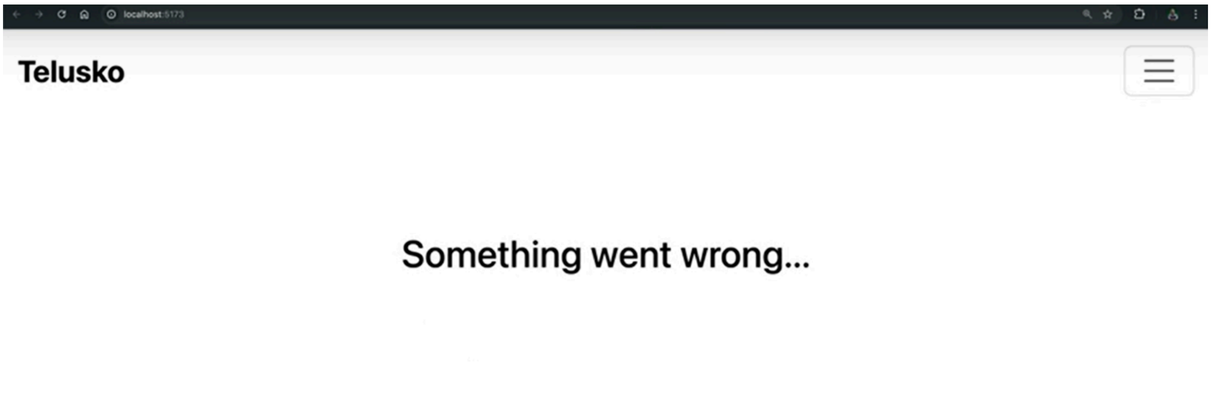
#23: CORS Issues in Spring Boot

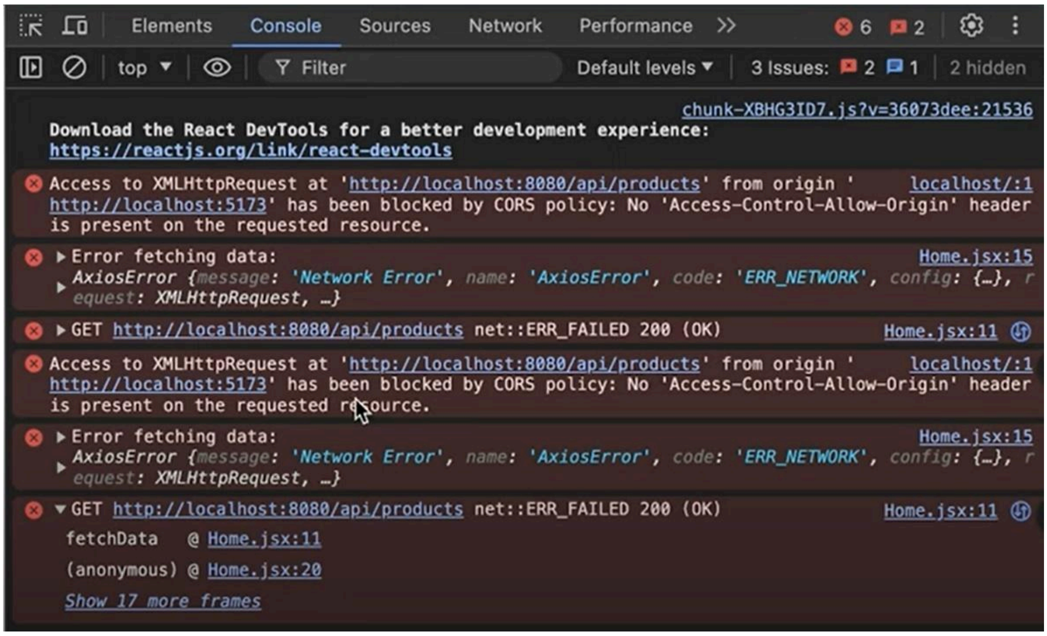
In our previous chapter, we successfully obtained output on both the Postman tool and the browser. However, we encountered an issue when trying to access our Spring Boot application from our React application. Let's decode this problem and find a solution.

Issue Encountered



Something went wrong...

