

Angular

Tahaluf Training Center 2021



Chapter 7

1 CORS

2 Hits API

- Delete
- GetByName
- GetById



CORS

Cross-Origin Resource Sharing (CORS) is a mechanism based on HTTP headers that allow browsers to identify which request comes from the allowed domain list, at the same time filter out unknown requests.



CORS

Browsers make a “preflight” request to the server hosting the cross-origin resource to check that the server will permit the actual request. In that preflight, the browser sends information about the HTTP method and headers that will be used later in the actual request.



CORS

Fetching data from sites and APIs that enforced CORS policy without proper headers can lead to the "Access to XMLHttpRequest has been blocked by CORS policy" error message in Chrome DevTools. The error comes in several forms, including useful information for debugging.



CORS

Angular is running in development mode. Call
enableProdMode() to enable production mode.

[core.js:28047](#)

[WDS] Live Reloading enabled.

[index.js:52](#)

✖ Access to XMLHttpRequest at '<https://localhost:44379/api/Course>' from origin '<http://localhost:4200>' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.



CORS

To avoid this error, add the following lines of code to your API project.

In the **Startup.cs** file → at the beginning of **ConfigureServices** method.



CORS

```
services.AddCors(corsOptions =>
    {
        corsOptions.AddPolicy("x",
            builder =>
            {

                //builder.WithOrigins("http://127.0.0.1:4200",
                //builder.WithOrigins("http://localhost:4200", "https://localhost:4200")
                //
                .AllowAnyHeader()
                //
                .AllowAnyMethod();

            builder.AllowAnyOrigin().AllowAnyMethod().AllowAnyHeader();
            });
    });
```



CORS

Then in the **Startup.cs** file → **Configure** method add this line of code:

```
app.UseCors("x");
```



Chapter 7

1 CORS

2 Hits API

- Delete
- GetByName
- GetById



Hits API

In home.service.ts

```
deleteCourse(id: number) {  
  this.spinner.show();  
  this.http  
    .delete('https://localhost:44379/api/Course/delete/' + id)  
    .subscribe(  
      (data) => {  
        this.spinner.hide();  
        this.toaster.success('Deleted!');  
      },  
      (error) => this.toaster.error(error.status)  
    );  
}
```



Hits API

In commponent.ts

```
delete(){  
  if(this.courseId){  
    this.homeService.deleteCourse(this.courseId);  
  }else{  
    this.toaster.error('Deleted');  
  }  
}
```



Hits API

In component.html

```
<mat-card-actions>  
  <button mat-button (click)="delete()">DELETE</button>  
  <button mat-button>UPDATE</button>  
</mat-card-actions>
```



Hits API

In home.service.ts

```
getByCourseName(courseName: string) {  
  this.http  
    .get('https://localhost:44379/api/Course/getByCourseName/' + courseName)  
    .subscribe((result) => {  
      debugger;  
      this.selectedCourse = result;  
    });  
}
```



Hits API

In commponent.ts

```
courseByName() {  
  if (this.courseName) this.homeService  
    .getByCourseName(this.courseName);  
}
```



Hits API

In home.service.ts

```
getCourseById(id: number){  
  return this.http.get('https://localhost:44379/api/Course/getById/' + id)  
    .subscribe((data)=> {this.selectedCourse = data} );  
}
```



Hits API

In commponent.ts

```
openCourseDetailes() {  
  if (this.courseId) {  
    this.homeService.getCourseById(this.courseId);  
    this.openCourse.emit();  
  }  
}
```

