Install

$ npm install csurf

Khi sử dụng csurf thì có 2 options như thế này :

Sử dụng session hoặc cookie

1. Session : cài đặt middleware express-session
2. Cookie :
   * Sử dụng cookie-session
   * Hoặc cài đặt middleware cookie-parser

Cách 1 : Session

app.use(session({

    secret: "xHjnM5sTJungtin",

    resave: true,

    saveUninitialized: false

}));

const csrf = require("csurf");

const csrfProtection = csrf();

app.use(csrfProtection) // phải đặt sau session

Cách 2 : Cookie

const csrf = require("csurf");

const csrfProtection = csrf({ cookie: true });

const cookieParser = require("cookie-parser");

app.use(cookieParser());

app.use(csrfProtection); // phải đặt sau cookie-parser

Set CSRF Token cho View

Cách 1 : Mỗi khi render

router.get("/create", (req, res) => {

    res.render("dashboard/course/course-form", {

        layout: "dashboard",

        csrfToken: req.csrfToken()

    });

});

Cách 2 : Dùng middleware

const csrf = require("csurf");

const csrfProtection = csrf();

app.use(csrfProtection); // phải đặt sau session

app.use((req, res, next) => {

    res.locals.csrfToken = req.csrfToken();

    next();

}); // middleware của csrfToken phải đặt trước tất cả routes

/\* Route \*/

app.locals.URL = URL;

app.use(homeRoute);

app.use("/dashboard/courses", courseRoute);

Phải đặt middleware này trước tất cả các routes

\*res.locals : là thuộc tính của middleware chỉ dành cho ~ response nào render view

Ở View

<form action="/process" method="POST">

  <input type="hidden" name="\_csrf" value="{{csrfToken}}">

  Favorite color: <input type="text" name="favoriteColor">

  <button type="submit">Submit</button>

</form>

CSRF nhận từ những nguồn nào ?

* req.body.\_csrf - typically generated by the body-parser module.
* req.query.\_csrf - a built-in from Express.js to read from the URL query string.
* req.headers['csrf-token'] - the CSRF-Token HTTP request header.
* req.headers['xsrf-token'] - the XSRF-Token HTTP request header.
* req.headers['x-csrf-token'] - the X-CSRF-Token HTTP request header.
* req.headers['x-xsrf-token'] - the X-XSRF-Token HTTP request header.

## Sử dụng Ajax

Bước 1 : store csrf-token ở meta tag

<meta **name**="csrf-token" **content**="{{csrfToken}}">

Bước 2 : gán token vào post request

var token **=** document.querySelector('meta[name="csrf-token"]').getAttribute('content')

fetch('/process', {

  credentials**:** 'same-origin', *// <-- includes cookies in the request*

  headers**:** {

    'CSRF-Token'**:** token *// <-- is the csrf token as a header*

  },

  method**:** 'POST',

  body**:** {

    favoriteColor**:** 'blue'

  }

})

## Ignore Routes

Cái này là khi ta phát triển API song song với app

Mà API từ đâu cần CSRF 🡪 mình bỏ qua bằng cách khai báo middleware sau những routes đó

// mount api before csrf is appended to the app stack

app.use('/api', api)

// now add csrf and other middlewares, after the "/api" was mounted

app.use(bodyParser.urlencoded({ extended**:** false }))

app.use(cookieParser())

app.use(csrf({ cookie**:** true }))

Custom Error Handling

var bodyParser **=** require('body-parser')

var cookieParser **=** require('cookie-parser')

var csrf **=** require('csurf')

var express **=** require('express')

var app **=** express()

app.use(bodyParser.urlencoded({ extended**:** false }))

app.use(cookieParser())

app.use(csrf({ cookie**:** true }))

*// error handler*

app.use(function (err, req, res, next) {

**if** (err.code **!==** 'EBADCSRFTOKEN') **return** next(err)

*// handle CSRF token errors here*

  res.status(403)

  res.send('form tampered with')

})