Python Basics Interview Questions Python Quiz Popular Packages Python Projects Practice Python AI Wit

Implement Token Authentication using Django REST Framework

Last Updated : 19 Nov, 2021

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 side.This article revolves about implementing token authentication using Django REST Framework to make an API. The token authentication works by providing token in exchange for exchanging usernames and passwords.

Modules required :

- **django :**

```
pip install django
```

- **crispy_forms :**

```
 pip install --upgrade django-crispy-forms
```

- **django rest_framework :**

```
pip install djangorestframework
```

- **HTTPie :**

```
pip install httpie
```

and a project to add API, here we are using Sign Up and log in

Creating Viewset and Serializers

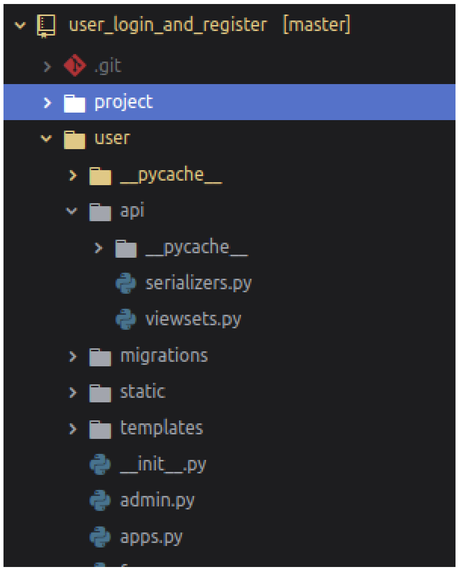
Go to ***user*** folder in given project

and make a api folder to keep all api related files

```
cd user && mkdir api
```

Now, make *user/api/serializers.py* and *user/api/viewsets.py* in api folder

```
cd user/api && touch serializers.py viewsets.py
```



now edit *user/api/serializers.py*

Python3

```python
from rest_framework import serializers
from django.contrib.auth.models import User


class userSerializers(serializers.ModelSerializer):

    class Meta:
        model = User
        fields =  '__all__'
```

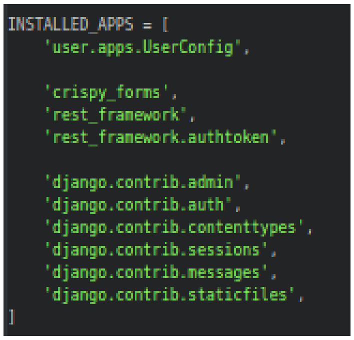
also edit *user/api/viewsets.py*

Python3

```
from rest_framework import viewsets
from .serializers import userSerializers
from django.contrib.auth.models import User


class userviewsets(viewsets.ModelViewSet):
    queryset = User.objects.all()
    serializer_class = userSerializers
```

Edit settings.py

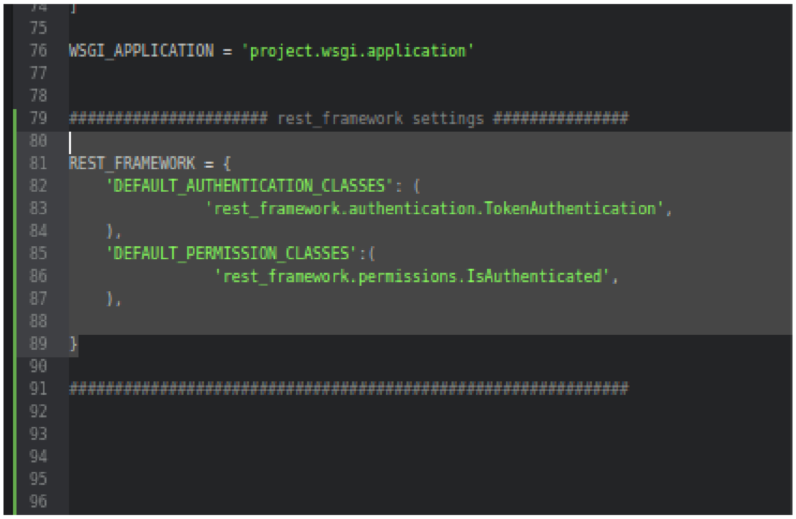
add rest_framework and rest_framework.authtoken in INSTALLED_APPS in setting.py



```
INSTALLED_APPS = [
    'user.apps.UserConfig',

    'crispy_forms',
    'rest_framework',
    'rest_framework.authtoken',

    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

Edit rest_framework settings as below

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
                'rest_framework.authentication.TokenAuthentication',
    ),
    'DEFAULT_PERMISSION_CLASSES':(
                'rest_framework.permissions.IsAuthenticated',
    ),


}
```



```
]

WSGI_APPLICATION = 'project.wsgi.application'


###################### rest_framework settings ###############

REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
                'rest_framework.authentication.TokenAuthentication',
    ),
    'DEFAULT_PERMISSION_CLASSES':(
                 'rest_framework.permissions.IsAuthenticated',
    ),

}

################################################################
```

