

Spring Data Projesi ve Repository Bean Tanımlama



Spring Data Nedir?

- **Farklı veri erişim teknolojileri ile Spring** programlama modeli üzerinden çalışmayı kolaylaştıran bir projedir
- **Repository** bean'lerini **dinamik olarak oluşturmayı** sağlar
- Repository arayüzündeki metot isimlerinden **dinamik sorgular** üretir
- **Repository** bean'lerini ve **sorguları özelleştirme imkanı** da sunar

Spring Data Nedir?

- Arka taraftaki **veri erişim teknolojisine has özellikleri** de kaybetmeden çalışmayı hedefler
- **İlişkisel ve ilişkisel olmayan veri tabanları**, map-reduce frameworkleri, bulut tabanlı veri servislerini destekler

Repository Arayüzü

- Repository sınıflarının implement ettiği **marker arayüzdür**
- Temel amacı repository tarafından yönetilecek **domain sınıfının ve PK değerinin tipini** belirlemektir

```
public interface Repository<T, ID extends Serializable> {  
  
}
```

CrudRepository Arayüzü

- Yönetilen domain sınıfı üzerinde **CRUD işlemleri** yapmayı sağlayacak bir arayüz tanımlar

```
public interface CrudRepository<T, ID extends Serializable>
    extends Repository<T, ID> {
    <S extends T> S save(S entity);
    <S extends T> Iterable<S> save(Iterable<S> entities);
    T findOne(ID id);
    boolean existsById(ID id);
    Iterable<T> findAll();
    Iterable<T> findAllById(Iterable<ID> ids);
    long count();
    void deleteById(ID id);
    void delete(T entity);
    void deleteAll(Iterable<? extends T> entities);
    void deleteAll();
}
```

PagingAndSorting Repository Arayüzü

- CrudRepository arayüzünden türeyen ve entity'lere erişimi **paging** ile kolaylaştırmayı hedefleyen bir arayüzdür

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

```
    extends CrudRepository<T, ID> {
```

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

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

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

```
    }
```

↓
Sorgu sonucu belirtilen sayfadaki
entity'leri içerir

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

JpaRepository Arayüzü

- PagingAndSortingRepository arayüzünden türer ve **JPA'ya özel kabiliyetler** ilave eder

```
public interface JpaRepository<T, ID> extends
    PagingAndSortingRepository<T, ID> {

    void flush();
    <S extends T> S saveAndFlush(S entity);
    void deleteInBatch(Iterable<T> entities);
    void deleteAllInBatch();
    T getOne(ID id);
}
```

Veri Erişim Teknolojilerine Özel Diğer Repository'ler

- **JpaRepository** dışında **MongoRepository**, **CouchbaseRepository** gibi başka alt arayüzler de mevcuttur
- Bu arayüzler bahsedilen **temel repository kabiliyetlerini** sunarlar
- Ayrıca **veri erişim teknolojisine has kabiliyetler** de içerirler

Uygulamaya Ait Repository Arayüzlerinin Oluşturulması

- JpaRepository veya veri erişim teknolojisine özel arayüzden türeyen **uygulamaya özel arayüzler** tanımlanır
- Bu repository arayüzlerine daha sonra senaryolara özgü **DSL sorgu metotları** eklenebilir

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

    ...

}
```

Repository Nesnelerinin Yaratılması

- Spring Data kabiliyetleri **XML** veya **Java tabanlı** Spring konfigürasyonları ile devreye alınabilir
- Uygulama içerisinde belirli **paketlerin altı taranarak** Repository arayüzlerinden türeyen bean instance'ları otomatik olarak yaratılmaktadır

Repository Nesnelerinin Yaratılması

Java tabanlı konfigürasyon

```
@Configuration
@EnableJpaRepositories(
    basePackages={"com.javaegitimleri.petclinic.repository"})
public class ApplicationConfiguration {
    ...
}
```

XML tabanlı konfigürasyon

```
<beans ...>

    <jpa:repositories
        base-package="com.javaegitimleri.petclinic.repository">
    </jpa:repositories>

</beans>
```

Spring Boot ve Spring Data

- pom.xml içerisine data jpa starter'ı eklenerek **Spring Data projesinin JPA modülünün** özellikleri devreye alınabilir

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

- Repository arayüzlerini tespit etmek için **basePackage** olarak **@SpringBootApplication** anotasyonuna sahip **sınıfın bulunduğu paketi** tanımlar

Repository Nesnelerine Erişim

- Proxy repository nesneleri diğer normal Spring bean'leri gibi **enjekte** edilebilirler
- Ya da ApplicationContext'den **lookup** yapılabilirler

```
@Service
@Transactional
public class PetClinicServiceImpl implements PetClinicService {

    private VetRepository vetRepository;

    @Autowired
    public void setVetRepository(VetRepository vetRepository) {
        this.vetRepository = vetRepository;
    }
    ...
}
```

İletişim

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

