



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

CSRF Protection in Flask

Last Updated : 05 Feb, 2023

Let's see how you can manually protect your data using CSRF protection by doing a mini-project in [Flask](#). In this, we will have to create a webpage containing 2 forms using [Python](#) one of them is having protection. By creating forms like these we can easily see the results and advantages of using CSRF protection for our application.

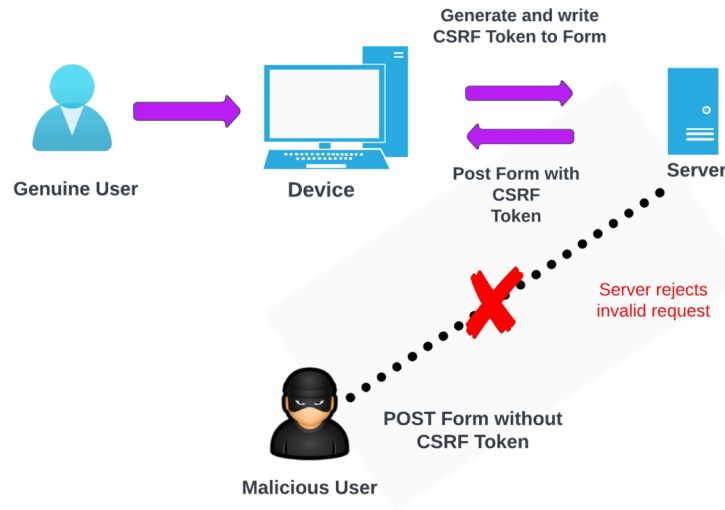
What is CSRF?

Cross-Site Request Forgery(CSRF) is a weighty exposure that results from weak gathering administration. If that requests shipped by an application aren't rare, it's likely for an aggressor to art a certain request and transmits that to a consumer. If the consumer communicates accompanying the workout request, and gatherings aren't controlled correctly, an aggressor grant permission within financial means to acquire the gathering similarity of that consumer and complete activity requests on their side.

Solution for Preventing CSRF Attacks

Cross-Site Request Forgery (CSRF) attacks are comparably smooth to diminish. One of the plainest habits to manage this is through the use of CSRF tokens, which are uncommon principles dynamically created by a server-side request and shipped to the customer. Since these principles are rare for each request, and uniformly changeful, it is almost hopeless for a raider to pre-generate the URLs/requests for an attack.

recreate image



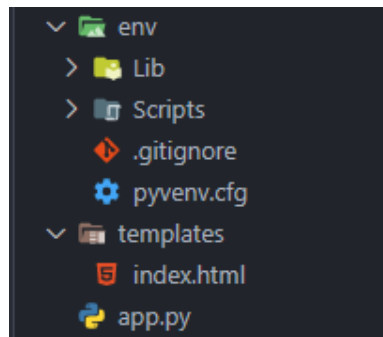
Example of CSRF Protection in Flask

Step 1: Create a Virtual environment for our application and install the following packages.

Step 2: Installing Packages.

```
pip install flask, flask-wtf
```

Step 3: You should have to create a folder structure like this.



Step 4: app.py

In Flask, we are having generally 2 ways to create a form one by using FlaskForm and another by creating forms manually. FlaskForm processes the request that already getting CSRF Protection. Csrf requires a secret key by default, it uses the Flask app's Secret Key. If you like to set up a separate token then you can use WTF_CSRF_SECRET_KEY instead of using a flask app's

secret key. While using FlaskForm, you will have to render the forms CSRF field. You can disable the CSRF Protection in all views by default, then set `WTF_CSRF_CHECK_DEFAULT` to False in the app.py file.

Python3

```
from flask import Flask, render_template, request
from flask_wtf import CSRFProtect

app = Flask(__name__)
app.secret_key = b'_53oi3uriq9pifpff;apl'
csrf = CSRFProtect(app)

@app.route("/protected_form", methods=['GET', 'POST'])
def protected_form():
    if request.method == 'POST':
        name = request.form['Name']
        return (' Hello ' + name + '!!!')
    return render_template('index.html')

@app.route("/unprotected_form", methods=['GET', 'POST'])
def unprotected_form():
    if request.method == 'POST':
        name = request.form['Name']
        return (' Hello ' + name + '!!!')
    return render_template('index.html')

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

Step 5: templates/index.html

A simple HTML page is set up for the app to show the unprotected and protected submission of the form.

HTML

```
<html>
<head></head>
<body>
<form action="{ { url_for('protected_form') } }" method="POST">
    <label for="Name">Your Name Please ? </label>
```

```

<input type="text" name="Name">
<input type="hidden" name="csrf_token" value = "{{ csrf_token() }}" />
<button type="submit">Submit</button>
</form>

<form action="{{ url_for('unprotected_form') }}" method="POST">
  <label for="Name">Your Name Please ? </label>
  <input type="text" name="Name">
  <button type="submit">Submit</button>
</form>

</body>
</html>

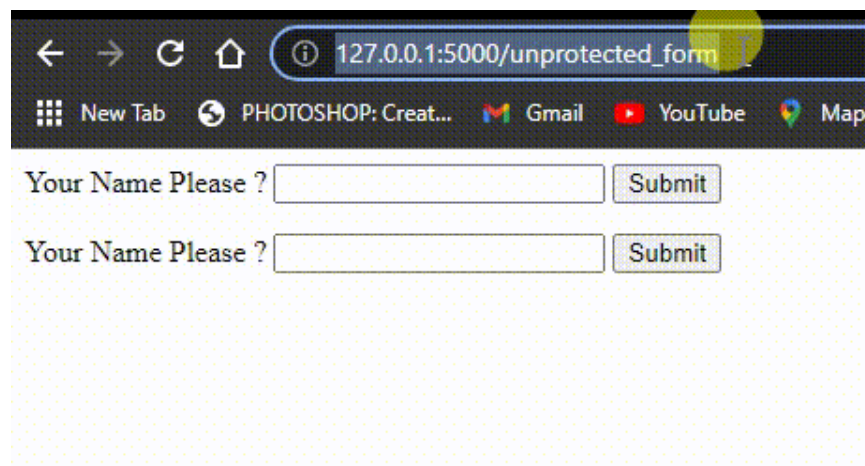
```

Step 5: Now run it to see the webpage and perform the practice.

```
python app.py
```

Output:

Visit '127.0.0.1:5000/protected_form' and try submitting both forms and one by one you should get the following outputs. While submitting the first form we applied the token inside the form so that it checks the token if it presents it serves the request else it generates an error.



kunal...



5

[Previous Article](#)[Flask Rendering Templates](#)[Next Article](#)[Template Inheritance in Flask](#)

Similar Reads

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 fix - CSRF token mismatch error

In this article, we are looking for a possible solution to fix the "CSRF token mismatch error". we will start by understanding what is csrf ? and why we...

3 min read

How to Disable Django's CSRF Validation?

Cross-Site Request Forgery (CSRF) protection is a critical security feature in Django that helps protect your web applications from certain types of attacks....

3 min read

CSRF token in Django

Django provides a feature known as a CSRF token to get away from CSRF attacks that can be very dangerous. when the session of the user starts on a website, a...

6 min read

Cross Site Scripting (XSS) Protection in Django

Django protects your website from XSS attacks by employing CSRF tokens, unique for each user, hidden in forms. This article will show how to implement...

3 min read

Python Flask - ImmutableMultiDict

MultiDict is a sub-class of Dictionary that can contain multiple values for the same key, unlike normal Dictionaries. It is used because some form elements ha...

2 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
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

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

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

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

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved