

Spring Data Projesi ve Fetch Sorguları



Fetch Sorguları

- Sorgu sonucu dönen entity nesnelerin **lazy ilişkilerinin eager** yüklendiği sorgulardır
- **Manuel sorgu** tanımlayarak sorgu içerisinde “**collection fetch**” işlemi yapılabilir
- JPA'nın **Named Entity Graph** kabiliyetinden yararlanılabilir

Manuel Sorgu ile Fetch

- HQL sorgusu içerisinde “**fetch**” ifadesi kullanılarak lazy ilişki eager fetch edilir

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

    @Query("select distinct o from Owner o left join fetch o.pets p
           where o.firstName = :firstName")
    List<Owner> findOwnersByFirstName(
        @Param("firstName")String firstName);

    ...
}
```

Named Entity Graph ile Eager Fetch

- Öncelikle global metadata olarak named entity graph tanımı yapılır

```
@NamedEntityGraph(name="Owner.pets",  
                  attributeNodes=@NamedAttributeNode("pets"))  
  
@Entity  
public class Owner {  
    @Id  
    private Long id;  
  
    @OneToMany  
    @JoinColumn(name="owner_id")  
    private Set<Pet> pets = new HashSet<>();  
  
    //...  
}
```

Named Entity Graph ile Eager Fetch

- Ardından da finder metot üzerinde **@EntityGraph** anotasyonu ile named entity graph devreye alınır

```
public interface OwnerRepository extends JpaRepository<Owner, Long> {  
    @EntityGraph("Owner.pets")  
    List<Owner> findByFirstName(String firstName);  
}
```

Ad-Hoc Entity Graph ile Eager Fetch

- **@NamedEntityGraph** tanımı yapmadan **finder metot üzerinde ad-hoc @EntityGraph tanımı** ile de eager fetch yapılabilir

```
public interface OwnerRepository extends JpaRepository<Owner, Long> {  
  
    @EntityGraph(attributePaths="pets")  
    List<Owner> findByFirstName(String firstName);  
  
}
```

İletişim

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

