Python Flask – Redirect and Errors

Last Updated : 07 Nov, 2022

We'll discuss redirects and errors with Python Flask in this article. A redirect is used in the Flask class to send the user to a particular URL with the status code. conversely, this status code additionally identifies the issue. When we access a website, our browser sends a request to the server, and the server replies with what is known as the HTTP status code, which is a three-digit number, The different reasons for errors are:

- Unauthorized access or poor request.
- Unsupported media file types.
- Overload of the backend server.
- Internal hardware/connection error.

Syntax of Redirect

> ***flask.redirect(location,code=302)***
>
> ***Parameters:***
>
> - ***location(str):*** *the location which URL directs to.*
> - ***code(int):*** *The status code for Redirect.*
> - ***Code:*** *The default code is 302 which means that the move is only temporary.*
>
> ***Return:*** *The response object and redirects the user to another target location with the specified code.*

The different types of HTTP codes are:

Code	Status
300	Multiple_choices
301	Moved_permanently
302	Found
303	See_other
304	Not_modified
305	Use_proxy
306	Reserved
307	Temporary_redirect

Import the redirect attribute

We can redirect the URL using Flask by importing **redirect.** Let's see how to import redirects from Flask

Python3

```
from flask import redirect
```

Example:

Let's take a simple example to redirect the URL using a flask named **app.py**.

Python3

```
# importing redirect
from flask import Flask, redirect, url_for, render_template, request
```

```
# Initialize the flask application
app = Flask(__name__)

# It will load the form template which
# is in login.html
@app.route('/')
def index():
    return render_template("login.html")


@app.route('/success')
def success():
    return "logged in successfully"

# loggnig to the form with method POST or GET
@app.route("/login", methods=["POST", "GET"])
def login():

    # if the method is POST and Username is admin then
    # it will redirects to success url.
    if request.method == "POST" and request.form["username"] == "admin":
        return redirect(url_for("success"))

    # if the method is GET or username is not admin,
    # then it redirects to index method.
    return redirect(url_for('index'))


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

login.html

Create a folder named templates. In the templates, creates a templates/**login.html** file. On this page, we will take input from the user of its username.

HTML

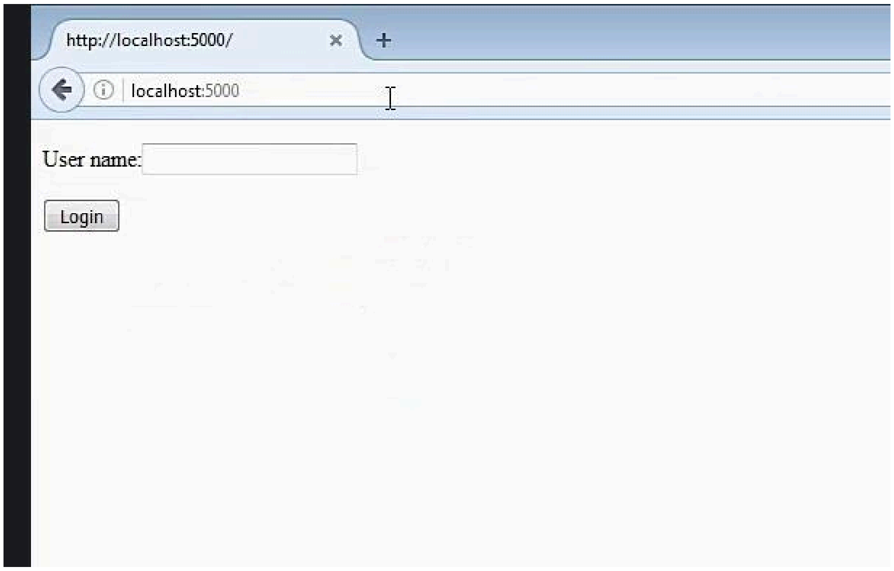
```
<!DOCTYPE html>
<html lang="en">
   <body>
      <form method="POST", action="\login">
         Username: <input  name=username type=text></input><br>
         <button type="submit">Login</button>
      </form>
```

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

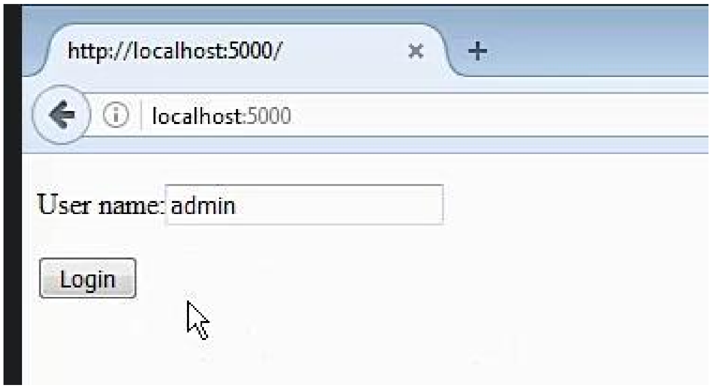
Now we will deploy our code.

