



---

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

# Flask App Routing

Last Updated : 10 Mar, 2023

---

**App Routing** means mapping the URLs to a specific function that will handle the logic for that URL. Modern web frameworks use more meaningful URLs to help users remember the URLs and make navigation simpler.

**Example:** In our application, the URL ("/") is associated with the root URL. So if our site's domain was `www.example.org` and we want to add routing to "`www.example.org/hello`", we would use `/hello`".

To bind a function to an URL path we use the *app.route* decorator. In the below example, we have implemented the above routing in the flask.

---

## main.py

```
from flask import Flask

app = Flask(__name__)

# Pass the required route to the decorator.
@app.route("/hello")
def hello():
    return "Hello, Welcome to GeeksForGeeks"

@app.route("/")
def index():
    return "Homepage of GeeksForGeeks"

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

The hello function is now mapped with the `/hello` path and we get the output of the function rendered on the browser.

**Step to run the application:** Run the application using the following command.

```
python main.py
```

**Output:** Open the browser and visit *127.0.0.1:5000/hello*, you will see the following output.



**Dynamic URLs** – We can also build dynamic URLs by using variables in the URL. To add variables to URLs, use `<variable_name>` rule. The function then receives the `<variable_name>` as keyword argument.

**Example:** Consider the following example to demonstrate the dynamic URLs.

---

## main.py

```
from flask import Flask

app = Flask(__name__)

@app.route('/user/<username>')
def show_user(username):
    # Greet the user
    return f'Hello {username} !'

# Pass the required route to the decorator.
@app.route("/hello")
def hello():
    return "Hello, Welcome to GeeksForGeeks"

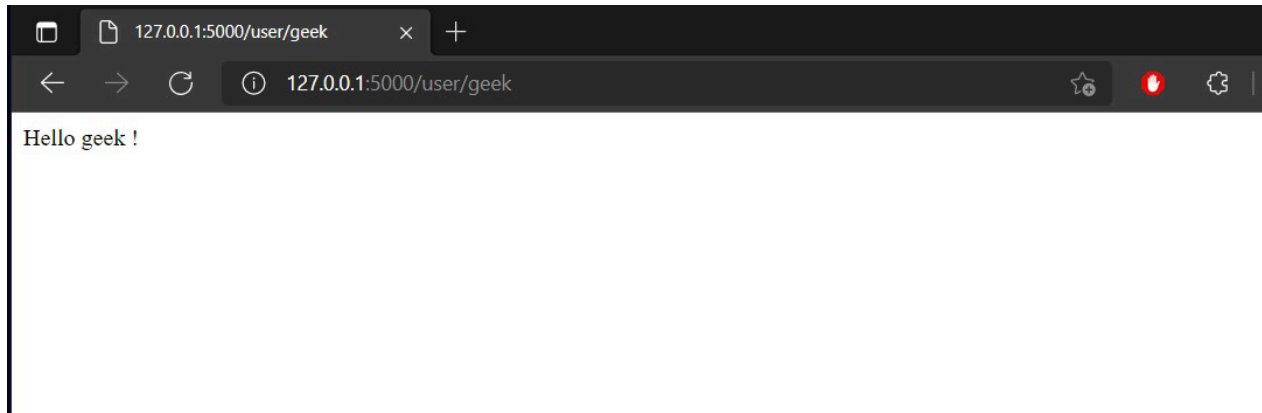
@app.route("/")
def index():
    return "Homepage of GeeksForGeeks"

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

**Step to run the application:** Run the application using the following command.

```
python main.py
```

**Output:** Open the browser and visit `127.0.0.1:5000/user/geek`, you will see the following output.



Additionally, we can also use a converter to convert the variable to a specific data type. By default, it is set to string values. To convert use `<converter:variable_name>` and following converter types are supported.

- **string:** It is the default type and it accepts any text without a slash.
- **int:** It accepts positive integers.
- **float:** It accepts positive floating-point values.
- **path:** It is like a string but also accepts slashes.
- **uuid:** It accepts UUID strings.

**Example:** Consider the following example to demonstrate the converter type.

---

## main.py

```
from flask import Flask

app = Flask(__name__)

@app.route('/post/<int:id>')
def show_post(id):
    # Shows the post with given id.
    return f'This post has the id {id}'

@app.route('/user/<username>')
def show_user(username):
```

```
# Greet the user
return f'Hello {username} !'

# Pass the required route to the decorator.
@app.route("/hello")
def hello():
    return "Hello, Welcome to GeeksForGeeks"

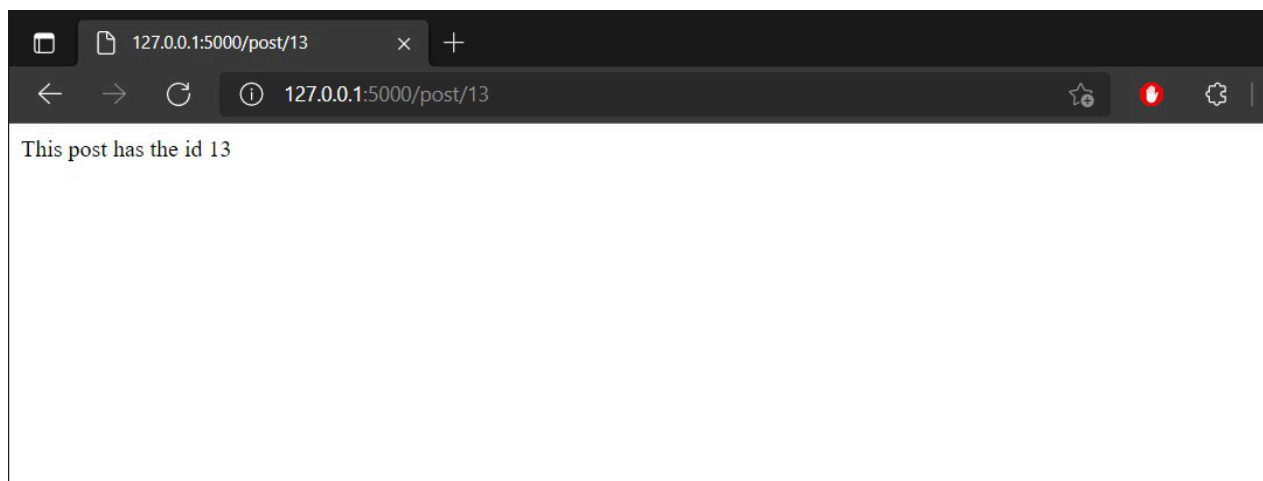
@app.route("/")
def index():
    return "Homepage of GeeksForGeeks"

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

**Step to run the application:** Run the application using the following command.

