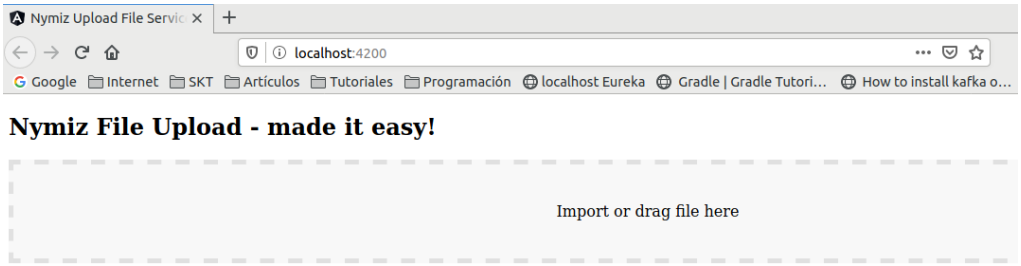
Conexión Frontend Angular con Backend Spring Boot

Frontend (Angular)

Tenemos un sencillo frontend en Angular:



En el componente (app.component.ts), se debe indicar

- línea 12: la URL del servicio del backend que se quiere invocar
- línea 20: hay que indicar una cabecera que indique 'multipart/form-data'

```
1  import { Component, ElementRef, OnInit, ViewChild } from '@angular/core';
2  import {FileUploader} from 'ng2-file-upload';
3
4  @Component({
5    selector: 'app-root',
6    templateUrl: './app.component.html',
7    styleUrls: ['./app.component.scss']
8  })
9  export class AppComponent implements OnInit {
10
11   title = 'Nymiz Upload File Service';
12   serviceUrl: string = 'http://localhost:8080/api/v1/files';
13
14   @ViewChild('fileInput') fileInput: ElementRef;
15
16   uploader: FileUploader;
17   isDropOver: boolean;
18
19   ngOnInit(): void {
20     const headers = [{name: 'Accept', value: 'multipart/form-data'}
21                     ];
22                     // {name: 'Accept', value: 'application/json'}
23                     //  {name: 'content-type', value: 'multipart/form-data'}
24                     // {name: 'access-control-allow-credentials', value: 'false'}
25                     // {name: 'access-control-allow-origin', value: '*'}
26     //this.uploader = new FileUploader({url: this.serviceUrl, autoUpload: true, headers: headers});
27
28     this.uploader = new FileUploader({
29       url: this.serviceUrl,
30       autoUpload: true,
31       method: 'post',
32       itemAlias: 'file',
33       allowedFileType: ['doc', 'docx', 'pdf', 'PDF'],
34       headers: headers
35       });
36
37       this.uploader.onCompleteAll = () => alert('File uploaded');
38   }
```

El contenido del template (app.component.html) es:

```
<h2> Nymiz File Upload - made it easy! </h2>

<div>
  <input #fileInput type="file" ng2FileSelect [uploader]="uploader" />

  <div class="drop-box" ng2FileDrop
       [ngClass]="{'dragover': isDropOver}"
       [uploader]="uploader"
       (fileOver)="fileOverAnother($event)"
       (click)="fileClicked()">
    <span class="drag-in-title">Import or drag file here</span>
    <span class="drag-over-title">Drop the file</span>
  </div>
</div>

<router-outlet></router-outlet>
```

Backend (Spring Boot y Java)

En el controlador, tendremos el endpoint que es llamado por el frontend.

Se debe especificar la misma URL ("/api/v1/files" en este caso) y un parámetro de tipo MultipartFile. Es muy importante que este @RequestParam tenga el mismo valor (("file")) que el itemAlias en el frontend (línea 32 del app.component.ts)

```
package com.serikat.anonymizer.controller;

import java.io.IOException;..

@RestController
public class UploadFileController {

  private final FileService fileService;

  @Autowired
  public UploadFileController(FileService fileService) {
        this.fileService = Objects.requireNonNull(fileService);
  }

  @PostMapping(value = "/api/v1/files")
  @ResponseStatus(HttpStatus.OK)
  public void handleFileUpload(@RequestParam("file") MultipartFile file) throws IOException {

        System.out.println("Received file: " + file.getName());
        fileService.storeFile(file);
  }

}
```

Por último, es necesario realizar la configuración **CORS** (aunque tanto el frontend y el backend están en el mismo host -localhost en nuestro caso-, al tratarse de puertos diferentes, tenemos dominios diferentes):

Para ello, podemos añadir el siguiente bean (línea 13) en la clase principal:

```
1  package com.serikat.anonymizer;
2
3  import org.springframework.boot.SpringApplication;..
4
5  @SpringBootApplication
6  @EnableDiscoveryClient
7  public class SktAnonymizerFileUploadBackendApplication {
8
9    public static void main(String[] args) {
10     SpringApplication.run(SktAnonymizerFileUploadBackendApplication.class, args);
11   }
12
13   @Bean
14   public WebMvcConfigurer corsConfigurer() {
15     return new WebMvcConfigurer() {
16         @Override
17         public void addCorsMappings(CorsRegistry registry) {
18           // registry.addMapping("/**").allowedOrigins("*")
19           //         .allowedMethods("GET", "POST", "PUT", "DELETE")
20           // better we specify the mapping and the allowed origin:
21           registry.addMapping("/api/v1/files")
22              .allowedOrigins("http://localhost:4200")
23              .allowedHeaders("Content-Type", "Access-Control-Allow-Origin",
24                     "Access-Control-Allow-Methods", "Access-Control-Allow-Headers",
25                     "Authorization")
26              .exposedHeaders("Content-Type", "Access-Control-Allow-Origin",
27                     "Access-Control-Allow-Methods","Access-Control-Allow-Headers",
28                     "Authorization")
29              .allowCredentials(true);
30           }
31     };
32   }
33
34 }
```

Bibliografía

https://morioh.com/p/bf88b4dfbb2c

y buscar mucho en google :)