```
python app.py
```

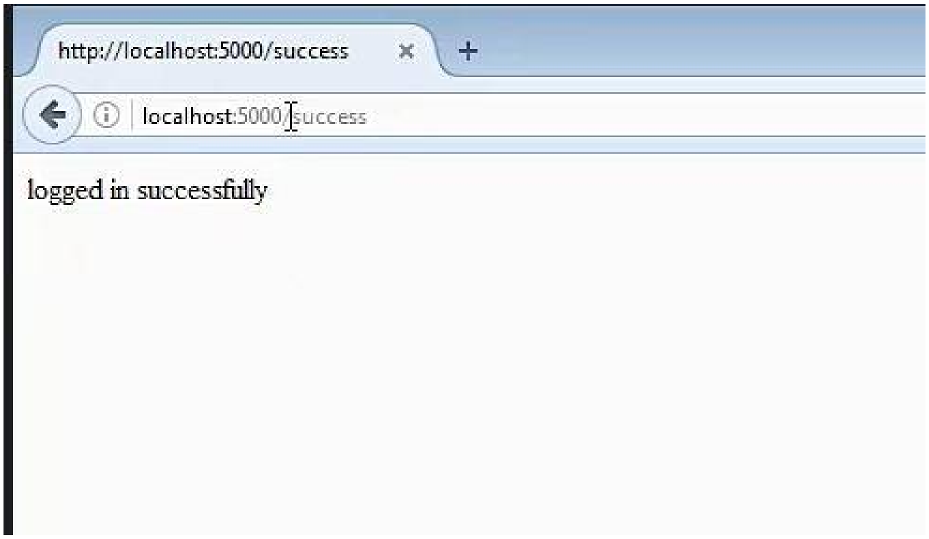
Output:



Case 1: If the Username is **admin** and the method is **POST** then it will redirect to the success URL and display logged in successfully.

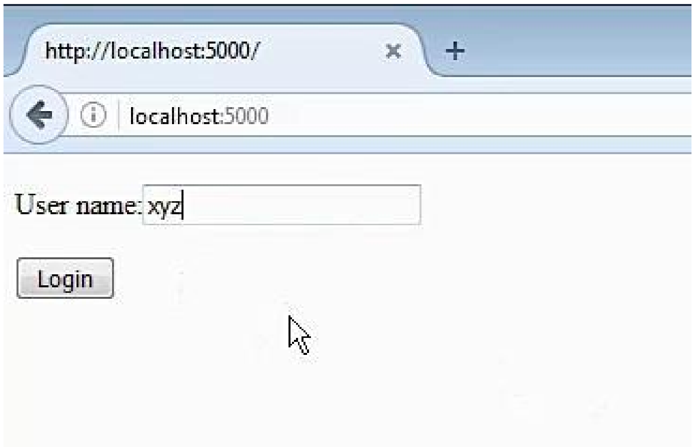


User name: admin



logged in successfully

Case 2: In the other case, if the Username is something like xyz or anything then it will redirect to the **index** method i.e., it will load the form again until the username is admin.



User name: xyz

Flasks Errors

If there is an error in the address or if there is no such URL then Flask has an **abort()** function used to exit with an error code.

Syntax of abort() method

> ***Syntax:*** *abort(code, message)*

- ***code:*** *int, The code parameter takes any of the following values*
- ***message:*** *str, create your custom message Error.*

The different types of errors we can abort in the application in your Flask.

Code	Error
400	Bad request
401	Unauthenticated
403	Forbidden
404	Not Found
406	Not Acceptable
415	Unsupported Media Type
429	Too Many Requests

Example to demonstrate abort

In this example, if the username starts with a number then an error code message will through, else on success “Good username” will be printed.

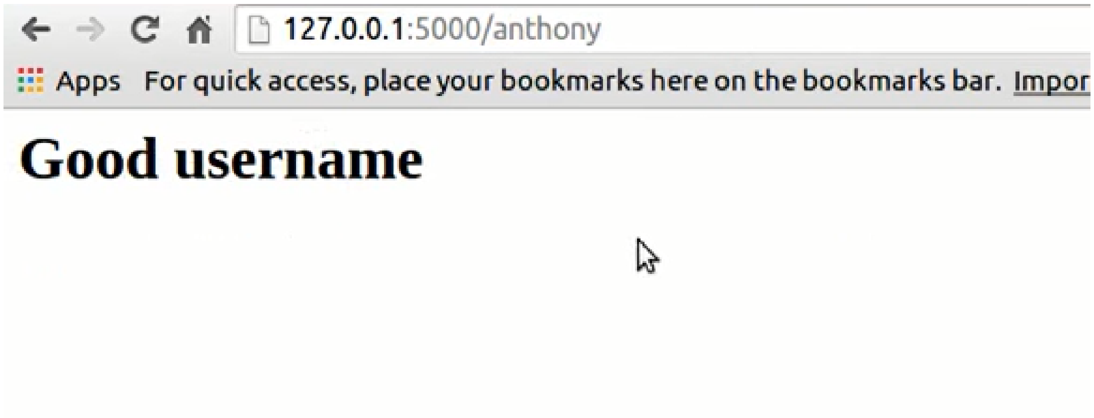
Example 1:

Python3

