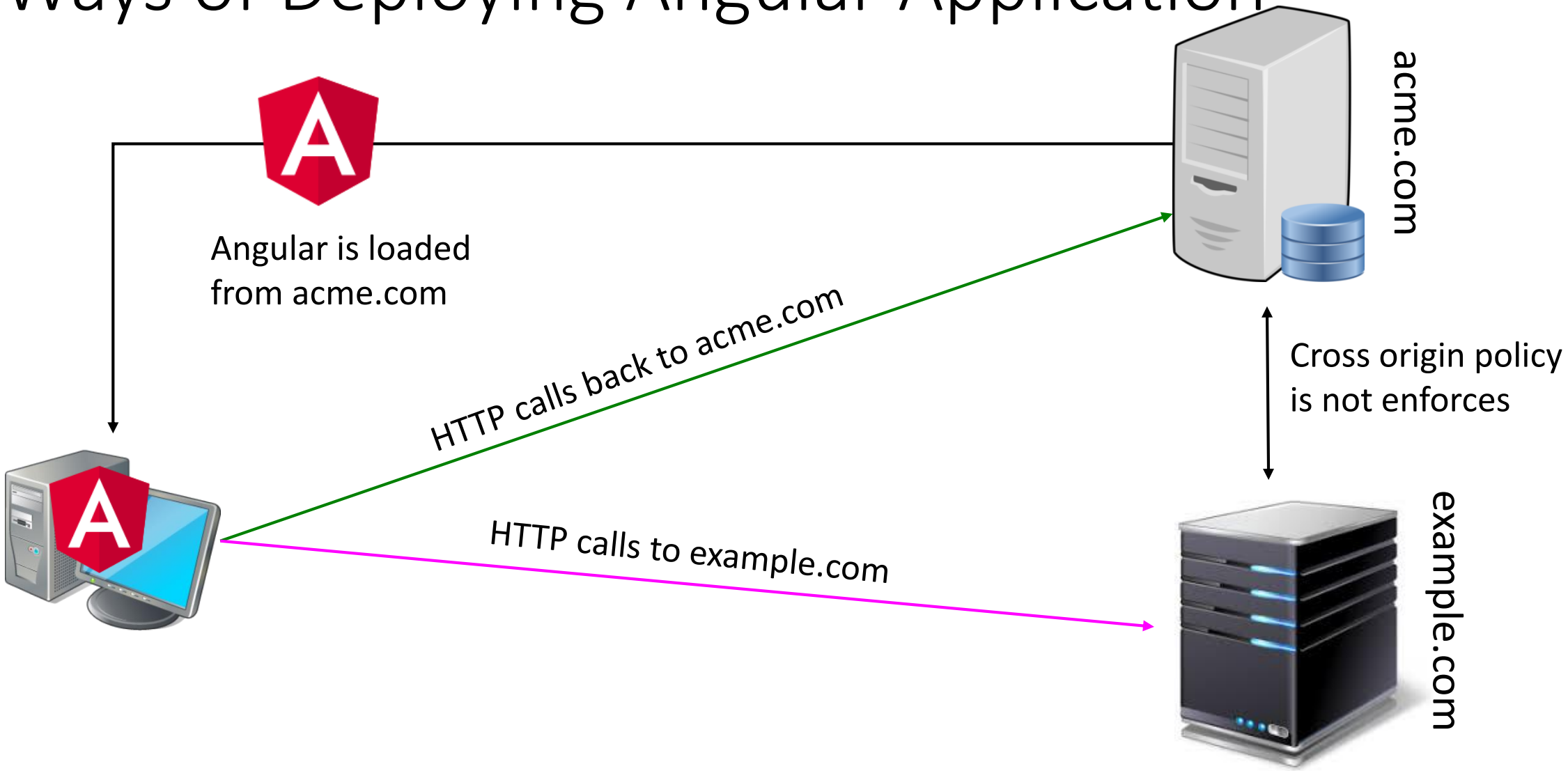