Creating router

goto *project/* and create router.py

```
cd project/ && touch router.py
```

edit *project/router.py*,

Python3

```python
from user.api.viewsets import userviewsets
from rest_framework import routers

router = routers.DefaultRouter()
router.register('user', userviewsets, base_name ='user_api')
```

Editing url.py

goto to *project/urls.py*

and edit it

Import router and rest_framework.authtoken for token authentication

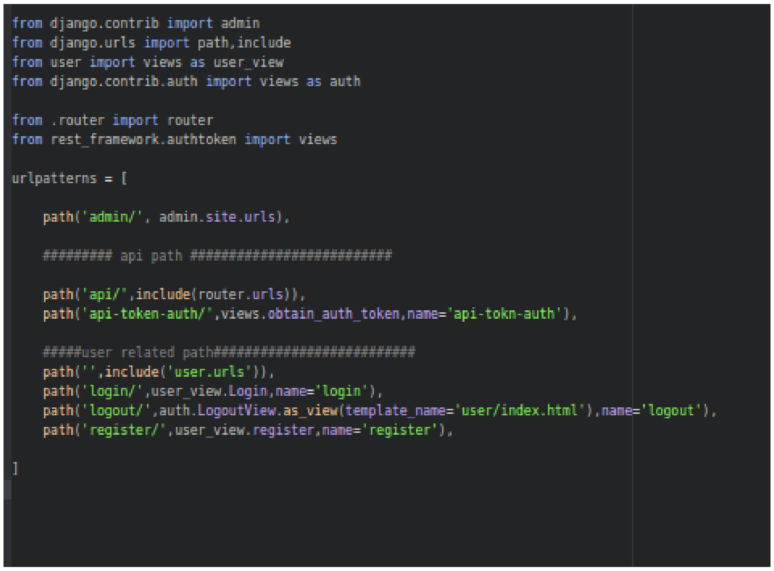
Python3

```python
from .router import router
from rest_framework.authtoken import views
```

add API related paths

Python3

```
path('api/', include(router.urls)),
path('api-token-auth/', views.obtain_auth_token, name='api-token-auth'),
```



```
from django.contrib import admin
from django.urls import path,include
from user import views as user_view
from django.contrib.auth import views as auth

from .router import router
from rest_framework.authtoken import views

urlpatterns = [

    path('admin/', admin.site.urls),

    ######### api path ##########################

    path('api/',include(router.urls)),
    path('api-token-auth/',views.obtain_auth_token,name='api-tokn-auth'),

    #####user related path##########################
    path('',include('user.urls')),
    path('login/',user_view.Login,name='login'),
    path('logout/',auth.LogoutView.as_view(template_name='user/index.html'),name='logout'),
    path('register/',user_view.register,name='register'),

]
```

Testing API

first, migrate models

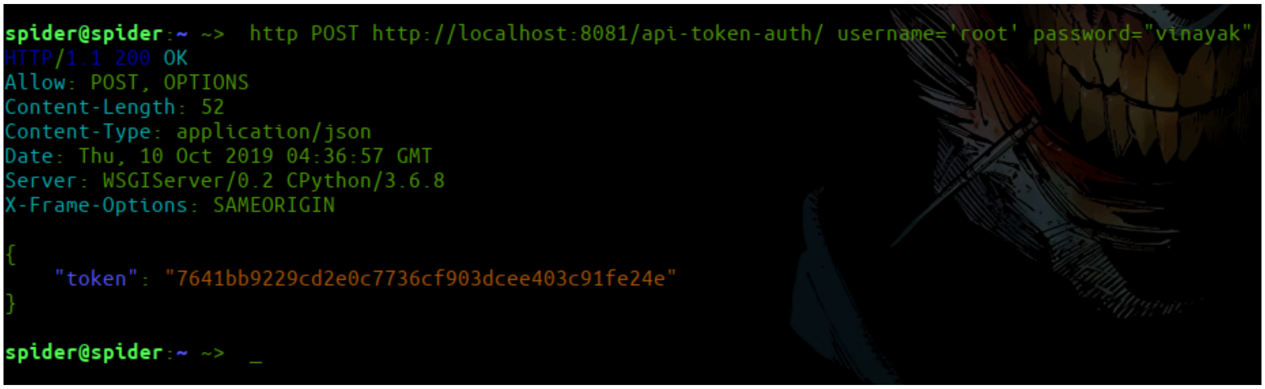
```
python manage.py migrate
```

start server using below command

```
python manage.py runserver
```

open another terminal and let us check our API using HTTP POST request for a token and paste username and password.

