**06.04.2023**

İlgili günde kullanıcıların sisteme dosya yükleyip indirmesini sağlayacak olan servisler geliştirilmiştir.

Kullanıcıların dosyalarının saklanması ve saklanan bu dosyaların tekrardan kullanıcıya sunulabilmesi veritabanında saklanması gerekmektedir. Projede ORM(Object Relational Mapping) yapısı kullanıldığından her bir resim nesnesini temsil etmesi için Entity katmanında ImageData isminde bir sınıf oluşturulup @Entity anotasyonu verilmiştir. Bu sınıfa verilen @Table(name = "image\_data") anotasyonu sayesinde image\_data isminde bir tabloyu temsil ettiği bildirilmiştir. Sınıf içerisinde id, fileowner, name, type, image\_data isminde field’lar tanımlanmıştır. Fileowner field’ı dosyanın sahibi olan kişinin username değerini tutmaktadır. image\_data field’ı resmin byte dizisine çevrilmiş halini saklamaktadır.

Veritabanında saklanan resim verilerine JPA ile erişilebilmesi için Repository katmanında StorageRepository isminde JpaRepository sınıfını kalıtım alan bir interface tanımlanmıştır. Bu interface içerisinde dosya sahibinin ismiyle arama yapılmasını sağlayacak gövdesiz metot aşağıdaki gibi @Query anotasyonuyla tanımlanmıştır. Bu Query içerisine koşullu bir arama sorgusu yazılmıştır.

@Query("SELECT f FROM ImageData f WHERE f.fileowner = ?1")

public List<ImageData> findByFileOwner(String username);

Kullanıcının upload veya download edilmesi için belirttiği dosya ile ilgili veritabanı üzerinde işlem yapılması için Service katmanı oluşturulmuştur. Bu katman StorageService isminde @Service anotasyonuna sahip bir sınıf oluşturulmuştur. Bu servis içerisinde dosyanın veritabanına yüklenmesini sağlayacak olan fonksiyon aşağıdaki gibi tanımlanmıştır. Fonksiyon kaydetme işlemi için Repository katmanındaki StorageRepository arayüzünün sahip olduğu save metodunu kullanmaktadır. Kaydetme işlemi sırasında resim verisi üzerinde sıkıştırma işlemi uygulanmaktadır.

@Autowired

private StorageRepository storageRepo;

public void uploadImage(MultipartFile file) throws IOException {

ImageData imageData = storageRepo.save(ImageData.builder()

.fileowner(SecurityContextHolder.getContext().getAuthentication().getName())

.name(file.getOriginalFilename())

.type(file.getContentType())

.imageData(ImageUtils.compressImage(file.getBytes())).build());

}

Aynı servis içerisinde dosyanın veritabanından çekilmesini sağlayacak olan fonksiyon aşağıdaki gibi tanımlanmıştır. Fonksiyon öncelikle image\_data tablosu üzerinde kendisine gönderilen ID bilgisi ile arama işlemi yaparak dosyayı bulup decompress (çözümleme) işlemi uygulamaktadır.

public byte[] downloadImage(Long id){

Optional<ImageData> dbImageData = storageRepo.findById(id);

if(dbImageData.isPresent()){

byte[] images=ImageUtils.decompressImage(dbImageData.get().getImageData());

return images;

}

else return null; }