While running our server-side application on port 8080, we attempted to access it from our React application running on port 5173. Unfortunately, we were unable to obtain the product list, encountering the following error message:



```
                                                  chunk-XBHG3ID7.js?v=36073dee:21536
Download the React DevTools for a better development experience:
https://reactjs.org/link/react-devtools
Access to XMLHttpRequest at 'http://localhost:8080/api/products' from origin '      localhost/:1
http://localhost:5173' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header
is present on the requested resource.
▸ Error fetching data:                                                         Home.jsx:15
▸ AxiosError {message: 'Network Error', name: 'AxiosError', code: 'ERR_NETWORK', config: {…}, r
  equest: XMLHttpRequest, …}
▸ GET http://localhost:8080/api/products net::ERR_FAILED 200 (OK)           Home.jsx:11
Access to XMLHttpRequest at 'http://localhost:8080/api/products' from origin '      localhost/:1
http://localhost:5173' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header
is present on the requested resource.
▸ Error fetching data:                                                         Home.jsx:15
▸ AxiosError {message: 'Network Error', name: 'AxiosError', code: 'ERR_NETWORK', config: {…}, r
  equest: XMLHttpRequest, …}
▾ GET http://localhost:8080/api/products net::ERR_FAILED 200 (OK)           Home.jsx:11
   fetchData    @ Home.jsx:11
   (anonymous) @ Home.jsx:20
   Show 17 more frames
```

Access to XMLHttpRequest from port 5173 to origin port 8080 has been blocked by CORS policy.

This error indicates that Cross-Origin Resource Sharing (CORS) is not permitted for security reasons, preventing data from being displayed.

This error occurs due to Cross-Origin Resource Sharing (CORS), a security feature implemented by web browsers to prevent websites from making requests to a different domain, protocol, or port than the one from which the original request originated.

CORS Policy: CORS is a security mechanism enforced by browsers that requires the backend server to explicitly allow requests from different origins (like different ports). Without this permission, the browser will block the request and throw a CORS error.

Steps to Reproduce the Issue

1. Start your server-side application on port 8080.
2. Hit the URL for your React application running on port 5173.
3. Check the browser console by right-clicking on the page and selecting the "Inspect" option. You will see the CORS error message.

Solution

To resolve this issue, we need to make changes on the server-side. Specifically, we will update our ProductController class by adding the @CrossOrigin annotation. Here's how to do it:

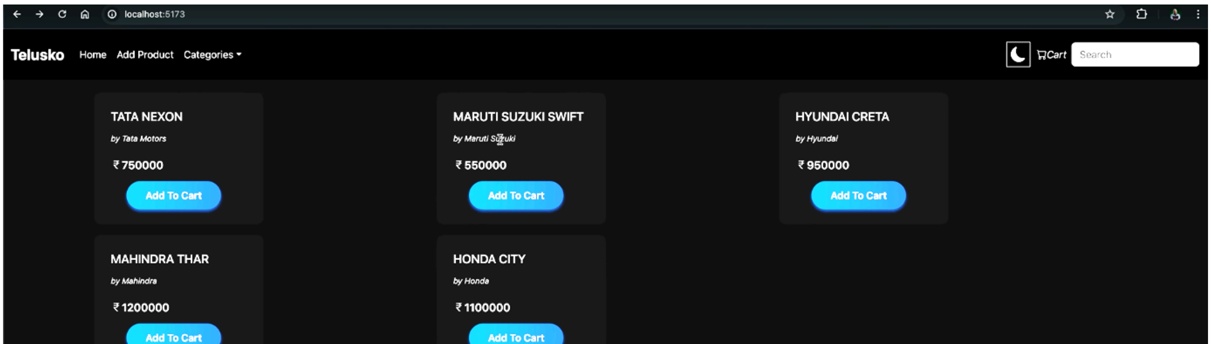
1. **Navigate to the ProductController class.**
2. **Add the @CrossOrigin annotation at the class level.** This will allow cross-origin requests from your React application.

```
@CrossOrigin
@RestController
public class ProductController {
   // Your existing code
}
```

By adding this annotation, we permit requests from the specified origin (in this case, http://localhost:5173), resolving the CORS issue.

Verifying the solution

1. **Restart your server-side application.**
2. **Reload the React application page in the browser.**
3. **Check the product list.** You should now see the product list updated with all products displayed.



Additional Features

In the next chapter, we will explore more features, such as adding a product from the frontend. These features are still under development.

Conclusion

By understanding and resolving the CORS issue, we ensure seamless communication between our React frontend and Spring Boot backend.

Stay Curious and Happy Coding! ☺