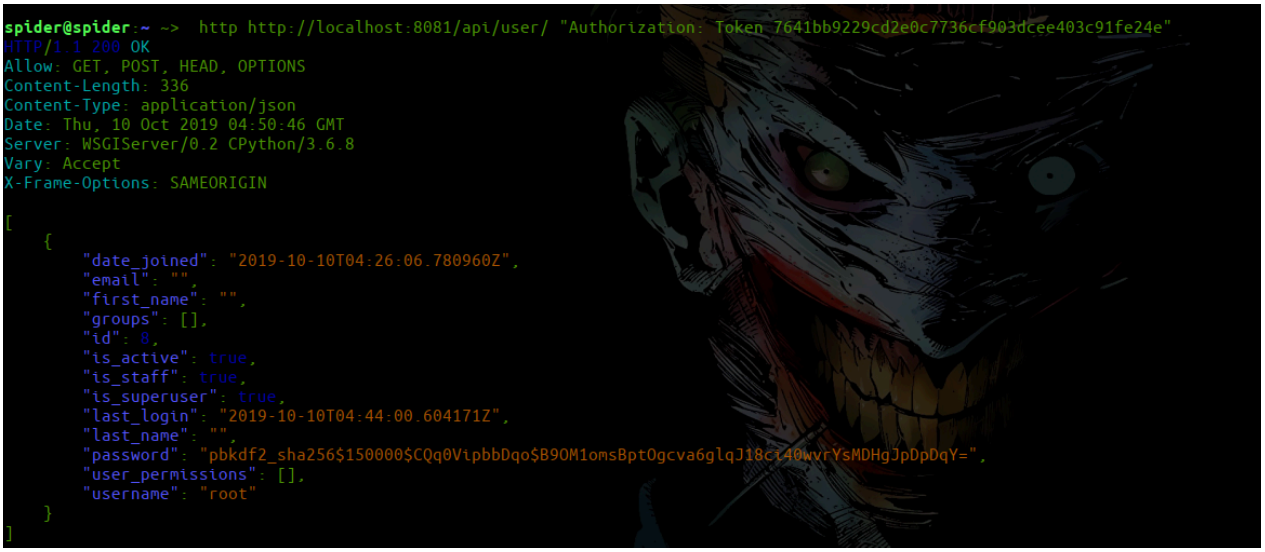
```
http POST http://localhost:8081/api-token-auth/ username='your_username'
password="your_password"
```



```
spider@spider:~ ~>  http POST http://localhost:8081/api-token-auth/ username='root' password="vinayak"
HTTP/1.1 200 OK
Allow: POST, OPTIONS
Content-Length: 52
Content-Type: application/json
Date: Thu, 10 Oct 2019 04:36:57 GMT
Server: WSGIServer/0.2 CPython/3.6.8
X-Frame-Options: SAMEORIGIN

{
    "token": "7641bb9229cd2e0c7736cf903dcee403c91fe24e"
}

spider@spider:~ ~>  _
```

now use this token to get data from API, place your API token

```
http http://localhost:8081/api/user/ "Authorization: Token API_KEY_HERE"
```



```
spider@spider:~ ~>  http http://localhost:8081/api/user/ "Authorization: Token 7641bb9229cd2e0c7736cf903dcee403c91fe24e"
HTTP/1.1 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Length: 336
Content-Type: application/json
Date: Thu, 10 Oct 2019 04:50:46 GMT
Server: WSGIServer/0.2 CPython/3.6.8
Vary: Accept
X-Frame-Options: SAMEORIGIN

[
    {
        "date_joined": "2019-10-10T04:26:06.780960Z",
        "email": "",
        "first_name": "",
        "groups": [],
        "id": 8,
        "is_active": true,
        "is_staff": true,
        "is_superuser": true,
        "last_login": "2019-10-10T04:44:00.604171Z",
        "last_name": "",
        "password": "pbkdf2_sha256$150000$CQq0VipbbDqo$B9OM1omsBptOgcva6glqJ18ci40wvrYsMDHgJpDpDqY=",
        "user_permissions": [],
        "username": "root"
    }
]
```

itsvin... 11

Previous Article

How to throttle API with Django Rest Framework

Next Article

Similar Reads

JWT Authentication with Django REST Framework

JSON Web Token is an open standard for securely transferring data within parties using a JSON object. JWT is used for stateless authentication mechanisms for...

2 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

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

Token Authentication in Django Channels and Websockets

Prerequisites: Django, WebSockets, Django channels, Token authentication The most popular Django topics right now are WebSockets and Django channels...

13 min read

How to Create a basic API using Django Rest Framework ?

Django REST Framework is a wrapper over the default Django Framework, basically used to create APIs of various kinds. There are three stages before...

4 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

Boolean Fields in 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. Every serializer comes with some...

4 min read

String Fields in 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. Every serializer comes with some...

5 min read

Article Tags : Python Technical Scripter

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

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

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
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects