

Spring Data Projesi ve Web MVC Desteği



Spring Data WebSupport

- Spring Data projesi, **Web MVC katmanı ile** yakın bir **entegrasyon** kabiliyeti de sunar
- Bunun için projede Spring MVC kabiliyetinin de devrede olması gerekir

```
@Configuration
@EnableWebMvc
@EnableSpringDataWebSupport
public class AppConfig {
}
```

Java tabanlı config

XML tabanlı config

```
<beans...>
  <bean class="org.springframework.data.web.config.SpringDataWebConfiguration"/>

  <bean class="org.springframework.data.web.config.HateoasAwareSpringDataWebConfiguration"/>
</beans>
```

Spring Data WebSupport

- **Request parametresi veya path variable** ile belirtilen domain nesneleri otomatik biçimde **Repository bean ile resolve edilerek** controller metotlarına geçilir

Arka tarafta register edilen konfigürasyon sayesinde path variable ile PK'sı belirtilen Vet entity'si Repository ile bulunur ve Controller metoduna geçilir

```
@Controller
@RequestMapping("/vets")
public class VetController {

    @GetMapping("/{id}")
    public String showVetForm(@PathVariable("id") Vet vet, Model model) {
        model.addAttribute("vet", vet);
        return "editVetForm";
    }
}
```

Spring Data WebSupport

- **Pageable ve Sort nesneleri de request param değerleri üzerinden otomatik olarak resolve edilip controller metotlarına geçilebilir**

```
@Controller
@RequestMapping("/vets")
public class VetController {
    @Autowired
    private VetRepository vetRepository;

    @GetMapping
    public String displayVets(Model model, Pageable pageable) {
        Page<Vet> vets = vetRepository.findAll(pageable);
        model.addAttribute("vets", vets);
        return "displayVets";
    }
}
```

Arka tarafta register edilen
HandlerMethodArgumentResolver nesneleri
sayesinde request parametreleri ile geçen
Pageable ve Sort nesneleri otomatik olarak
resolve edilir



- Pageable için kullanılan **request parametreleri**
 - page: sayfa numarası, 0'dan başlar
 - size: sayfada yer alacak kayıt sayısı, default:20
 - sort: sıralama yapılacak property, default asc
 - sort=firstname&sort=lastname,desc

HATEOAS Desteği

- **Pageable collection**'ları REST response'u olarak dönmeyi sağlar
- Bunun için **PagedResources** sınıfı kullanılır
- Pageable response, REST ResponseEntity içerisine PagedResources nesnesi ile wrap edilerek konur
- Dönen response content içerisinde **page metadata**, ve bir **sonraki sayfaya link** yer alır

HATEOAS Desteği

- PageResources nesnesini oluşturmak için **PagedResourcesAssembler** bean'i mevcuttur
- Spring Data'nın HATEOAS desteği devreye alındığında bu bean **Controller metot'larında input argüman** olarak tanımlanabilir

HATEOAS Desteği

```
@RestController
@RequestMapping("/vets")
public class VetRestController {

    @Autowired
    private VetRepository vetRepository;

    @GetMapping
    public ResponseEntity<PagedResources<Resource<Vet>>> findVets(
        Pageable pageable,
        PagedResourcesAssembler<Vet> pagedResourcesAssembler) {

        Page<Vet> page = vetRepository.findAll(pageable);

        PagedResources<Resource<Vet>> pagedResources =
            pagedResourcesAssembler.toResource(page);

        ResponseEntity<PagedResources<Resource<Vet>>> responseEntity =
            new ResponseEntity<PagedResources<Resource<Vet>>>(
                pagedResources, HttpStatus.OK);

        return responseEntity;
    }
}
```


HATEOAS Desteği

```
{ "links" : [ { "rel" : "next",
                "href" : "http://localhost:8080/vets?page=1&size=20" }
            ],
  "content" : [
    ... // 20 Vet instances rendered here
  ],
  "pageMetadata" : {
    "size" : 20,
    "totalElements" : 50,
    "totalPages" : 3,
    "number" : 0
  }
}
```

İletişim

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