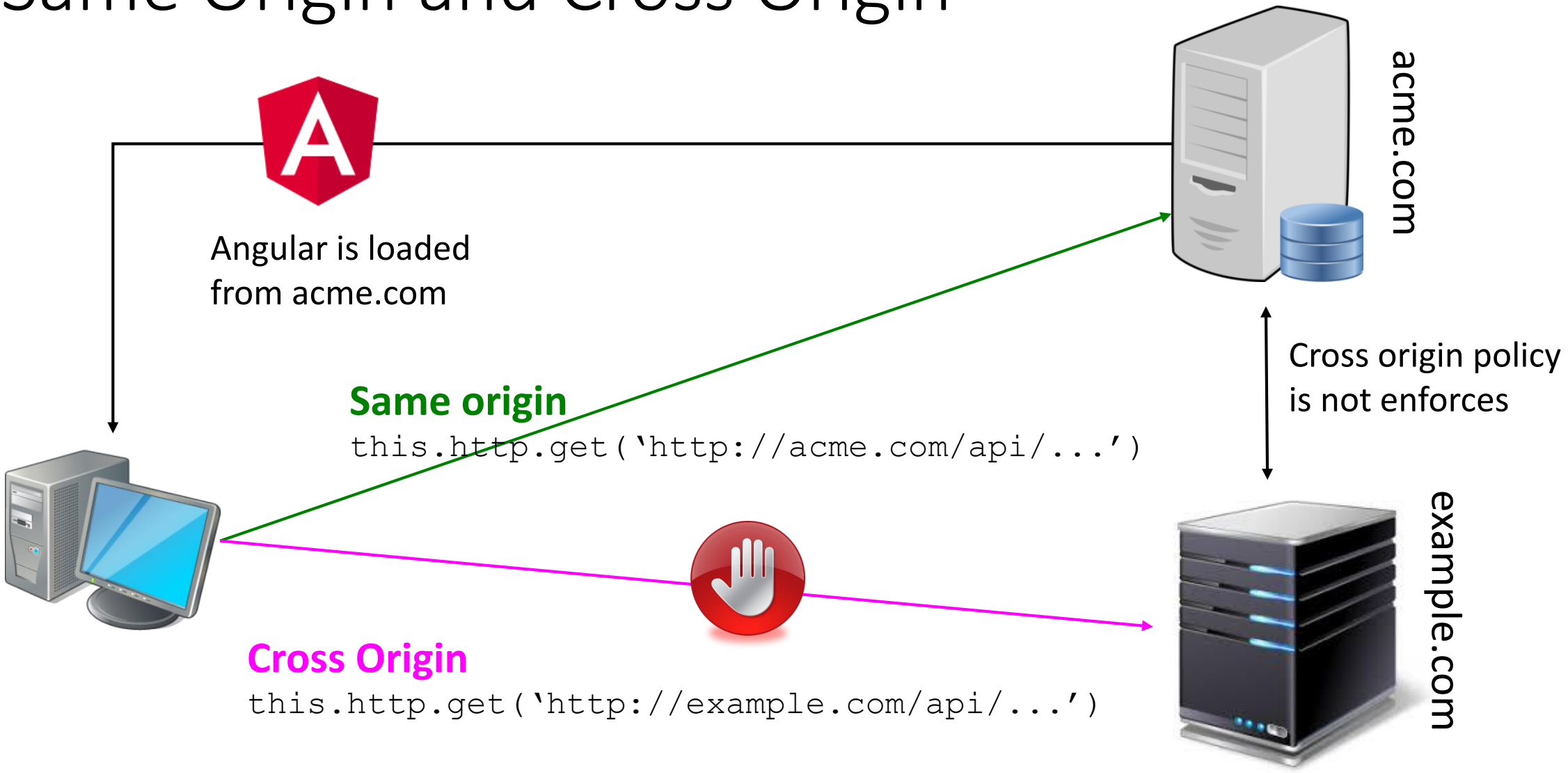
Day 5

Ways of Deploying Angular Application

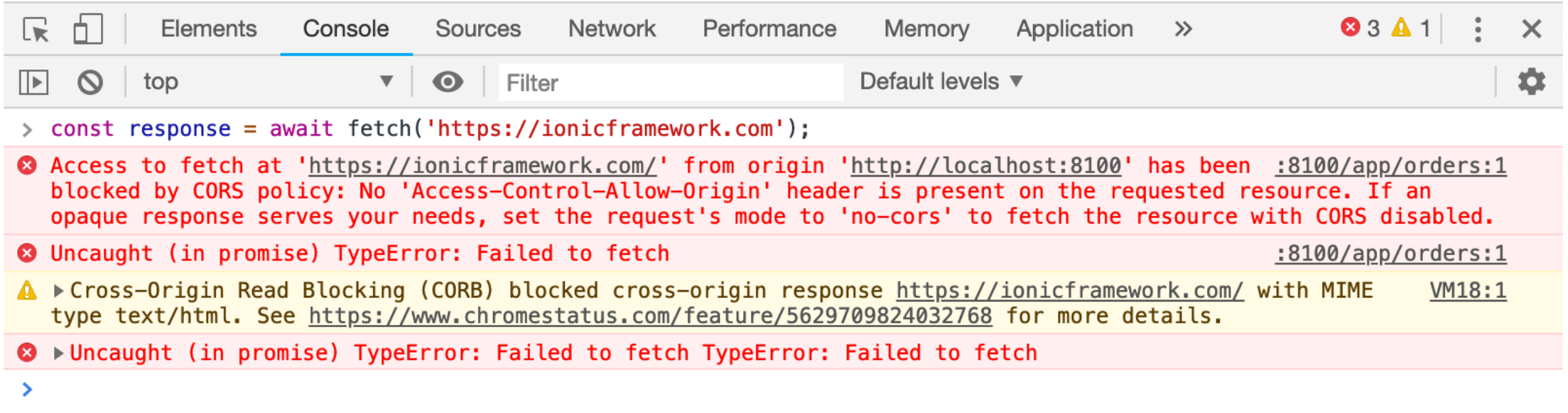


Same Origin and Cross Origin



Cross Origin Error

- Browser reject cross origin request for certain type of media eg. JSON, XML



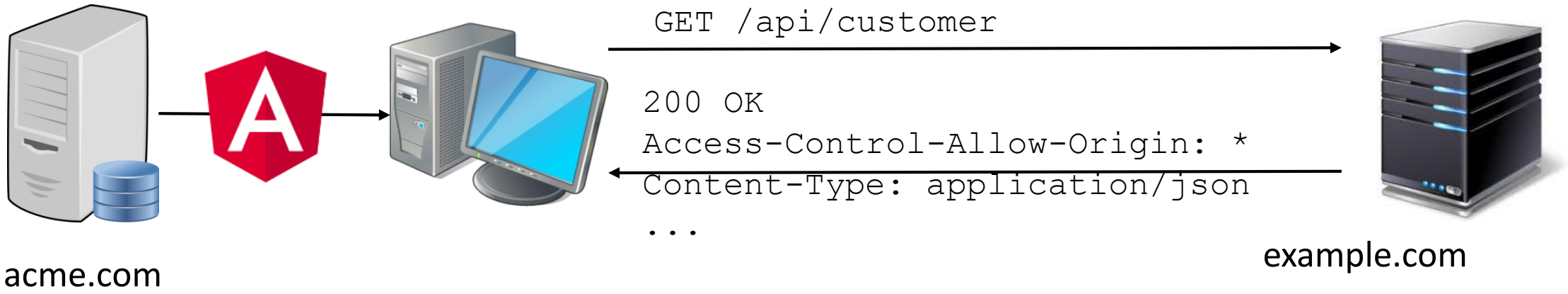
Displayed in Developer Tools

Cross Origin Resource

- Browser will only permit certain types of cross origin resource access
  - GET method
  - Media type include CSS, JavaScript , media (eg images, videos)
  - All other methods and media types are blocked
- Cross origin resource allows clients to make cross origin request
  - Using any method POST, PUT with any media type
- REST servers must opt-in
  - By adding extra headers in the response
- CORS is not enforce if request is from server to server
  - Eg. SpringBoot/Express calling a API endpoint

Setting Angular Development for Cross Origin

- Angular HTTP is making request to a REST endpoint that is hosted on a different origin
  - Different from the one that the Angular application is served from
- The REST endpoint needs to have CORS headers in its response
  - `Access-Control-Allow-Origin` header
  - To indicate if a response can be shared with request from a different origin



Enabling CORS in SpringBoot with Annotations

```
@RestController
@RequestMapping(path="/api/customer")
@CrossOrigin(origins="*")
public class CustomerRESTController {

    @GetMapping(path="{custId}")
    @CrossOrigin(origins="*")
    public ResponseEntity<String> getCustomer(
            @PathVariable String custId) {
        ...
    }
}
```

Annotation can be added to the controller or specific method

Response will include the following header

```
Access-Control-Allow-Origin: *
```

Enabling CORS in SpringBoot Globally

Implement the `WebMvcConfigurer` interface

```
public class EnableCORS implements WebMvcConfigurer {
    final String path;
    final String origins;
    public EnableCORS(String path, String origins) {
        this.path = path;
        this.origins = origins;
    }

    @Override
    public void addCorsMappings(CorsRegistry registry) {
        registry.addMapping(path)
            .allowedOrigins(origins)
    }
}
```

Override the `addCorsMappings` method

Configure the resource path and the allowed origins

Enabling CORS in SpringBoot Globally

```
@SpringBootApplication
public class CustomerRestApplication {
    public static void main(String[] args) {
        SpringApplication.run(CustomerRestApplication.class, args);
    }

    @Bean
    public WebMvcConfigurer corsConfigurer() {
        return new EnableCORS("/api", "*");
    }
}
```

Configure CORS globally by returning the configured CORS configuration
Allow CORS on `/api` for all origins

Setting Angular Development for Same Origin

- Access cross origin resource, Angular and the REST backend is running on 2 different servers
  - Angular makes HTTP call to a separate server

```
this.http.get('https://acme.com/api/customer/1')
```

- Same origin XHR calls are made to the same server from which the Angular application is loaded from
  - Problem because the call goes back to the Angular development server

```
this.http.get('/api/customer/1')
```

Development Environment

Type `http://localhost:4200` into the browser

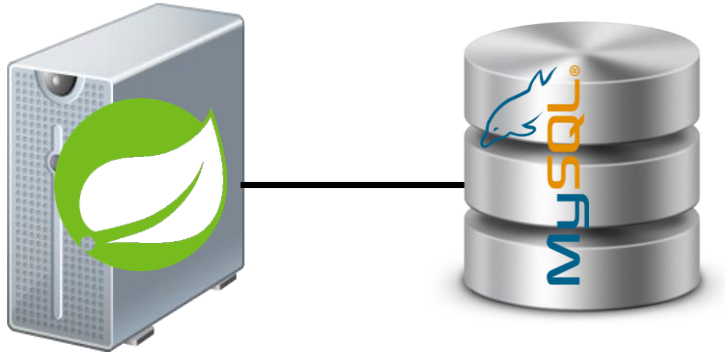
Angular application makes HTTP request `http://localhost:3000/customers` to SpringBoot application

Browser makes request to Angular development server to load application

`http://localhost:4200`

`ng serve`

`http://localhost:3000`



SpringBoot

Serving Angular Application from SpringBoot

- Change all HTTP request from `http://<server>/customers` to just `/customers`
  - Make all request independent of the deployment since the Angular application is part of the SpringBoot application
- Create a proxy to route all request from `/customers` to `http://<server>/customers`
  - Run development server with proxy
  - During development, its still 2 servers
- Build Angular for production once the client-side application completes
- Copy Angular application artefacts to SpringBoot's application static folder
  - Eg. `public`

Development Environment

Type `http://localhost:4200` into the browser

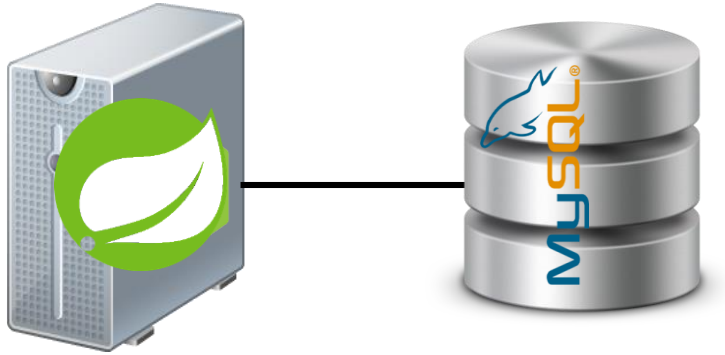
Route `/customers` request to `http://localhost:3000/customers`

Browser makes request to Angular development server to load application

`http://localhost:4200`

`ng serve`

`http://localhost:3000`



SpringBoot

Change All HTTP Request

```
import { HttpClient, HttpParams } from '@angular/common/http';

import { firstValueFrom, take } from 'rxjs'
@Injectable()
export class CustomerService {

   constructor(private http: HttpClient) { }

   getAllCustomers(offset = 0, limit = 50): Promise<any> {
      const qs = new HttpParams()
         .set('offset': offset)
         .set('limit': limit);
      return (
         firstValueFrom(
            this.http.get('/customers', { params: qs })
               .pipe(take(1))
         )
      );
   }
}
```

http://localhost:3000/customers

Proxy File

proxy.config.js

```
const PROXY_CONFIG = [
   {
      context: [ '/**' ],
      target: 'http://localhost:8080',
      secure: false,
      logLevel: 'debug'
   }
]
module.exports = PROXY_CONFIG;
```

Forward all HTTP request starting with **/** to **http://localhost:8080**

```
ng serve --proxy-config proxy.config.js
```

Build and Copy Artefacts to SpringBoot

`ng build --prod`

`mvn package`