Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

з дисципліни

«Безпека програмного забезпечення»

тема: «Засвоювання базових навичок роботи з валідацією токенів»

Виконав:

студент групи ІТ-03

Філоненко Владислав

Київ 2023

**Завдання**

Розширити Лабораторну работу 4 перевіркой сігнатури JWT токена.

Приклади SDK https://auth0.com/docs/quickstart/backend. У випадку асиметричного ключа, public є можливість отримати за посиланням https://kpi.eu.auth0.com/pem, aбо за формулою https://[API\_DOMAIN]/pem

Надати код рішення.

**Хід роботи**

**Код index.js**

const express = require('express');

const bodyParser = require('body-parser');

const path = require('path');

const axios = require('axios');

const fs = require('fs');

const { auth } = require('express-oauth2-jwt-bearer');

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

const port = 3000;

const app = express();

app.use(cookieParser());

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.listen(port, () => {

    console.log(`Example app listening on port ${port}`);

});

app.get('/', (req, res) => {

    const token = req.get('Authorization');

    if (token) {

        return res.json({

            token: token,

            logout: 'http://localhost:3000/logout',

        });

    }

    res.sendFile(path.join(\_\_dirname + '/index.html'));

});

app.post('/api/login', async (req, res) => {

    const { login, password } = req.body;

    const getUserTokenData = {

        method: 'post',

        url: `https://dev-rk1t874k8t8hyiky.us.auth0.com/oauth/token`,

        headers: {

            'Content-Type': 'application/x-www-form-urlencoded',

        },

        data: {

            grant\_type: 'password',

            username: login,

            password: password,

            audience: 'https://dev-rk1t874k8t8hyiky.us.auth0.com/api/v2/',

            scope: 'offline\_access',

            client\_id: 'Q4znBfWauNXYNvJjMkT7VKCBzhGHLtC2',

            client\_secret: 'Ri7cVPMWzOVsF-zBkddJJEoAAEUB7tzAbm4aW6gWO3RuSy4ZWQjpdOzcCoNZPTPu',

        },

    };

    await axios

        .request(getUserTokenData)

        .then((response) => {

            console.log('login', response.data);

            const currentDate = new Date();

            currentDate.setHours(currentDate.getHours() + 4); // 3 + 1

            res.cookie('token', response.data.access\_token, {

                expires: currentDate,

                httpOnly: true,

            });

            console.log(new Date());

            res.status(200).json({ token: response.data.access\_token });

        })

        .catch((error) => {

            console.error('Error login:', error);

        });

    res.status(401).send();

});

app.get('/logout', function (req, res) {

    console.log(req);

    res.cookie('token', '', {

        expires: 0,

        httpOnly: true,

    }).send();

});

const setHeader = (req, res, next) => {

    console.log(req.cookies);

    if (req.cookies.token) {

        req.headers.authorization = `Bearer ${req.cookies.token}`;

    }

    next();

};

const checkJwt = auth({

    audience: 'https://dev-rk1t874k8t8hyiky.us.auth0.com/api/v2/',

    issuerBaseURL: `https://dev-rk1t874k8t8hyiky.us.auth0.com/`,

});

app.get('/api/public', function (req, res) {

    res.json({

        message: 'Ви перейшли до публічної сторінки!!!!!',

    });

});

app.get('/api/private', setHeader, checkJwt, function (req, res) {

    res.json({

        message: 'Ви перейшли до приватної сторінки!!!!!',

    });

});

**Код index.html**

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8" />

        <meta name="viewport" content="width=device-width, initial-scale=1.0" />

        <title>Login</title>

        <script src="https://unpkg.com/axios/dist/axios.min.js"></script>

    </head>

    <body>

        <main id="main-holder">

            <div>

                <a href="/api/public">public</a>

                <a href="/api/private" id="private">private</a>

                <a href="/logout" id="logout">Logout</a>

            </div>

            <h1 id="login-header">Login</h1>

            <div id="login-error-msg-holder">

                <p id="login-error-msg">

                    Invalid username <span id="error-msg-second-line">and/or password</span>

                </p>

            </div>

            <form id="login-form" action="/api/login" method="post">

                <input

                    type="text"

                    name="login"

                    id="username-field"

                    class="login-form-field"

                    placeholder="Username" />

                <input

                    type="password"

                    name="password"

                    id="password-field"

                    class="login-form-field"

                    placeholder="Password" />

                <input type="submit" value="Login" id="login-form-submit" />

            </form>

        </main>

    </body>

    <style>

        html {

            height: 100%;

        }

        body {

            height: 100%;

            margin: 0;

            font-family: Arial, Helvetica, sans-serif;

            display: grid;

            justify-items: center;

            align-items: center;

            background-color: #3a3a3a;

        }

        #logout {

            opacity: 0;

        }

        #main-holder {

            width: 100%;

            height: 70%;

            display: grid;

            justify-items: center;

            align-items: center;

            background-color: white;

            border-radius: 7px;

            box-shadow: 0px 0px 5px 2px black;

        }

        #login-error-msg-holder {

            width: 100%;

            height: 100%;

            display: grid;

            justify-items: center;

            align-items: center;

        }

        #login-error-msg {

            width: 23%;

            text-align: center;

            margin: 0;

            padding: 5px;

            font-size: 12px;

            font-weight: bold;

            color: #8a0000;

            border: 1px solid #8a0000;

            background-color: #e58f8f;

            opacity: 0;

        }

        #error-msg-second-line {

            display: block;

        }

        #login-form {

            align-self: flex-start;

            display: grid;

            justify-items: center;

            align-items: center;

        }

        .login-form-field::placeholder {

            color: #3a3a3a;

        }

        .login-form-field {

            border: none;

            border-bottom: 1px solid #3a3a3a;

            margin-bottom: 10px;

            border-radius: 3px;

            outline: none;

            padding: 0px 0px 5px 5px;

        }

        #login-form-submit {

            width: 100%;

            padding: 7px;

            border: none;

            border-radius: 5px;

            color: white;

            font-weight: bold;

            background-color: #3a3a3a;

            cursor: pointer;

            outline: none;

        }

        .info {

            width: 50%;

            overflow-x: hidden;

        }

    </style>

    <script>

        const token = sessionStorage.getItem('token');

        if (token) {

            axios

                .get('/', {

                    headers: {

                        Authorization: `Bearer ${token}`,

                    },

                })

                .then((response) => {

                    const { token } = response.data;

                    if (token) {

                        const mainHolder = document.getElementById('main-holder');

                        const loginHeader = document.getElementById('login-header');

                        loginForm.remove();

                        loginErrorMsg.remove();

                        loginHeader.remove();

                        const createDiv = document.createElement('div');

                        createDiv.className = 'info';

                        createDiv.innerText = `Token ${token}`;

                        mainHolder.append(createDiv);

                        logoutLink.style.opacity = 1;

                    }

                });

        }

        const loginForm = document.getElementById('login-form');

        const loginButton = document.getElementById('login-form-submit');

        const loginErrorMsg = document.getElementById('login-error-msg');

        const logoutLink = document.getElementById('logout');

        logoutLink.addEventListener('click', (e) => {

            e.preventDefault();

            sessionStorage.removeItem('token');

            axios({

                method: 'get',

                url: '/logout',

            });

            location.reload();

        });

        loginButton.addEventListener('click', (e) => {

            e.preventDefault();

            const login = loginForm.login.value;

            const password = loginForm.password.value;

            axios({

                method: 'post',

                url: '/api/login',

                data: {

                    login,

                    password,

                },

            })

                .then((response) => {

                    const { token } = response.data;

                    sessionStorage.setItem('token', token);

                    location.reload();

                })

                .catch((response) => {

                    loginErrorMsg.style.opacity = 1;

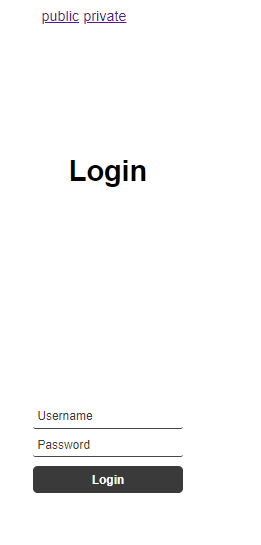
                });

        });

    </script>

</html>

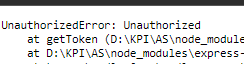
**Результат**



**По посиланню public отримуємо**



**По посиланню private отримуємо, якщо користувач не авторизований**



**По посиланню private отримуємо, якщо користувач авторизований**



**Висновок**

В ході лабораторної роботи, розширили лабораторну роботу 4 методом auth для перевірки чи авторизований користувач.