



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

Flask Cookies

Last Updated : 05 Feb, 2023

Flask is a lightweight, web development framework built using python language. Generally, for building websites we use HTML, CSS and JavaScript but in flask the python scripting language is used for developing the web-applications.

To know more about Flask and how to run an application in Flask: [Flask – First Application Creation](#)

What are Cookies?

Technically, cookies track user activity to save user information in the browser as key-value pairs, which can then be accessed whenever necessary by the developers to make a website easier to use. These enhances the personal user experience on a particular website by remembering your logins, your preferences and much more.

For running a flask application, we need to have some prerequisites like installing the flask.

Prerequisites:

Use the upgraded version of pip by below command in your terminal. In this article, I am using Visual Studio Code to run my flask applications.

```
Python -m pip install -upgrade pip
```

```
Python -m pip install flask
```

Setting Cookies in Flask:

set_cookie() method: Using this method we can generate cookies in any application code. The syntax for this cookies setting method:

```
Response.set_cookie(key, value = '', max_age = None, expires = None,  
path = '/', domain = None,  
                    secure = None, httponly = False)
```

Parameters:

- key – Name of the cookie to be set.
- value – Value of the cookie to be set.
- max_age – should be a few seconds, None (default) if the cookie should last as long as the client's browser session.
- expires – should be a datetime object or UNIX timestamp.
- domain – To set a cross-domain cookie.
- path – limits the cookie to given path, (default) it will span the whole domain.

Example:

Python3

```
from flask import Flask, request, make_response  
  
app = Flask(__name__)  
  
# Using set_cookie( ) method to set the key-value pairs below.  
@app.route('/setcookie')  
def setcookie():  
  
    # Initializing response object  
    resp = make_response('Setting the cookie')  
    resp.set_cookie('GFG', 'ComputerScience Portal')  
    return resp  
  
app.run()
```

Running the code in Visual Studio Code application.

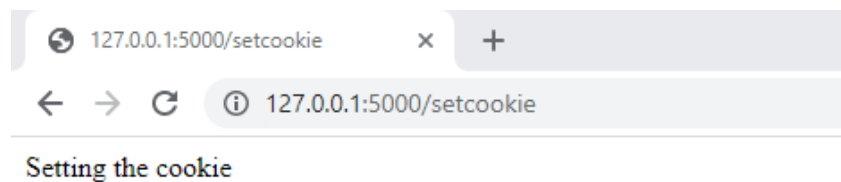
```

C:\Users\admin\Desktop>python flask-cookies.py
* Serving Flask app "flask-cookies" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [15/Dec/2022 22:10:51] "GET /setcookie HTTP/1.1" 200 -

```

My Visual Studio Code terminal

Output: Go to the above-mentioned url in the terminal -For Example – <http://127.0.0.1:5000/setcookie>. Here the route-name is setcookie.



Output

Getting Cookies in Flask:

cookies.get()

This get() method retrieves the cookie value stored from the user's web browser through the request object.

Python3

```

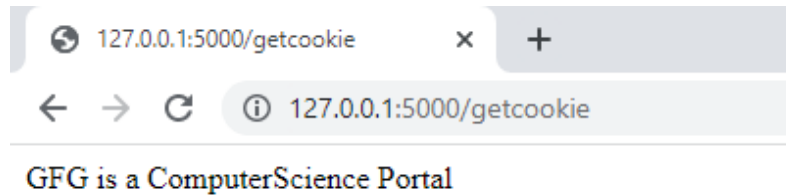
from flask import Flask, request, make_response
app = Flask(__name__)

# getting cookie from the previous set_cookie code
@app.route('/getcookie')
def getcookie():

```

```
GFG = request.cookies.get('GFG')  
return 'GFG is a '+ GFG  
  
app.run()
```

Output:



Login Application in Flask using cookies

Let's develop a simple login page with Flask using cookies. First, we are creating the main python file – app.py in our code editor. Next, we will create the UI of our web page which is Login.html where the user can enter his username and password. In this app.py, we are storing the username as cookie to know which user logged in to the website. In the below code, we are requesting the stored cookie from the browser and displaying it on the next page which routes to user details page.

