

ProxyFactoryBean Kabiliyeti



ProxyFactoryBean



- Spring'in sunduğu scope, tx yönetimi, caching, validation, aop gibi kabiliyetlerin dışında herhangi bir özelliği hayata geçirmek için de proxy örüntüsü kullanılabilir
- Bu durumda "explicit" olarak proxy oluşturulması gerekecektir
- Spring bunun için ProxyFactoryBean sınıfını sunar

ProxyFactoryBean



- Spring FactoryBean implemantasyonudur
- Proxy nesnenin sınıfını interface veya sınıf tabanlı yöntemle dinamik olarak üretir
- Hedef bean'i sarmalayan bir proxy yaratır
- Proxy nesneye özel davranış
 org.aopalliance.intercept.MethodInterce
 ptor arayüzü ile implement edilir





ProxyFactoryBean

```
<bean id="petClinicDao"</pre>
    class="com.javaegitimleri.petclinic.dao.PetClinicDaoInMemoryImpl"/>
<bean id="petClinicServiceTarget"</pre>
    class="com.javaegitimleri.petclinic.service.PetClinicServiceImpl">
    <constructor-arg ref="petClinicDao"/>
</bean>
<bean id="petClinicServiceProxy"</pre>
        class="org.springframework.aop.framework.ProxyFactoryBean">
    cproperty name="target" ref="petClinicServiceTarget"/>
    property name="interfaces">
        <array>
           <value>
                com.javaegitimleri.petclinic.service.PetClinicService
           </value>
        </array>
    cproperty name="interceptorNames">
        <value>cachingInterceptor,loggingInterceptor</value>
    </property>
</bean>
```

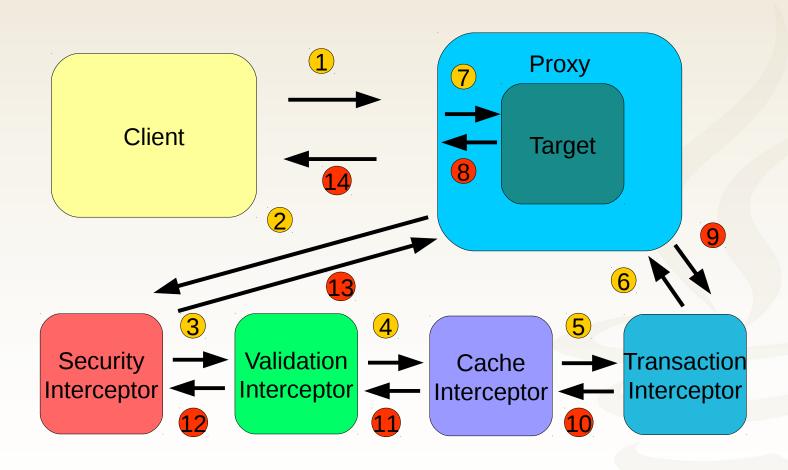


MethodInterceptor

```
<bean id="cachingInterceptor"</pre>
  class="com.javaegitimleri.petclinic.service.CachingInterceptor"/>
  <bean id="loggingInterceptor"</pre>
  class="com.javaegitimleri.petclinic.service.LoggingInterceptor"/>
public class LoggingInterceptor implements MethodInterceptor {
    @Override
    public Object invoke(MethodInvocation invocation) throws Throwable {
        try {
            System.out.println("method entered");
            Object result = invocation.proceed();
            System.out.println("method executed successfully, returning
result :" + result);
            return result:
        } catch (Throwable t) {
            System.out.println("exception occurred :" + t);
            throw t;
        } finally {
            System.out.println("method exited");
```



MethodInterceptor Zinciri





İletişim

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