```
# importing abort
from flask import Flask, abort

# Initialize the flask application
app = Flask(__name__)


@app.route('/<uname>')
def index(uname):
    if uname[0].isdigit():
        abort(400)
    return '<h1>Good Username</h1>'


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

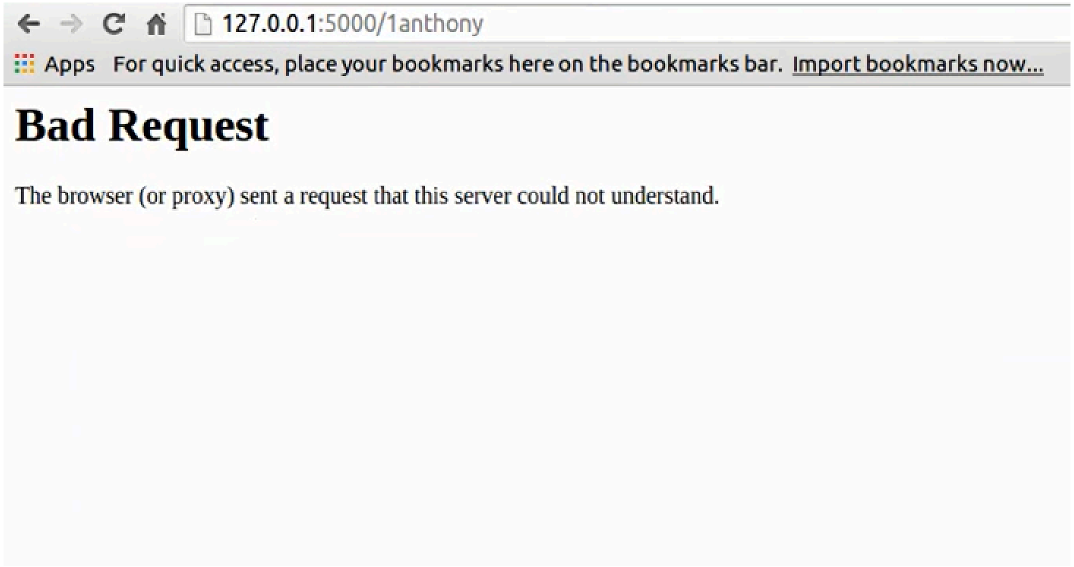
Output:

Case 1: If the username doesn't start with a number.