app.py

Python3

```
from flask import Flask, request, make_response, render_template  
  
app = Flask(__name__)  
  
@app.route('/', methods = ['GET'])  
def Login():  
    return render_template('Login.html')
```

```
@app.route('/details', methods = ['GET','POST'])
def login():
    if request.method == 'POST':
        name = request.form['username']
        output = 'Hi, Welcome '+name+ ' '
        resp = make_response(output)
        resp.set_cookie('username', name)
    return resp

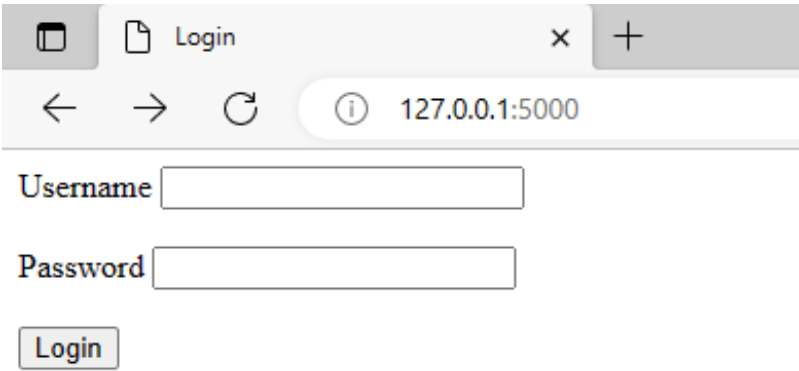
app.run(debug=True)
```

Login.html

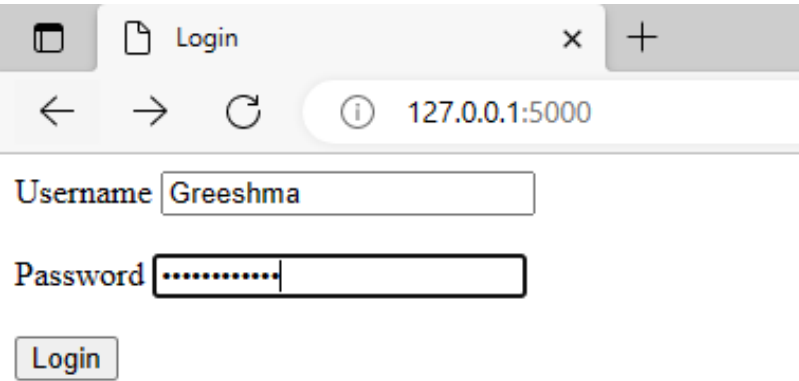
HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
    <form method="post" action="/details">
      <label for="username">Username</label>
      <input type="text" name="username" id="username"/>
      <br/>
      <br>
      <label for="password">Password</label>
      <input type="password" name="password" id="password"/>
      <br/>
      <br>
      <input type="submit" name="submit" id="submit" value="Login"/>
    </form>
  </body>
</html>
```

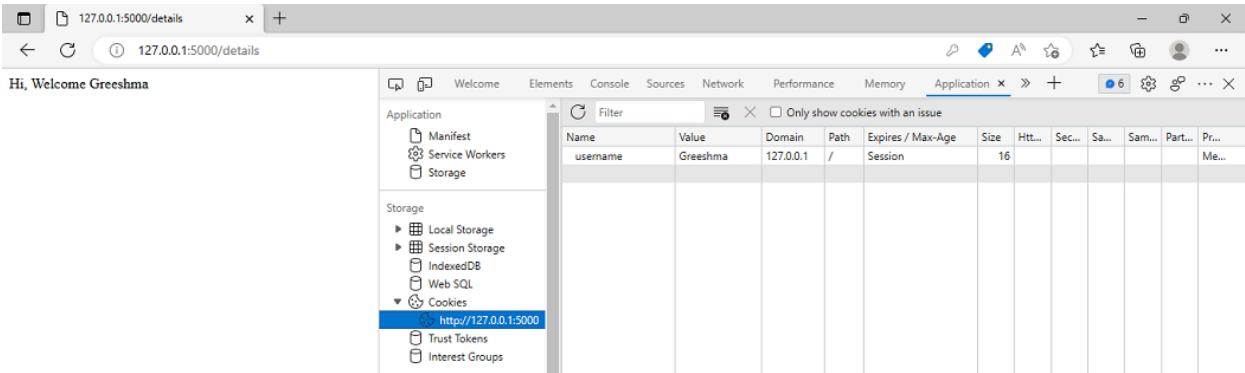
Output:



Login Page



User Logging



User Logged In and Cookie Tracker

From the above image, the user can also see the cookies of a website. Here, ‘username’ is the key and its value is ‘Greeshma’ which reminds us that cookies are generally key-value pairs. To see the cookies in your browser, click the last

3 dots on the browser's right corner>> More Tools>>Developer Tools>>Application window.

Getting website Visitors counted through cookies

In the below code, we want to know the number of visitors visiting our website. We are first retrieving the visitor's count by the usage of cookies. But there is no variable named visitors count that we created previously. As this key(visitors count) is not present in the dictionary, it will take the value of 0 that is specified in the second parameter as per the dictionary collection in python. Hence, for the first-time visitors count=0, then incrementing the count according to the user's visit to the website. The `make_response()` gets the response object and is used for setting the cookie.

Python3

