

# Spring Data Repository Bean'lerine Custom Metot Eklenmesi



# Repository Arayüzüne Custom Metot Eklenmesi

- Zaman zaman Repository bean'lerine **custom metot implemantasyonları** eklenmesi gerekmektedir
- Custom metotları içeren Repository arayüzünden ayrı **başka bir arayüz** daha tanımlanır
- Custom arayüzü implement eden **bir sınıf oluşturulur**
- Bu sınıf içerisinde de **custom metotlar implement edilir**

# Repository Arayüzüne Custom Metot Eklenmesi

```
public interface VetRepositoryCustom {  
    List<Vet> findVetsWithSpecialties();  
}
```

Belirtilen suffix'e sahip olması önemlidir!

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

Dinamik repository bean'nin yaratılacağı arayüz de tanımlanan bu custom arayüzü extend etmelidir

# Repository Arayüzüne Custom Metot Eklenmesi

Belirtilen suffix'e sahip olması önemlidir!

```
public class VetRepositoryImpl implements VetRepositoryCustom {  
  
    private SessionFactory sessionFactory;  
  
    public void setSessionFactory(SessionFactory sessionFactory) {  
        this.sessionFactory = sessionFactory;  
    }  
  
    @Override  
    public List<Vet> findVetsWithSpecialties() {  
        return sessionFactory.getCurrentSession()  
            .createQuery("select distinct v from Vet v left join fetch  
v.specialties").list();  
    }  
}
```

# Repository Arayüzüne Custom Metot Eklenmesi

- Eğer custom repository implemantasyonuna herhangi bir **bağımlık enjekte edilmeyecek ise** bean tanımlamaya gerek yoktur
- Spring custom arayüzü implement eden “**Impl**” **uzantılı sınıfı** tespit eder ve bundan bir instance yaratır
- Bu sınıfın **Repository arayüzü ile aynı paket altında** olması gerekir

# Repository Arayüzüne Custom Metot Eklenmesi

- Eğer bağımlılık enjekte edilecek ise bu durumda Impl uzantılı isimde bir bean tanımlanmalıdır

```
<beans...>  
  <bean id="vetRepositoryImpl"  
    class="com.javaegitimleri.petclinic.repository.VetRepositoryImpl">  
    <property name="sessionFactory" ref="sessionFactory"/>  
  </bean>  
</beans>
```

# Custom Metodun Bütün Repository'lere Eklenmesi

- Bütün Repository bean'lerinin sahip olacağı **ortak metotlar yazmak da mümkündür**
- Bu durumda custom metotları içerecek arayüz **Spring Data'nın Repository arayüzünden extend etmelidir**
- Projenin **Repository arayüzleri** ise bu durumda **custom arayüzden extend edeceklerdir**

# Custom Metodun Bütün Repository'lere Eklenmesi

**@NoRepositoryBean**

Bu arayüzden dinamik repository bean'i oluşturulmasını engellemek için eklenir

```
public interface BaseRepository<T, ID extends Serializable>  
    extends JpaRepository<T, ID> {  
    Session getSession();  
}
```

```
public abstract class BaseRepositoryImpl<T, ID extends Serializable>  
    extends SimpleJpaRepository<T, ID>  
    implements BaseRepository<T, ID> {
```

```
    private EntityManager entityManager;
```

```
    public BaseRepositoryImpl(Class<T> domainClass, EntityManager em) {  
        super(domainClass, em);  
        entityManager = em;  
    }
```

```
@Override  
    public Session getSession() {  
        return (Session) entityManager.getDelegate();  
    }  
}
```



# Custom Metodun Bütün Repository'lere Eklenmesi

```
<beans ...>
```

```
<jpa:repositories
```

```
base-package="com.example.repository"
```

```
base-class="com.example.repository.BaseRepositoryImpl">
```

```
</jpa:repositories>
```

```
</beans>
```

```
@EnableJpaRepositories(repositoryBaseClass=BaseRepositoryImpl.class)
```

```
@Configuration
```

```
public class PetClinicConfiguration {
```

```
...
```

```
}
```

# İletişim

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

