

Project Name : **House Rental Management System Api**



Öğrenci Numarası: 20290284

Öğrenci Adı SOYADI: Mert Patlar

Danışman Adı SOYADI: Öğr. Gör. Enver BAĞCI

Bu Proje Hakkında

Bu proje, insanların istedikleri gayrimenkul ve konaklamayı hızlı ve kolay bir şekilde bulmaları için tasarlanmıştır. Aynı zamanda mülk sahiplerinin de uygun kiracıları bulmalarına yardımcı olan bir platformdur.

Genel Görünüm

- Proje dosya yapısı 5 ana katmana ayrılmıştır:
 - **controller**: İsteklerin karşılandığı yer.
 - **model**: Veritabanı modelleri veya veri taşıma objeleri.
 - **repository**: Veri erişim katmanı.
 - **service**: İş mantığı.
 - **storage**: Uygulamaya özel yardımcı sınıflar.

Controller Katmanı

Dosyalar:

- `AdminController.java`: Yönetici işlemlerinin kontrolü.
- `LoginController.java`: Kullanıcı giriş işlemleri.
- `MailSendController.java`: Mail gönderim işlemleri.

Görevler:

- Kullanıcıdan gelen HTTP isteklerini karşılamak.
- İş mantığını Service katmanına yönlendirmek.

Model Katmanı

Dosyalar:

- `AdvertisingForm.java`: Reklam formlarını temsil eder.
- `MailModal.java`: Mail verilerini tutar.
- `Status.java`: Uygulama durumunu temsil eden yapı.
- `UserModel.java`: Kullanıcı bilgilerini temsil eder.

Görevler:

- Veritabanı ve iş mantığı arasında veri taşımak.

```

@PostMapping("/saveadvertising")
AdvertisingForm advertisingForm(@RequestBody AdvertisingForm advertisingForm) {
    return advertiseService.save(advertisingForm);
}

@PostMapping("/saveadvertising_withfile")
public ResponseEntity<Map> saveFormData(@ModelAttribute AdvertisingForm advertisingForm,
    @RequestParam("file") MultipartFile file, @RequestParam("user_id") long userId) {
    Map<String, Object> map = new HashMap<String, Object>();
    try {

        String fileName = fileStorageService.storeFile(file);

        String fileDownloadUri = ServletUriComponentsBuilder.fromCurrentContextPath().path("/downloadFile/")
            .path(fileName).toUriString();

        advertisingForm.setImages(fileName);
        advertisingForm.setImagesUri(fileDownloadUri);

        UserModel user = userService.findById(userId).get();
        advertisingForm.setUser(user);

        advertisingForm = advertiseService.save(advertisingForm);
        map.put("status", "Success");
        map.put("data", advertisingForm);
        map.put("message", "Data saved successfully");
        return ResponseEntity.ok(map);

    } catch (Exception e) {
        map.put("status", "failed");
        map.put("data", null);
        map.put("message", e.getLocalizedMessage());
        return ResponseEntity.status(500).body(map);
    }
}

```

Örnek olarak
AdminController.java
 dosyasında bulunan
 dosyasız ve dosyalı
 şekilde ilan
 paylaşmamızı
 sağlayan Admin
 metodları.

```

@Entity
@Table(name = "advertisingform")
public class AdvertisingForm {

    @Id()
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(unique = true, length = 20)
    private long advertisingId;

    @Transient
    @JsonIgnore
    MultipartFile file;

    private String location;
    private String type;
    private String status;
    private int bedrooms;
    private int bathrooms;
    private int price;
    private int sqft;
    private String additionalInformation;
    private String images;
    private String imagesUri;

    private String img2;
    private String imgUri2;

    private String img3;
    private String imgUri3;

    private String video;
    private String videoType;

    private double lat;
    private double lng;

    @ManyToOne(optional = false)
    @JoinColumn(name = "adminid", nullable = true)
    private UserModel user;

```

Örnek olarak **AdvertisingForm.java** dosyasında bulunan ilan modeli.

Repository Katmanı

Dosyalar:

- `AdvertisingService.java`: Reklam verileriyle ilgili işlemler.
- `EmailService.java`: Mail gönderim işlemleri.
- `UserService.java`: Kullanıcı verileriyle ilgili işlemler.

Görevler:

- Veritabanı işlemlerini gerçekleştirmek.

Service Katmanı

Dosyalar:

- `FileStorageService.java`: Dosya yükleme ve yönetimi.

Görevler:

- İş mantığını gerçekleştirir.
- Veritabanı veya diğer dış sistemlerle iletişim kurar.

```

@Repository
public interface AdvertisingService extends JpaRepository<AdvertisingForm, Long>{

    @Query("Select add from AdvertisingForm add where add.user.id = ?1")
    List<AdvertisingForm> findById(long id);

    @Query("SELECT a FROM AdvertisingForm a WHERE a.type= ?1")
    public List<AdvertisingForm> categorytype(String type);

    @Query("SELECT a from AdvertisingForm a where a.location like %:searchText% or "
        + "a.type like %:searchText% or a.status like %:searchText% or "
        + "a.price like %:searchText%")
    public List<AdvertisingForm> searchAdvertise(String searchText);

    @Query("Select add from AdvertisingForm add where add.location = ?1 and"
        + " (add.price BETWEEN ?2 AND ?3) and"
        + " (add.sqft BETWEEN ?4 AND ?5) and"
        + " add.type =?6 and"
        + " add.status =?7 and"
        + " add.bedrooms =?8 and"
        + " add.bathrooms =?9")
    List<AdvertisingForm> findByadvancedSearchingggggg(String location, int price, int price2, int sqft, int sqft2, String type,
        String status, int bedrooms, int bathrooms);

    @Query("Select p from AdvertisingForm p where p.price between ?1 and ?2")
    List<AdvertisingForm> findByadvancedSearching(int price, int price2);

}

```

AdvertisingService.java
dosyasının görünümü bu
şekildedir.

Storage Katmanı

Dosyalar:

- `HouseRentalManagement.java`: Özel bir ev kiralama yönetim sistemi örneği.
- `SendMailExample.java`: Mail gönderimiyle ilgili örnek kod.
- `ServlInitializer.java`: Servlet başlatma işlemleri.

Kaynaklar ve Dosya Yükleme

- **resources**: Proje içinde kullanılan ek kaynaklar.
- **uploads**: Yüklenen dosyaların depolandığı dizin.

Kapanış

Projenin görünüşü ve işlemleri hakkındaki sunu için [buraya](#) tıklayabilirsiniz.

SON