

# Spring Boot ve Spring Data REST



# Spring Data REST

- Spring Data **Repository**'lerini otomatik olarak **REST resource**'ları şeklinde sunmayı sağlar
- İstemcinin repository'ler tarafından sunulan fonksiyonalliteyi keşfetmesi için **hypermedia** kullanılır
- REST API'si üzerinde kolayca işlem yapabilmek için **HAL** (hypermedia application language) **Browser**'da mevcuttur

# Spring Data REST Mimarisi

Spring Data REST

Spring Data (JPA)

Spring Web MVC

# Spring Data REST

- Devreye almak için **pom.xml**'de aşağıdaki bağımlılık tanımlarını yapmak yeterlidir

```
<dependency>  
    <groupId>org.springframework.boot</groupId>  
    <artifactId>spring-boot-starter-data-rest</artifactId>  
</dependency>
```

```
<dependency>  
    <groupId>org.springframework.data</groupId>  
    <artifactId>spring-data-rest-hal-browser</artifactId>  
</dependency>
```

# Spring Data REST

- Default olarak Data REST API kök (/) **URI**'dan sunulur
- İstenirse bu değiştirilebilir ve bir **base path** tanımlanabilir

application.properties



```
spring.data.rest.base-path=/rest
```

# Spring Data REST

- Repository bean'lerine erişim **domain sınıf isminin küçük harfle başlayan çoğul hali** ile gerçekleştirilir

```
public interface OwnerRepository extends JpaRepository<Owner, Long> {
}
```

- Yukarıdaki gibi bir repository'nin fonksiyonlarına erişim **/owners** request URI üzerinden sağlanır
- /owners/{id}** ile spesifik bir domain nesnesine resource olarak erişilebilir

# REST Statü Kodları

- GET isteklerinde **200**
- Yeni bir resource yaratan POST ve PUT isteklerinde **201**
- Response body dönmeyen ve resource update gerçekleştiren PUT, PATCH, DELETE isteklerinde **204**
- Response body dönen ve resource update gerçekleştiren PUT isteklerinde **200**

# HAL Browser

← → ↺ 🏠

localhost:8080/rest/browser/index.html#/rest

🔍 ⋮ 🛡️ ☆

🔍 📄 📄 📄 >> ☰

The HAL Browser (for Spring Data REST)    Go To Entry Point    About The HAL Browser (for Spring Data REST)

## Explorer

/rest

Go!

## Custom Request Headers

## Properties

{  
}

## Links

rel	title	name / index	docs	GET	NON-GET
persons				👉	👈
vets				→	👈
profile				→	👈

## Inspector

### Response Headers

200 success

Cache-Control: no-cache, no-store, max-age=0, must-revalidate  
Content-Type: application/hal+json; charset=UTF-8  
Date: Wed, 06 Jun 2018 11:30:30 GMT  
Expires: 0  
Pragma: no-cache  
Transfer-Encoding: chunked  
X-Content-Type-Options: nosniff  
X-Frame-Options: DENY  
X-XSS-Protection: 1; mode=block

### Response Body

{  
 "\_links": {  
 "persons": {  
 "href": "http://localhost:8080/rest/persons?page,size,sort",  
 "templated": true  
 },  
 "vets": {  
 "href": "http://localhost:8080/rest/vets"  
 },  
 "profile": {  
 "href": "http://localhost:8080/rest/profile"  
 }  
 }  
}



# Resource Discoverability

- Resource API'sinin kendi kendine keşfedilebilir olması (**HATEOAS**) sağlanmıştır
- Spring Data REST, resource linklerini **HAL** formatında oluşturur
- Linkler **JSON** şeklinde dönen resource içerisinde property şeklinde tanımlanır

# Resource Discoverability

- `http://localhost:8080/hal/vets/1` request URI sonucu dönen response body içerisinde **link bilgileri** şu şekilde yer alır:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "_links": {
    "self": {
      "href": "http://localhost:8080/hal/vets/1"
    },
    "user": {
      "href": "http://localhost:8080/hal/vets/1"
    },
    "profile": {
      "href": "http://localhost:8080/hal/vets/1/profile"
    }
  }
}
```

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

