQLite](#) [Flask](#) [Json](#) [Postman](#) [Interview Ques](#)

How to Create a basic API using Django Rest Framework ?

Last Updated : 01 Sep, 2023

Django REST Framework is a wrapper over the default Django Framework, basically used to create APIs of various kinds. There are three stages before creating an API through the REST framework, Converting a Model's data to JSON/XML format (Serialization), Rendering this data to the view, and Creating a URL for mapping to the views.

This article revolves around how to create a basic API using the Django REST Framework. It assumes you are familiar with Django basics – [Django tutorial](#)—also, installation of [Django REST Framework](#). Assuming you have created a project named **geeksforgeeks** with Django, let's initiate Django REST Framework.

Steps

- *Add rest_framework to INSTALLED_APPS*
- *Create a app and model*
- *Serialization*
- *Creating a viewset*
- *Define URLs of API*
- *Run server and check API*

Creating API in Django using Django Rest Framework

Add rest_framework to INSTALLED_APPS

To initialize REST Framework in your project, go to settings.py, and in `INSTALLED_APPS` add `'rest_framework'` at the bottom to create api in python django.

Python3

Application definition

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

Create a app and model

Now, let's create a app using command,

```
python manage.py startapp apis
```

A folder with name apis would have been registered by now. let's add this app to `INSTALLED_APPS` and `urls.py` also.

In, settings.py.

Python3

Application definition

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

```
]
```

Now, add APIs urls in urls.py. In `geeksforgeeks.urls.py`.

Python3

```
from django.contrib import admin
# include necessary libraries
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    # add apis urls
    path('', include("apis.urls"))
]
```

Create a model

To demonstrate, create api in python django and using an API, let's create a model named "GeeksModel". In `apis/models.py`

Python3

```
from django.db import models

class GeeksModel(models.Model):
    title = models.CharField(max_length=200)
    description = models.TextField()

    def __str__(self):
        return self.title
```

now our app is ready, let's serialize the data and create views from the same.

Serializing Django objects

Serializers allow complex data such as querysets and model instances to be converted to native Python datatypes that can then be easily rendered into JSON, XML or other content types. Serializers also provide deserialization, allowing parsed data to be converted back into complex types, after first validating the incoming data. Let's start creating a serializer, in file `apis/serializers.py`,

Python3

```
# import serializer from rest_framework
from rest_framework import serializers

# import model from models.py
from .models import GeeksModel

# Create a model serializer
class GeeksSerializer(serializers.HyperlinkedModelSerializer):
    # specify model and fields
    class Meta:
        model = GeeksModel
        fields = ('title', 'description')
```

Creating a viewset

To render data into frontend, and handle requests from user, we need to create a view. In Django REST Framework, we call these as viewsets, so let's create a view in `apis/views.py`,

Python3

```
# import viewsets
from rest_framework import viewsets

# import local data
from .serializers import GeeksSerializer
from .models import GeeksModel

# create a viewset
```

```
class GeeksViewSet(viewsets.ModelViewSet):  
    # define queryset  
    queryset = GeeksModel.objects.all()  
  
    # specify serializer to be used  
    serializer_class = GeeksSerializer
```

Define URLs of API

Now create api in python django and Specify the url path of APIs to be accessed, In apis/urls.py,

Python3

```
# basic URL Configurations  
from django.urls import include, path  
# import routers  
from rest_framework import routers  
  
# import everything from views  
from .views import *  
  
# define the router  
router = routers.DefaultRouter()  
  
# define the router path and viewset to be used  
router.register(r'geeks', GeeksViewSet)  
  
# specify URL Path for rest_framework  
urlpatterns = [  
    path('', include(router.urls)),  
    path('api-auth/', include('rest_framework.urls'))  
]
```