127.0.0.1:5000/anthony
Apps
For quick access, place your bookmarks here on the bookmarks bar. Impor

Good username

Case 2: If the username starts with a number.


127.0.0.1:5000/1anthony
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Bad Request

The browser (or proxy) sent a request that this server could not understand.

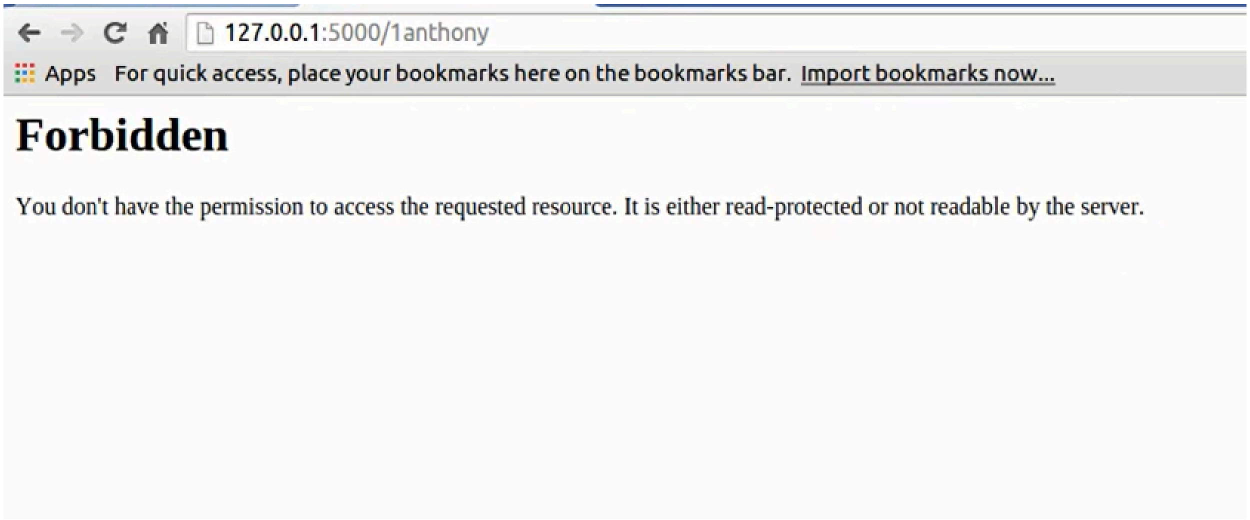
Example 2:

In the above Python code, the status code to the abort() is **400**, so it raises a Bad Request Error. Let's try to change the code to **403**. If the username starts with a number then it raises a Forbidden Error.

Python3

```
# importing abort
from flask import Flask, abort

# Initialize the flask application
app = Flask(__name__)


@app.route('/<uname>')
def index(uname):
    if uname[0].isdigit():
        abort(403)
    return '<h1>Good Username</h1>'


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

Output:



127.0.0.1:5000/1anthony
Apps For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...

Forbidden

You don't have the permission to access the requested resource. It is either read-protected or not readable by the server.

bhar...

5

Previous Article

Redirecting to URL in Flask

Next Article

How to Change Port in Flask app

Similar Reads

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

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

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

Django: Redirect to Previous Page After Login

Handling user navigation smoothly is crucial for a good user experience in web applications. One common requirement is to redirect users back to their previous...

4 min read

Django return redirect() with Parameters

When building web applications with Django, it's common to need to redirect users to different pages, often passing parameters to specify where and with...

3 min read

How to Configure Where to Redirect After a Log Out in Django?

One common requirement in many Django applications is redirecting users to a specific page after they log out. By default, Django redirects users to the login...

6 min read

Create GitHub API to fetch user profile image and number of repositories…

GitHub is where developers shape the future of software, together, contribute to the open-source community, manage Git repositories, etc. It is one of the most...

5 min read

Errors and Exceptions in Python

Errors are problems in a program that causes the program to stop its execution. On the other hand, exceptions are raised when some internal events change the...

4 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

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