

CAPSTONE

GOTCHAS

LOGIN MESSAGE


This message on the Home page is counter-intuitive.

You will see this message when you **ARE** logged in. A better version of this message might be "If you are seeing this, you are authenticated."



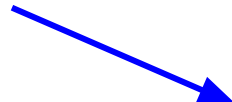
CORS ISSUES - BACKEND

In the controller on the right, note the `@CrossOrigin` annotation.



If you forget to add this to a new controller, it will most likely work when you test with Postman but **fail when you call the end point from your Vue code.**

Here is an example similar to the error you might see:



```
@RestController
@CrossOrigin
public class TestController {

    @GetMapping("/sayHi")
    public String
sayHi (@RequestParam String name) {
        return "Hi " + name + "!";
    }
}
```

```
Access to XMLHttpRequest at 'http://localhost:9000/sayHi?name=Yoav'
from origin 'http://localhost:8080' has been blocked by CORS policy:
Response to preflight request doesn't pass access control check: No
'Access-Control-Allow-Origin' header is present on the requested
resource.
```

CORS ISSUES - FRONTEND

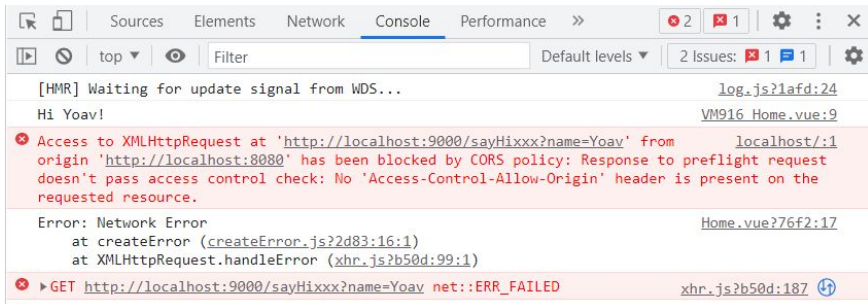
There is an issue with the capstone starter code which causes a CORS error rather than a 404 Not Found error that should occur.

This occurs when your front end code (again, this will NOT happen in Postman - only on the browser) calls an non-existent endpoint.

For example, if the Vue code attempts to hit the endpoint `/sayHiBlah` and only `/sayHi` exists as an endpoint, you might see an error similar to this:

```
@RestController
@CrossOrigin
public class TestController {


    @GetMapping("/sayHi")
    public String sayHi(@RequestParam
String name) {
        return "Hi " + name + "!";
    }
}
```



CORS ISSUES - FRONTEND

There is an issue with the capstone starter code which causes a CORS error rather than a 404 Not Found error that should occur.

This occurs when your front end code (again, this will NOT happen in Postman - only on the browser) calls an non-existent endpoint.

For example, if the Vue code attempts to hit the endpoint `/sayHiBlah` and only `/sayHi` exists as an endpoint, you might see an error similar to this: 

```
@RestController
@CrossOrigin
public class TestController {

    @GetMapping("/sayHi")
    public String sayHi(@RequestParam
String name) {
        return "Hi " + name + "!";
    }
}
```

Access to XMLHttpRequest at
'http://localhost:9000/sayHiBlah?name=Yoav' from origin
'http://localhost:8080' has been blocked by CORS policy:
Response to preflight request doesn't pass access
control check: No 'Access-Control-Allow-Origin' header
is present on the requested resource.