

# Proje Adı: Kitap Yönetimi REST API

**Proje Amacı:** Kullanıcıların kitap ekleyebileceği, listeleyebileceği, güncelleyebileceği ve silebileceği bir REST API oluşturmak. Ayrıca, kullanıcı kayıt ve giriş işlemlerini session tabanlı güvenlik ile yönetmek.

**Proje Süresi:** Proje teslim sürtesi 10 gündür.

## Uygulamadaki özellikler ve Kurallar:

- Kullanıcı Register
- Kullanıcı Giriş
- Kitap Ekleme
- Kitap Listleme (Sayfalamalı)
- Bir Kitap Detayı
- Kitap Arama (Sayfalamalı)
- Kitap Silme
- Kullanıcı login olmadan yukarıki hiç bir maddeyi yapamamalıdır.
- Kullanıcı login olduktan sonra sadece kendi kitaplarına hakimiyet sağlayabilmelidir, örn: başka birinin kitabını silmek istediğinde uygulama bu kitap size ait değil gibi bir uyarı vermelidir, bu örnek diğer tüm başlıklar için geçerlidir.
- Tüm adımlarda Validation ile veri denetlenmelidir.

## Teknolojiler:

- Spring Boot
- Security (Session tabanlı)
- Spring Data JPA (H2 veritabanı ile)
- Validations
- Maven
- Postman

## Proje Yapısı:

```
kitap-kutuphanesi-api/  
├── src/main/java/  
│   ├── com/example/kitapkutuphanesi/  
│   │   ├── controller/  
│   │   │   ├── AuthController.java  
│   │   │   └── BookController.java  
│   │   ├── dto/  
│   │   │   ├── BookDTO.java  
│   │   │   ├── UserDTO.java  
│   │   │   ├── LoginRequest.java  
│   │   │   └── RegisterRequest.java  
│   │   ├── entity/  
│   │   │   ├── Book.java  
│   │   │   └── User.java  
│   │   ├── exception/  
│   │   │   ├── GlobalExceptionHandler.java  
│   │   │   └── ResourceNotFoundException.java  
│   │   ├── filter/  
│   │   │   ├── AuthFilter.java  
│   │   │   └── FilterConfig.java  
│   │   ├── repository/  
│   │   │   ├── BookRepository.java  
│   │   │   └── UserRepository.java  
│   │   └── service/  
│   │       └── AuthService.java
```

```
├── BookService.java
│   ├── config/
│   │   ├── SecurityConfigFilter.java
│   │   └── KitapKutuphanesiApplication.java
│   └── resources/
│       ├── application.properties
│       └── pom.xml
├── .gitignore
├── README.md
└── Postman_Collection.json
```

## Proje Detayları:

### 1. Entity (Varlık) Katmanı:

- Book.java: Kitap bilgilerini (id, başlık, yazar, yayın yılı) tutar.
- User.java: Kullanıcı bilgilerini (id, kullanıcı adı, şifre) tutar.

### 2. DTO (Veri Transfer Nesnesi) Katmanı:

- BookDTO.java: Kitap bilgilerini dış dünyaya sunmak için kullanılır.
- UserDTO.java: Kullanıcı bilgilerini dış dünyaya sunmak için kullanılır.
- LoginRequest.java: Kullanıcı giriş isteği için kullanılır.
- RegisterRequest.java: Kullanıcı kayıt isteği için kullanılır.

### 3. Repository (Veri Deposu) Katmanı:

- BookRepository.java: Kitap veritabanı işlemlerini yönetir.
- UserRepository.java: Kullanıcı veritabanı işlemlerini yönetir.

### 4. Service (Servis) Katmanı:

- BookService.java: Kitap işlemleri (ekleme, listeleme, güncelleme, silme) mantığını içerir.
- AuthService.java: Kullanıcı kayıt ve giriş işlemlerini yönetir.

### 5. Controller (Denetleyici) Katmanı:

- BookController.java: Kitap API uç noktalarını yönetir.
- AuthController.java: Kullanıcı kayıt ve giriş API uç noktalarını yönetir.

### 6. Exception (İstisna) Katmanı:

- GlobalExceptionHandler.java: Genel istisna yönetimini sağlar.
- ResourceNotFoundException.java: Kaynak bulunamadığında fırlatılır.

### 7. Filter (Filtre) Katmanı:

- AuthFilter.java: Session tabanlı kimlik doğrulama filtresi.
- FilterConfig.java: Filtre yapılandırması.

### 8. Config (Yapılandırma) Katmanı:

- SecurityConfig.java: Spring Security yapılandırması.

### 9. Validation (Doğrulama):

- DTO'lar üzerinde Hibernate Validator ile veri doğrulama işlemleri yapılır.

### 10. Postman Koleksiyonu:

- API uç noktalarını test etmek için bir Postman koleksiyonu hazırlanır.

### 11. Github Deposu:

- Proje, Github'da bir depoda saklanır ve öğrencilerle paylaşılır.