



Youtube video downloader using Django

Last Updated : 25 May, 2022

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

Django. We will be using **pytube** module for that.

Prerequisite:

- **pytube**: It is python's lightweight and dependency-free module, which is used to download YouTube Videos.
- **Django**: It is python's framework to make web-applications.

Here, we will be using Django as a backend along with pytube module to create this tool. We can install pytube module by typing the below command in the terminal.

```
pip install pytube
```

So, let's dive in to make our YouTube video downloader tool.

First, we will create an HTML design (form) where the user can come and enter the URL of a video which he/she wants to download. We will use Django's **POST** method to get that URL (because it is secure). We also need to use csrf token if we are using the POST method. Syntax for csrf token is:

```
{% csrf_token %}
```

HTML

```
<!DOCTYPE html>
<html>
<body>

<h1>Youtube video downloader</h1>
```

```
<form action="" method="post">
    {% csrf_token %}

    <label for="link">Enter URL:</label>
    <input type="text" id="link" name="link"><br><br>
    <input type="submit" value="Submit">
</form>

</body>
</html>
```

Now, it's time to create a function that receives the video link and downloads that video. You need to import function YouTube from module pytube in views.py file. Now we can define the function to download video.

views.py

Python3

```
# importing all the required modules
from django.shortcuts import render, redirect
from pytube import *

# defining function
def youtube(request):

    # checking whether request.method is post or not
    if request.method == 'POST':

        # getting link from frontend
        link = request.POST['link']
        video = YouTube(link)

        # setting video resolution
        stream = video.streams.get_lowest_resolution()

        # downloads video
```

```
stream.download()

# returning HTML page
return render(request, 'youtube.html')
return render(request, 'youtube.html')
```

Now, we have to define the URL (path) for this function inside urls.py.

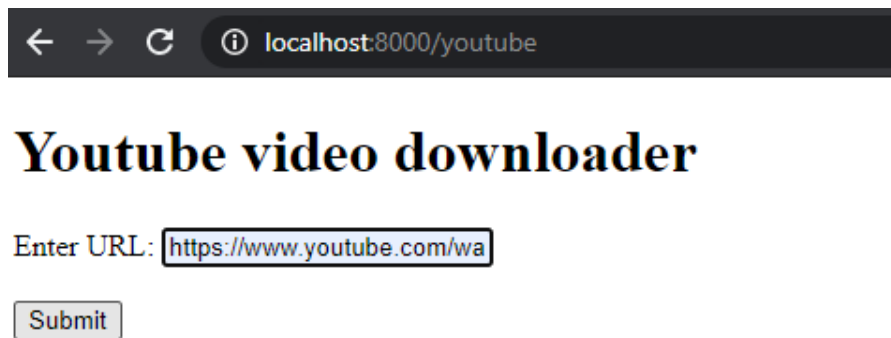
Python3

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('youtube', views.youtube, name='youtube'),
]
```

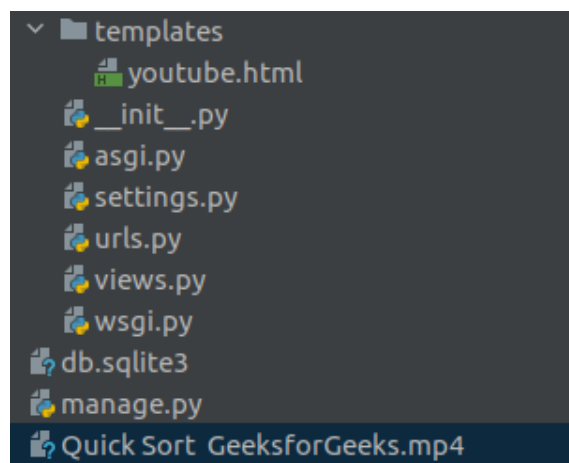
That is it for the coding part, now you can run the project by **python manage.py runserver** and head over to **http://localhost:8000/youtube** to see the output.

Output:



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/youtube'. The page title is 'Youtube video downloader'. Below the title, there is a form with the label 'Enter URL:' followed by a text input field containing 'https://www.youtube.com/wa'. Below the input field is a 'Submit' button.

When you click on submit a video will be downloaded in your Django project's directory.



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!

R romil...



9

Previous Article

How to add Google reCAPTCHA to Django forms ?

Next Article

E-commerce Website using Django

Similar Reads

GUI to get views, likes, and title of a YouTube video using YouTube API in...

Prerequisite: YouTube API, Tkinter Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, Tkinter is the most...

2 min read

How to Extract YouTube Comments Using Youtube API - Python

Prerequisite: YouTube API Google provides a large set of API's for the developer to choose from. Each and every service provided by Google has an associated...

2 min read

Python - Obtain title, views and likes of YouTube video using BeautifulSoup

In this article, we will learn how can we obtain data (like title, views, likes, dislikes etc) from any YouTube video using a Python script. For this task, we are going to...

2 min read

Create GUI for Downloading Youtube Video using Python

Prerequisites: Python GUI – tkinter YouTube is a very popular video-sharing website. Downloading a video's/playlist from YouTube is a tedious task....

12 min read

Playing Youtube Video using Python

In this article, we will see how we can play youtube video in python. In order to play youtube videos in python we need pafy and vlc module. Pafy is a Python...

2 min read

Download Youtube Video(s) or whole Playlist with Python

Prerequisite: Tkinterpyyoutubepytubethreading in TkinterYouTube API Python offers multiple options for developing GUI (Graphical User Interface). Out of all...

6 min read

Extract Video Titles with the Youtube API in Python

Extracting video titles from YouTube can be incredibly useful for various applications, such as data analysis, content creation, and trend tracking. The...

4 min read

Adding Tags Using Django-Taggit in Django Project

Django-Taggit is a Django application which is used to add tags to blogs, articles etc. It makes very easy for us to make adding the tags functionality to our django...

2 min read

How to customize Django forms using Django Widget Tweaks ?

Django forms are a great feature to create usable forms with just few lines of code. But Django doesn't easily let us edit the form for good designs. Here, we...

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

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

Languages

Python
Java
C++
PHP
GoLang
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

DevOps

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

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

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

GeeksforGeeks Videos

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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved