HTML   CSS   JavaScript   TypeScript   PHP   Tailwind   Bootstrap   Foundation   Bulma   ReactJS

Laravel | CSRF Protection

Last Updated : 30 Nov, 2019

Read   Discuss   Courses   Practice   Video

[Cross-Site Request Forgery (CSRF)](#) is a type of attack that performed by the attacker to send requests to a system with the help of an authorized user who is trusted by the system.

Laravel provides protection with the **CSRF** attacks by generating a **CSRF token**. This **CSRF token** is generated automatically for each user. This token is nothing but a random string that is managed by the Laravel application to verify the user requests.

How to Use: This CSRF token protection can be applied to any HTML form in Laravel application by specifying a hidden form field of CSRF token. The requests are validated automatically by the CSRF **VerifyCsrfToken** middleware.

There are three different ways in which you can do this.

1. **@csrf**
2. **csrf_field()**
3. **csrf_token()**

@csrf: This is a blade template directive for generating the hidden input field in the HTML form.

- **Syntax:**

```
<form method="POST">
   @csrf  // Generate hidden input field
   .....
   .....
</form>
```

- **Example:**

```
<!DOCTYPE html>
```

```
        </head>
        <body>
            <section>
                <h1>CSRF Protected HTML Form</h1>
                <form method="POST">
                    @csrf

                    <input type="text" name="username"
                                        placeholder="Username">
                    <input type="password" name="password"
                                        placeholder="Password">
                    <input type="submit" name="submit" value="Submit">
                </form>
            </section>
        </body>
    </html>
```

csrf_field(): This function can be used to generate the hidden input field in the HTML form.

Note: This function should be written inside double curly braces.

- **Syntax:**

```
<form method="POST"<

    // Generate hidden input field
    {{ csrf_field() }}
    .....
    .....
</form>
```

- **Example:**

```
<!DOCTYPE html>
<html>
    <head>
        <title>Laravel | CSRF Protection</title>
    </head>
    <body>
        <section>
            <h1>CSRF Protected HTML Form</h1>
            <form method="POST">
                {{ csrf_field() }}

                <input type="text" name="username"
                                    placeholder="Username">
                <input type="password" name="password"
                                    placeholder="Password">
                <input type="submit" name="submit"
```
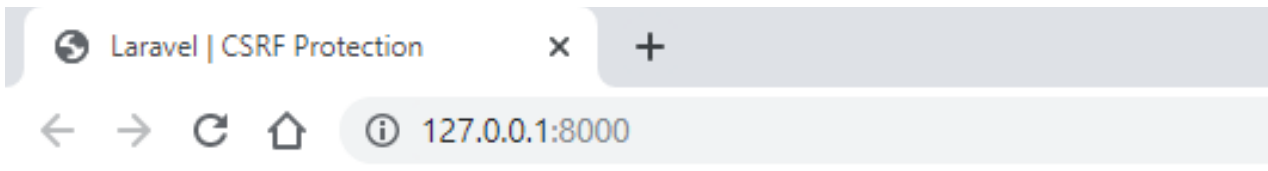
```
        </body>
    </html>
```

csrf_token(): This function just gives a random string. This function does not generate the hidden input field.

Note: HTML input field should be written explicitly. This function should be written inside double curly braces.

- **Syntax:**

```
<form method="POST">
    <input type="hidden" name="_token" value="{{ csrf_token() }}">
    .....
    .....
</form>
```

- **Example:**

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Laravel | CSRF Protection</title>
    </head>
    <body>
        <section>
            <h1>CSRF Protected HTML Form</h1>
            <form method="POST">
                <input type="hidden" name="_token" value="{{ csrf_token()

                <input type="text" name="username"
                             placeholder="Username">
                <input type="password" name="password"
                             placeholder="Password">
                <input type="submit" name="submit"
                                 value="Submit">
            </form>
        </section>
    </body>
</html>
```

Output: The output is going to be the same for any of the above three ways to generate a CSRF token. The CSRF token field should be written/generated at the start of every HTML form, using any of the three ways, in a Laravel application.

CSRF Protected HTML Form

Inspect Element Output:

Reference: https://laravel.com/docs/6.x/csrf

Related Articles

1.   Laravel | Artisan Commands to know in Laravel

2.   Laravel | Blade Templates Inheritance

3.   Laravel | Validation Rules

4.   Laravel | Artisan Console Introduction

5.   Laravel | Installation and Configuration

7.   Laravel | Route::resource vs Route::controller

8.   Different ways for passing data to view in Laravel

9.   Laravel | Controller Basics

10.   Laravel | View Basics

<div align="center">

Like   1

</div>

Next

How to check if the given date is weekend ?

Article Contributed By :

aakashpawar1999
@aakashpawar1999

Vote for difficulty

| Easy | Normal | Medium | Hard | Expert |

Article Tags :   Laravel,   Picked,   PHP,   Technical Scripter,   Web Technologies

Practice Tags :   PHP

| Improve Article | Report Issue |

GeeksforGeeks

A-143, 9th Floor, Sovereign Corporate Tower,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company
About Us

Careers

In Media

Contact Us

Privacy Policy

Copyright Policy

Advertise with us

Learn
DSA

Algorithms

Data Structures

SDE Cheat Sheet

Machine learning

CS Subjects

Video Tutorials

Courses

News
Top News

Technology

Work & Career

Business

Finance

Lifestyle

Knowledge

Languages
Python

Java

CPP

Golang

C#

SQL

Kotlin

Web Development
Web Tutorials

Django Tutorial

HTML

JavaScript

Bootstrap

ReactJS

NodeJS

Contribute
Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship