**Spring Security – CORS (CROSS ORIGIN RESOURCE SHARING)**

In Spring Boot, handling Cross-Origin Resource Sharing (CORS) is crucial when you have a front-end application (e.g., JavaScript, React, Angular) running on a different domain from your Spring Boot backend. CORS is a security feature implemented by web browsers to prevent cross-origin requests that could be malicious.

**A demo project was done to illustrate the CORS.**

myPortal is the backend application and myPortal-client is the frontend application.

**myPortal Project:**

**Dependencies used in myPortal application:**

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

**MyPortalApplication class**:

Here we are allowing the websites which are coming from the <http://localhost:9090> endpoint only.

@CrossOrigin is the annotation is responsible for the CORS implementation. Just annotate the method with @CrossOrigin (origins = "http://localhost:9090"). The method which is mapped to “/access” endpoint can be accessed only when it is redirected from <http://localhost:9090> website.

* If u have the multiple controller classes, then annotate the class with @CrossOrigin(origins="\*")
* If u want to implement the CORS to all the methods and make it globally, write the bean method as per below.

@SpringBootApplication

@RestController

@CrossOrigin(origins="\*") // If u have multiple controller classes

public class MyPortalApplication {

//@CrossOrigin (origins = "http://localhost:9090")

@GetMapping("/access")

public String greeting () {

return "Welcome to Neokred Portal";

}

// 'http://localhost:8080/access' from origin 'http://localhost:9090' has been blocked by CORS policy: No

// 'Access-Control-Allow-Origin' header is present on the requested resource.

//For Making the CORS globally

@Bean

public WebMvcConfigurer configure () {

return new WebMvcConfigurer() {

public void addCorsMapping(CorsRegistry registry) {

registry. addMapping("/\*").allowedOrigins("http://localhost:9090");

}

};

}

public static void main(String[] args) {

SpringApplication.run(MyPortalApplication.class, args);

}

}

**myPortalClient Project:**

myPortalClient Project is the frontend application with which we can access the backend application.

**Dependencies used for the project:**

<dependency>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework. boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

**Portal.js file:**

The accessPortal is the requestId and apiResponse is the responseId. The event.preventDefault() will prevent the default reloading of the page when it is clicked. In function ajaxGet(), we are specifying to redirect to the <http://localhost:8080/access> api.

//Ajax Call

GET: $(document). ready (

function () {

// GET REQUEST //requestId. Specify the same requestId in the HTML.

$("#accessportal"). click(function(event) {

event.preventDefault();

ajaxGet();

});

// DO GET //responseId

function ajaxGet() {

$. ajax({

url : "http://localhost:8080/access",

success: function(result) {

$("#apiResponse").html(result);

}

});

}

})

**Home.html page**

Here, we are creating the AccessMyPortal button, on which it will be redirected to <http://localhost:8080/access> api when clicked.

<!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="ISO-8859-1" />

<title>Business user Application</title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" />

<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script

src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>

<script

src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>

<script src="/portal.js"></script>

</head>

<body>

<br />

<br />

<div class="container" align="center">

<button id="accessportal" type="button" class="btn btn-primary">Access

My portal</button>

</div>

<h1 id="apiResponse"></h1>

</body>

</html>

**MyPortalClientApplication:**

We are mapping the home.html page to "/" endpoint.

@SpringBootApplication

@Controller

public class MyPortalClientApplication {

@GetMapping ("/")

public String home () {

return "home";

}

/\*

\* 'http://localhost:8080/access' from origin 'http://localhost:9090' has been

\* blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on

\* the requested resource.

\*/

public static void main (String [] args) {

SpringApplication.run(MyPortalClientApplication.class, args);

}

}

**Another project was done to illustrate the CORS.**

A login page was created to access the backend application only when you validate with the credentials.

Here we created the form where the user will enter the username and password.

**Home.html file.**

<!DOCTYPE html>

<html lang="en" xmlns:th="http://www.thymeleaf.org">

<head>

<meta charset="ISO-8859-1" />

<title>Login - Business User Application</title>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" />

<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<script

src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"></script>

<script

src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"></script>

<script src="validation.js"></script> <! -- Include the validation.js file -->

</head>

<body>

<br />

<br />

<div class="container" align="center">

<h1>Login </h1>

<form id="loginForm">

<div class="form-group">

<label for="username">Username:</label>

<input type="text" class="form-control" id="username" name="username" required>

</div>

<div class="form-group">

<label for="password">Password:</label>

<input type="password" class="form-control" id="password" name="password" required>

</div>

<button type="submit" class="btn btn-primary">Login</button>

</form>

</div>

<h1 id="apiResponse"></h1>

</body>

</html>

**Validation.js file**

The function validateCredentials is written to check the password for the given username. The password is the first four characters of the username. If the credentials are correct, it will redirect to

'http://localhost:8080/access' api.

$(document). ready (function () {

$('#loginForm'). submit(function(event) {

// Prevent the form submitting

event.preventDefault();

// Get username and password values

var username = $('#username').val();

var password = $('#password').val();

// Validate username and password

if (validateCredentials(username, password)) {

// If credentials are valid, redirect the user

window.location.href = 'http://localhost:8080/access';

} else {

// If credentials are not valid, display an error message

$('#apiResponse').text ('Invalid username or password. Please try again.');

}

});

// Function to validate username and password

function validateCredentials(username, password) {

// Check if password is the first four letters of the username

var firstFourLetters = username.substring(0, 4);

return password === firstFourLetters;

}

});