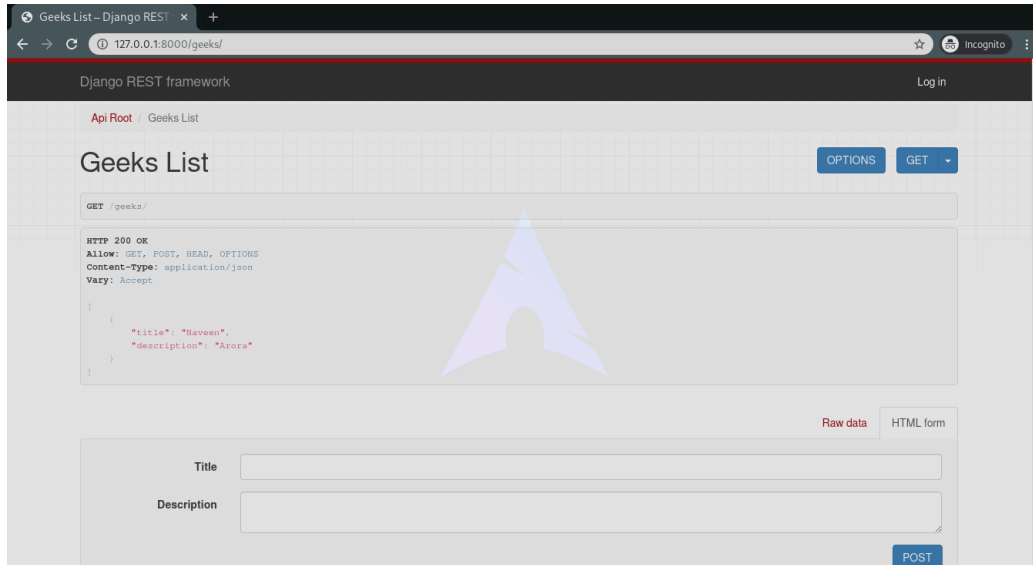
After everything is successfully ready, let's run some commands to activate the server.

Run server and check API

Run following commands to create the database, and then run server,

```
python manage.py makemigrations
python manage.py migrate
python manage.py runserver
```

Now visit <http://127.0.0.1:8000/geeks/>,



Are you ready to elevate your web development skills from foundational knowledge to advanced expertise? Explore our [Mastering Django Framework - Beginner to Advanced Course](#) on GeeksforGeeks, designed for aspiring developers and experienced programmers. This comprehensive course covers everything you need to know about Django, from the basics to advanced features. Gain practical experience through **hands-on projects** and real-world applications, mastering essential Django principles and techniques. Whether you're just starting or looking to refine your skills, this course will empower you to build sophisticated web applications efficiently. Ready to enhance your web development journey? Enroll now and unlock your potential with Django!

N Nave...



24

[Previous Article](#)

[Next Article](#)

Build a Contact form Using Django, React
and tailwind

Similar Reads

How to automatically create API Documentation in Django REST Framework?

Prerequisite - <https://www.geeksforgeeks.org/how-to-create-an-app-in-django/>
Writing documentation can be pretty daunting for developers and to be honest,...

3 min read

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

Basic Authentication - Django REST Framework

Authentication is a mechanism that provides access control based on the credentials associated with incoming requests. Django REST Framework provide...

4 min read

How to throttle API with Django Rest Framework

Have you ever passed through an overbridge and noticed the sign stating the bridge weight restriction? Those restrictions are put in place to ensure public...

4 min read

Adding Permission in API - Django REST Framework

There are many different scenarios to consider when it comes to access control. Allowing unauthorized access to risky operations or restricted areas results in a...

7 min read

Browsable API in Django REST Framework

The browsable API feature in the Django REST framework generates HTML output for different resources. It facilitates interaction with RESTful web service...

8 min read

Implement Token Authentication using Django REST Framework

Token authentication refers to exchanging username and password for a token that will be used in all subsequent requests so to identify the user on the server...

2 min read

Creating and Using Serializers - Django REST Framework

In Django REST Framework the very concept of Serializing is to convert DB data to a datatype that can be used by javascript. Serializers allow complex data suc...

3 min read

OTP Verification in Django REST Framework using JWT and Cryptography

In this article, we'll explore how to implement OTP verification in Django REST Framework using JWT and cryptography. We will guide you through the proces...

6 min read

Django REST Framework Installation

Django REST Framework can be installed via pip package similar to Django installation. Since Django REST Framework is a wrapper over default Django...

1 min read

Article Tags : [Python](#) [Django-REST](#) [Python Django](#)

Practice Tags : [python](#)



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
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

Languages

Python
Java
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

System Design

High Level Design

Low Level Design

UML Diagrams

Interview Guide

Design Patterns

OOAD

System Design Bootcamp

Interview Questions

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

Interview Preparation

Competitive Programming

Top DS or Algo for CP

Company-Wise Recruitment Process

Company-Wise Preparation

Aptitude Preparation

Puzzles

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved