Spring Data Projesi ve Projeksiyon Kabiliyeti

Projeksiyon Yöntemleri

- Spring Data JPA farklı projeksiyon yöntemleri sunar
  - Arayüz tabanlı projeksiyon
  - Sınıf tabanlı projeksiyon
- **Arayüz tabanlı** projeksiyon yöntemi de kendi içinde **iki farklı biçimde** ele alınır
  - Kapalı projeksiyon
  - Açık projeksiyon

Arayüz Tabanlı Kapalı Projeksiyon

- Entity sınıfın sadece sorgu sonucu **ihtiyaç duyulan property'lerini dönecek getter metot karşılıklarını** içeren bir **arayüz** tanımlanır
- Sorgu metodunun **return değeri bu arayüz tipinde** tanımlanır

```
public interface VetRepository extends JpaRepository<Vet, Long> {
    List<VetView> findByGraduationYearBetween(int from, int to);
}
```

```
public interface VetView {
    String getFirstName();
    String getLastName();
}
```

Arayüz Tabanlı Kapalı Projeksiyon

- Arka tarafta bu view arayüzünden **proxy nesneler** oluşturulur ve getter metot çağrıları **entity nesnelere delege** edilir
- **Nested projeksiyon arayüzleri** de desteklenmektedir

```
public interface OwnerView {
    String getFirstName();
    String getLastName();
    Set<PetView> getPets();
}
```

getPets() metot ismi Owner entity sınıftaki getPets() metodu ile eşleşmelidir

```
public interface PetView {
String getName();
}
```

Arayüz Tabanlı Açık Projeksiyon

- Bu yöntemde ise getter metotlarının entity sınıflarında herhangi bir **karşılığı olmasına gerek yok**tur

```
public interface VetView {
    @Value("#{target.firstName + ' ' + target.lastName}")
    String getFullName();
}
```

Buradaki SpEL ifadesinde target entity nesneye refer eder

- Sorgu sonucu **hangi property'lerin kullanıldığı bilinmediği için** sorguda JPA spesifik **optimizasyon imkanı yok**tur

Sınıf Tabanlı Projeksiyon

- Arayüz yerine ilgili property'leri içeren **projeksiyon sınıfları** tanımlanır

```
public class VetDTO {
    private String firstName;
    private String lastName;

    public VetDTO(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public String getFirstName() {
        return firstName;
    }
    public String getLastName() {
        return lastName;
    }

    //equals & hashCode metotları...
}
```

Constructor parametre isimleri entity sınıfın property'leri ile eşleşmelidir

Dönen sonuçların collection vb içerisinde işlenebilmesi için equals ve hashCode metotlarına da ihtiyaç vardır

Sınıf Tabanlı Projeksiyon

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

    List<VetDTO> findByGraduationYearBetween(int from, int to);

}
```

- Sınıf tabanlı projeksiyon yönteminde **nested projeksiyon desteklenmez**

Dinamik Projeksiyon

- **Projeksiyon tiplerinin** aynı finder metotlar için senaryoya göre **farklılaşması** gerekebilir
- Bunun için **projeksiyon tipi finder metotda input argüman** olarak verilmelidir

```
public interface VetRepository extends JpaRepository<Vet, Long> {
    <T> List<T> findByGraduationYearBetween(int from, int to, Class<T> type);
}
```

İletişim

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