

# Spring Data Projesi ve Sayfalama (Paging) & Sıralama (Sorting)



# PagingAndSortingRepository Arayüzü

- Spring Data projesi, sayfalama ve sırlama işlemleri için **PagingAndSortingRepository** arayüzünü sunmaktadır

```
public interface PagingAndSortingRepository<T, ID extends  
Serializable> extends CrudRepository<T, ID> {
```

```
    Iterable<T> findAll(Sort sort);
```

```
    Page<T> findAll(Pageable pageable);
```

```
}
```

Sort nesnesi içerisinde hangi property'lerin ASC veya DESC sıralanarak getirileceği tanımlanır

Sorgu sonucu belirtilen sayfadaki entity'leri içerir

Pageable nesnesi paging hakkında pageNumber, pageSize, sort, next, previous page gibi bilgiler içerir

# PagingAndSorting Repository Arayüzü

- Sayfalama ve sıralama kabiliyetleri için **bu arayüzden türeyen** Repository arayüzleri tanımlamamız gerekir
- **JpaRepository** arayüzü halihazırda bu arayüzden türemektedir

```
public interface VetRepository
    extends JpaRepository<Vet, Long> {

    Page<Vet> findByLastname(String lastname, Pageable pageable);

}
```

# Sayfalama (Paging)

- Sayfalama kabiliyetine sahip metotları çağırmak için Pageable arayüzünü implement eden **PageRequest** nesnesi oluşturmak gerekir

Current page numarası,  
sıfırdan başlar

Sayfa içerisinde yer  
alacak kayıt sayısı

```
PageRequest pageRequest = PageRequest.of(0, 10);
```

```
Page<Vet> page = vetRepository.findByLastname("Doe", pageRequest);
```



Dönen page nesnesi içerisinde toplam sayfa (Page) sayısı  
ve toplam kayıt sayısı (Vet) de yer alır

# Sayfalama (Paging)

- Page nesnesi içerisindeki toplam sayfa sayısı ve toplam kayıt **sayısı ilave bir count sorgusu** ile tespit edilmektedir
- Bu bilgilere ihtiyaç yoksa find metodu **Slice** veya **List<T>** dönebilir

```
public interface VetRepository extends JpaRepository<Vet, Long> {  
    Slice<Vet> findByLastname(String lastname, Pageable pageable);  
  
    List<Vet> findByFirstname(String firstname, Pageable pageable);  
}
```

# Sıralama (Sorting)

- PagingAndSorting arayüzü sorgu sonuçlarının **belirtilen property'ler üzerinden sıralanarak** getirilmesini de sağlamaktadır

```
Sort sortByLastnameAsc = Sort.by("lastname").ascending();
```

```
List<Vet> vets = vetRepository.findAll(sortByLastnameAsc);
```



```
Sort sortByLastnameAscAndGradYearDesc =  
    sortByLastnameAsc.and(Sort.by("graduationYear").descending());
```

```
vets = vetRepository.findAll(sortByLastnameAscAndGradYearDesc);
```

# İletişim

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