```
python main.py
```

**Output:** Open the browser and visit `127.0.0.1:5000/post/13`, you will see the following output.



**The `add_url_rule()` function** – The URL mapping can also be done using the `add_url_rule()` function. This approach is mainly used in case we are importing the view function from another module. In fact, the `app.route` calls this function internally.

**Syntax:**

```
add_url_rule(<url rule>, <endpoint>, <view function>)
```

**Example:** In the below example, we will try to map the `show_user` view function using this approach.

---

## main.py

```
from flask import Flask

app = Flask(__name__)

def show_user(username):
    # Greet the user
    return f'Hello {username} !'

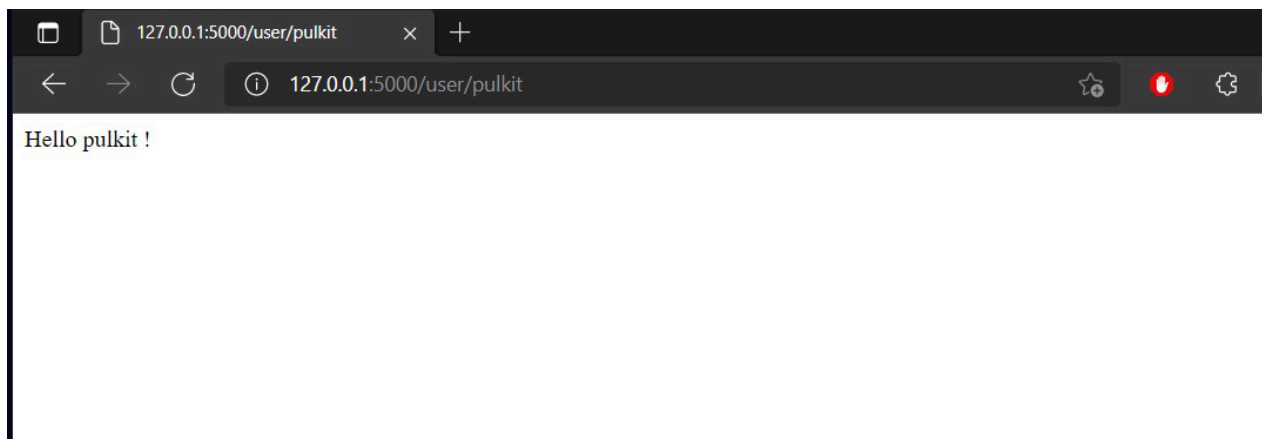
app.add_url_rule('/user/<username>', 'show_user', show_user)

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

**Step to run the application:** Run the application using the following command.

```
python main.py
```

**Output:** Open the browser and visit `127.0.0.1:5000/user/pulkit`, you will see the following output.



Looking to dive into the world of programming or sharpen your Python skills? Our [Master Python: Complete Beginner to Advanced Course](#) is your ultimate guide to becoming proficient in Python. This course covers everything you need to build a solid foundation from fundamental programming concepts to

advanced techniques. With **hands-on projects**, real-world examples, and expert guidance, you'll gain the confidence to tackle complex **coding challenges**. Whether you're starting from scratch or aiming to enhance your skills, this course is the perfect fit. Enroll now and master Python, the language of the future!

P pans...



14

### Previous Article

[How to Run a Flask Application](#)

### Next Article

[Flask - HTTP Method](#)

## Similar Reads

### What is Routing and Nested Routing in Angular 9/8 ?

In this article, we will learn the routing & nested routing concept in Angular. We will implement the concept to establish routing between different components ...

[5 min read](#)

### Mastering React Routing: Learn Navigation and Routing in React Apps

In this article, we will learn how to implement Routing in React in an efficient way by implementing various aspects of routing such as dynamic routing,...

[6 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 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](#)

### 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

## How to Build a Simple Android App with Flask Backend?

Flask is an API of Python that allows us to build up web applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's...

8 min read

## Create a Weather app using Flask | Python

Prerequisite : Flask installation Flask is a lightweight framework written in Python. It is lightweight because it does not require particular tools or libraries...

2 min read

## Todo list app using Flask | Python

There are many frameworks that allow building your webpage using Python, like Django, flask, etc. Flask is a web application framework written in Python. Flask...

3 min read

**Article Tags :** [Python](#) [Web Technologies](#) [Python Flask](#)

**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

## System Design

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns

## 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

## Interview Preparation

Competitive Programming  
Top DS or Algo for CP  
Company-Wise Recruitment Process  
Company-Wise Preparation  
Aptitude Preparation

OOAD

Puzzles

System Design Bootcamp

Interview Questions

### 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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved