**07.04.2023**

İlgili günde kullanıcıların yüklemek veya indirmek istediği dosyalar için API servisleri tanımlanmıştır. Bunun yanında kullanıcı “Dosya İşlemleri” sayfasına girdiği zaman kendisine ait olan dosyaların listelenmesi ve bu dosyaların silinmesi özelliği geliştirilmiştir.

Kullanıcının sisteme dosya yükleyip indirebilmesi için Client tarafından Server tarafına GET veya POST isteği atılmalıdır. Bu isteğin kabul edilmesi için Controller katmanında FileApiController isminde bir sınıf oluşturulup @Controller anotasyonu verilmiştir. Sınıfın bir önceki gün oluşturulan StorageService ismindeki servis sınıfına erişebilmesi için FileApiController sınıfına field olarak bu servis sınıfından bir field tanımlanmıştır. Tanımlanan bu field’a @Autowired anotasyonu verilmiştir.

Kullanıcının sisteme dosya yükleyebilmesi için Client tarafından Server tarafına POST isteği atılmalıdır. Dosya yükleme için gönderilen bu POST isteğinin fonksiyon tarafından kabul edilmesi için aşağıdaki fonksiyon tanımlanmıştır. Fonksiyon parametre olarak MultiPartFile tipinde bir dosya kabul etmektedir. Bu parametreyi “image” isminde beklemektedir. Kabul ettiği bu dosyayı StorageService sınıfındaki uploadImage fonksiyonuna göndermektedir. Yükleme işlemi tamamlandıktan sonra kullanıcı aynı sayfaya yönlendirilerek kullanıcının dosyanın yüklendiği görmesi sağlanmaktadır.

@PostMapping("/uploadImage")

public String uploadImage(@RequestParam("image") MultipartFile file) throws IOException {

storageService.uploadImage(file);

return "redirect:/home";

}

Kullanıcının sistemden dosya indirebilmesi için Client tarafından Server tarafına GET isteği atılmalıdır. Dosya indirme işlemi için gönderilen bu GET isteğinin fonksiyon tarafından kabul edilmesi için aşağıdaki fonksiyon tanımlanmıştır. Fonksiyon parametre olarak Long tipinde bir ID değeri almaktadır. Kabul ettiği bu ID değerini StorageService sınıfındaki downloadImage fonksiyonuna göndermekte ve yanıt olarak bir byteArray almaktadır. Bu byteArray’i HTTP paketinin body (gövde) kısmına ekleyerek kullanıcıya göndermektedir.

@GetMapping("/get/{id}")

public ResponseEntity<?> downloadImage(@PathVariable Long id){

byte[] imageData=storageService.downloadImage(id);

return ResponseEntity.status(HttpStatus.OK)

.contentType(MediaType.valueOf("image/png"))

.body(imageData);

}

Kullanıcı “Dosya İşlemleri” sayfasına girdiği zaman kendisine ait dosyaların listeleme tablosunda gözükmesi için “/home” isteğini kabul eden fonksiyonda aşağıdaki gibi güncelleme yapılmıştır.

@GetMapping("/home")

public String listUsers(Model model) {

List<ImageData> files = storageRepo.findByFileOwner(SecurityContextHolder.getContext().getAuthentication().getName());

model.addAttribute("files", files);

return "home";

}

Front-end (Ön yüz) kısmına gönderilen bu dosyaların Thymeleaf tarafından iterate edilmesi için home.html dosyasında aşağıdaki düzenlemeler yapılmıştır.

<tr th:each="file : ${files}">

<td>

<ul class="action-list">

<li><a th:href="@{/get/{id}(id=${file.id})}" th:download="\*{file.name}" class="btn btn-primary"><i class="fa fa-download"></i></a></li>

<li><a th:href="@{/delete/{id}(id=${file.id})}" class="btn btn-danger"><i class="fa fa-times"></i></a></li>

</ul>

</td>

<td>[[\*{file.id}]]</td>

<td>[[\*{file.name}]]</td>

</tr>

Kullanıcının dosya listesinde bulunan herhangi bir dosyayı dosyaya ait ID bilgisi ile silebilmesi için gerekli olan StorageService sınıfında aşağıdaki gibi tanımlanmıştır.

public void deleteImageDataById(Long id)

{

storageRepo.deleteById(id);

}

Kullanıcının dosya silme talebinin kabul edilmesi için FileApiController sınıfında “/delete/{id}” isteklerini kabul eden bir fonksiyon aşağıdaki gibi tanımlanmıştır. Dosya silme silme işlemi gerçekleştikten sonra kullanıcının sonucu görmesi için tekrardan “Dosya İşlemleri” sayfasına yönlendirme yapılmaktadır.

@GetMapping(value = "/delete/{id}")

public String deleteFile(@PathVariable("id")Long id)

{

storageService.deleteImageDataById(id);

return "redirect:/home";

}

Projenin bitmiş halinin ekran görüntülerine EK-20, EK-21, EK-22, EK-23, EK-24 görsellerinde yer verilmiştir.