



# Java Spring Boot Bootcamp


HAZIRLAYAN

MEHMET ALTAN

# HTTP STATÜ KODLARI

- ▶ HTTP isteklerinin ve yanıtlarının durumunu belirtmek için "HTTP durum kodları" veya "HTTP status kodları" denilen kodlar kullanılır.
- ▶ 1xx (Informational): İstek alındı ve işleniyor.
  - ▶ 100 Continue: Sunucu, isteği aldı ve işlemeye devam edebilir.
  - ▶ 101 Switching Protocols: Sunucu, istemcinin protokolü değiştirmesini istiyor.
- ▶ 2xx (Successful): İstek başarılı bir şekilde işlendi ve yanıtlandı.
  - ▶ 200 OK: İstek başarılı bir şekilde tamamlandı.
  - ▶ 201 Created: İstek sonucunda yeni bir kaynak oluşturuldu.
  - ▶ 204 No Content: İstek başarılı, ancak yanıtta içerik yok.

- ▶ 3xx (Redirection): İsteğin tamamlanabilmesi için ek adımlar gereklidir.
  - ▶ 301 Moved Permanently: Kaynak kalıcı olarak yeni bir konumda bulunuyor.
  - ▶ 302 Found (Moved Temporarily): Kaynak geçici olarak başka bir konumda bulunuyor.
  - ▶ 304 Not Modified: İstemci tarafından önbelleğe alınmış veri hala geçerli.
- ▶ 4xx (Client Error): İstemci tarafından yapılan istekte hata var.
  - ▶ 400 Bad Request: İstek anlaşılamadı veya geçersiz.
  - ▶ 401 Unauthorized: Kimlik doğrulama gerekiyor veya kimlik doğrulaması başarısız.
  - ▶ 403 Forbidden: İstek reddedildi, kaynağa erişim izni yok.
  - ▶ 404 Not Found: İstek kaynak bulunamadı.

- 
- ▶ 5xx (Server Error): Sunucu isteği yerine getiremedi.
    - ▶ 500 Internal Server Error: Genel bir sunucu hatası.
    - ▶ 502 Bad Gateway: Sunucu, geçersiz bir yanıt aldı.
    - ▶ 503 Service Unavailable: Sunucu geçici olarak hizmet veremiyor.
    - ▶ 504 Gateway Timeout: Aracı sunucu zaman aşımına uğradı.



# ResponseEntity

- ▶ Spring Boot'ta kullanılan ResponseEntity sınıfı, HTTP yanıtlarını oluşturmak ve özelleştirmek için kullanılan bir sınıftır. Bu sınıf, Spring framework'ün sunduğu çeşitli HTTP yanıt detaylarını kontrol etmeye ve özelleştirmeye yardımcı olur. Genellikle Spring Boot uygulamalarında kontrolcü (controller) metodlarında kullanılır.
- ▶ ResponseEntity sınıfı, HTTP yanıtlarını HTTP durum kodları (status code), başlık (header) bilgileri ve yanıt gövdesi (response body) gibi bileşenlerle oluşturmanıza olanak tanır. Bu, özellikle farklı durumlarda farklı HTTP yanıtları üretmek veya belirli HTTP başlıklarını ayarlamak istediğinizde oldukça kullanışlıdır.

# REST API Parametre Türleri

- ▶ Path Variables
  - ▶ Belirli bir kaynağın altında bulunan ver ve kaynaklara kullanım amacıyla sıkça kullanılır.
- ▶ **Query Parameters**

```
@GetMapping(value = "/search")
public ResponseEntity<Object> searchBooks(@RequestParam(name = "categoryId") int categoryId,
                                          @RequestParam(name = "name", required = false) String name) {
    ArrayList<BookDto> books = bookService.searchBooks(categoryId,name);
    if (books.size()>0) {
        return ResponseEntity.ok(books);
    } else {
        return ResponseEntity.status(HttpStatus.NOT_FOUND)
            .header(headerName:"App-Health", ...headerValues:"green")
            .body(body:"Aranılan kayıt bulunamadı.");
    }
}
```

► Request Body (İstek Gövdesi)

```
@PostMapping(value = "")  
public ResponseEntity<Integer> addBook(@RequestBody BookDto bookDto) {  
    return ResponseEntity.ok(bookService.addBook(bookDto));  
}
```

## ► Request Headers (İstek Başlıkları)

```
@GetMapping("/greet")
public ResponseEntity<String> greetUser(@RequestHeader("User-Agent") String userAgent) {
    return ResponseEntity.ok(body:"Merhaba, kullanıcı!");
}
```





# Ödev

- ▶ Cookie Values (Çerez Değerleri) Nedir Araştırınız?
- ▶ Matrix Variables (Matrix Değişkenleri) nedir araştırınız ve kütüphane uygulamamızda bir örnek tanımlayınız.

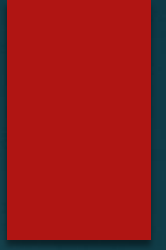
# REST API Dökümantasyonu

- ▶ Bir RESTful servisin nasıl kullanılacağını ve hangi endpointlerin, isteklerin ve yanıtların mevcut olduğunu açıklayan bir dokümandır.
- ▶ Bu dokümantasyon, geliştiricilerin ve diğer hizmet tüketenlerin (istemcilerin) API'nizi anlamalarına, entegre etmelerine ve kullanmalarına yardımcı olur. REST API dökümantasyonunun amacı şunlardır

- 
- ▶ API'nin Kullanımını Açıklama: API'nin nasıl kullanılacağını, hangi endpointlerin mevcut olduğunu, hangi istekleri göndermek gerektiğini ve nasıl yanıtlar alınacağını ayrıntılı bir şekilde anlatır. Bu, geliştiricilerin hızla API'yi kullanmaya başlamasına yardımcı olur.
  - ▶ Parametreler ve Veri Formatları: API isteklerinde kullanılması gereken parametrelerin, veri formatlarının (JSON, XML vb.) ve medya tiplerinin nasıl olması gerektiğini belirtir.
  - ▶ Hataları ve Durum Kodlarını Açıklama: API tarafından döndürülen hataları ve durum kodlarını (HTTP durum kodları gibi) açıklar. Bu, istemcilerin istekleri başarılı bir şekilde işlemek için hangi hataların meydana gelebileceğini anlamalarını sağlar.

- 
- ▶ Örnek İstekler ve Yanıtlar: API isteklerinin ve yanıtlarının örneklerini sağlar, böylece geliştiriciler gerçek dünya senaryolarını anlayabilir ve uygulayabilir.
  - ▶ Güvenlik Bilgileri: API'ye erişmek için gereken kimlik doğrulama yöntemlerini ve güvenlik önlemlerini açıklar.





- ▶ Sürüm Bilgisi: API'nin sürüm numarası, yayın tarihi ve değişiklikler hakkında bilgi içerebilir.
- ▶ Rate Limiting: API kullanımının sınırlamalarını ve sorgu sınırını (rate limiting) belirten bilgiler içerebilir.
- ▶ Sınırlamalar ve Kısıtlamalar: API'nin hangi durumlarda sınırlamaları veya kısıtlamaları olduğunu belirtir.

# Uygulama – Swagger ile Dökümantasyon

