



[Flask Templates](#) [Jinja2](#) [Flask-REST API](#) [Python SQLAlchemy](#) [Flask Bcrypt](#) [Flask Cookies](#) [Json](#) [Postman](#)

How to Upload File in Python-Flask

Last Updated : 07 Nov, 2022

File uploading is a typical task in web apps. Taking care of file upload in Flask is simple all we need is to have an HTML form with the encryption set to multipart/form information to publish the file into the URL. The server-side flask script brings the file from the request object utilizing the request.files[] Object. On effectively uploading the file, it is saved to the ideal location on the server.

Install the [Flask](#) by writing the command in terminal:

```
pip install flask
```

Stepwise Implementation

Step 1: A new folder “file uploading” should be created. Create the folders “templates” and “main.py” in that folder, which will store our HTML files and serve as the location for our Python code.

Step 2: For the front end, we must first develop an HTML file where the user can select a file and upload it by clicking the upload buttons. The user will click the submit button after choosing the file from their local computer in order to transmit it to the server.

Index.html

HTML

```
<html>
<head>
  <title>upload the file : GFG</title>
</head>
<body>
  <form action = "/success" method = "post" enctype="multipart/form-data">
    <input type="file" name="file" />
    <input type = "submit" value="Upload">
```

```
</form>
</body>
</html>
```

Step 3: We must make another HTML file just for acknowledgment. Create a file inside the templates folder called “Acknowledgement.html” to do this. This will only be triggered if the file upload went smoothly. Here, the user will receive a confirmation.

Acknowledgement.html

HTML

```
<html>
  <head>
    <title>success</title>
  </head>
  <body>
    <p>File uploaded successfully</p>
    <p>File Name: {{name}}</p>
  </body>
</html>
```

Step 4: Now inside the ‘main.py’ write the following codes. The name of the objective file can be obtained by using the following code and then we will save the uploaded file to the root directory.

main.py

Python3

```
from distutils.log import debug
from fileinput import filename
from flask import *
app = Flask(__name__)

@app.route('/')
def main():
    return render_template("index.html")
```

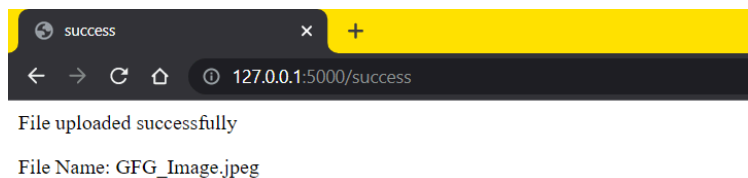
```
@app.route('/success', methods = ['POST'])
def success():
    if request.method == 'POST':
        f = request.files['file']
        f.save(f.filename)
        return render_template("Acknowledgement.html", name = f.filename)

if __name__ == '__main__':
    app.run(debug=True)
```

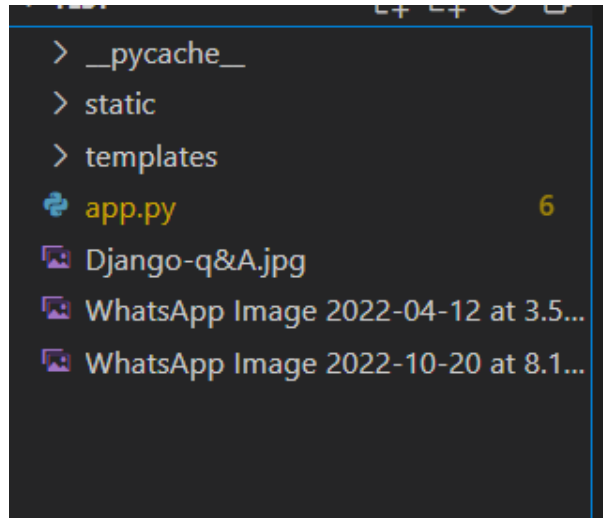
Output:

Run the following command in your terminal.

```
python main.py
```



Step 5: Now, to check if it is correctly working or not go to the folder where 'main.py'. Check in that folder you will find the files there.



File structure



swar...



3

Previous Article

[Uploading and Downloading Files in Flask](#)

Next Article

[Upload Multiple files with Flask](#)

Similar Reads

Upload and Read Excel File in Flask

In this article, we will look at how to read an Excel file in Flask. We will use the Python Pandas library to parse this excel data as HTML to make our job easier....

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

Upload Multiple files with Flask

In online apps, uploading files is a common task. Simple HTML forms with encryption set to multipart/form information are all that is required to publish a...

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

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

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

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

Upload file and read its content in cherrypy python

CherryPy is a web framework of Python which provides a friendly interface to the HTTP protocol for Python developers. It is also called a web application library. I...

3 min read

Upload File With Selenium in Python

Uploading files using Selenium in Python allows developers to automate the process of interacting with file input elements on web pages. Leveraging the...

2 min read

File upload 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](#)[Python Flask](#)[Technical Scripter 2022](#)**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](#)

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](#)

[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](#)

School Subjects

[Mathematics](#)[Physics](#)[Chemistry](#)[Biology](#)[Social Science](#)[English Grammar](#)[Commerce](#)[World GK](#)[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](#)

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](#)