```
from flask import Flask, request, make_response

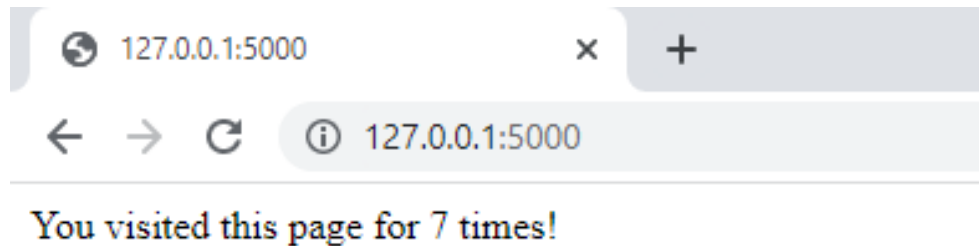
app = Flask(__name__)
app.config['DEBUG'] = True

@app.route('/')
def visitors_count():
    # Converting str to int
    count = int(request.cookies.get('visitors count', 0))
    # Getting the key-visitors count value as 0
    count = count+1
    output = 'You visited this page for '+str(count) + ' times'
    resp = make_response(output)
    resp.set_cookie('visitors count', str(count))
    return resp

@app.route('/get')
def get_visitors_count():
    count = request.cookies.get('visitors count')
    return count

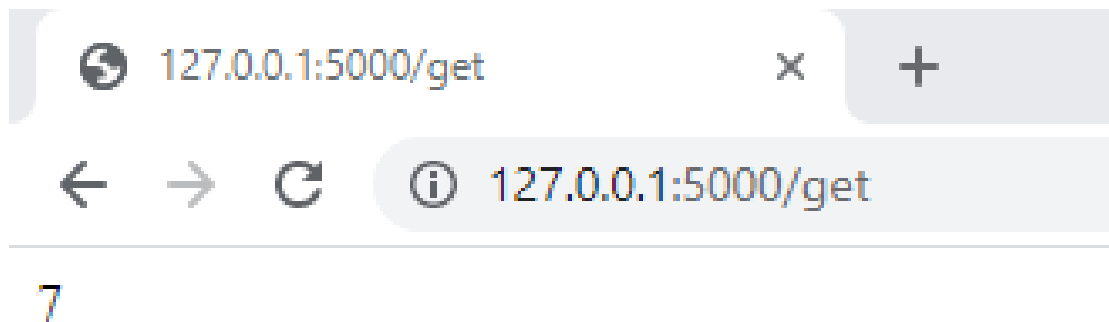
app.run()
```

Output: Url – `http://127.0.0.1:5000`



Output

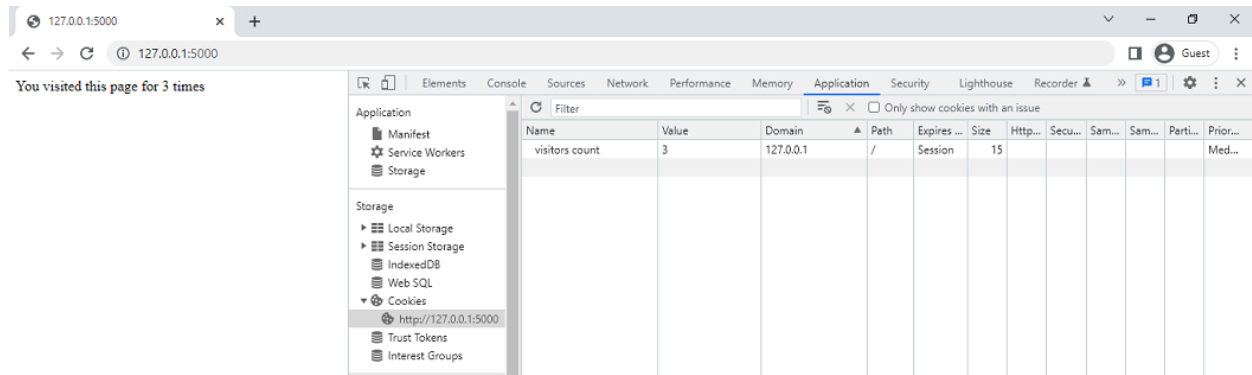
Url for below output- `http://127.0.0.1:5000/get`



Visitors count output using cookies

In the above output screenshot, the value of the website visitors count is retrieved using `request.cookies.get()` method.

Cookies Tracking in Browser –



Cookie Tracker for visitors count application

The flask cookies can be secured by putting the secure parameter in `response.set_cookie('key', 'value', secure = True)` and it is the best-recommended practice to secure cookies on the internet.

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!



GeeksforGeeks



3

Previous Article

How to use Flask-Session in Python Flask ?

Next Article

How to return a JSON response from a Flask API ?

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
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
System Design Bootcamp
Interview Questions

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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved