

Hibernate Interceptor



Hibernate 3 Öncesi Dönem ve Interceptor Arayüzü

- Hibernate 3 öncesi dönemde Session üzerinde meydana gelen işlemleri intercept etmek için kullanılan yöntemdir
- **Interceptor** arayüzü implement edilerek Hibernate event'lerinde işlem yapılabilir
- **EmptyInterceptor** default implementasyondur
- Interceptor'ü aktive etmek için **Configuration.setInterceptor()** ile interceptor nesnesi set edilmelidir
- Bu işlem SessionFactory **build edilmeden** yapılmalıdır

Hibernate Interceptor API

```
public class AuditLogInterceptor extends EmptyInterceptor {

    public boolean onLoad(Object entity, Serializable id,
        Object[] state, String[] propertyNames, Type[] types) {
        return false;
    }

    public boolean onSave(Object entity, Serializable id,
        Object[] state, String[] propertyNames, Type[] types) {
        return false;
    }

    public boolean onFlushDirty(Object entity, Serializable id,
        Object[] currentState, Object[] previousState,
        String[] propertyNames, Type[] types) {
        return false;
    }

    public void onDelete(Object entity, Serializable id,
        Object[] state, String[] propertyNames, Type[] types) {
    }

    public void afterTransactionCompletion(Transaction tx) {
    }
}
```

State
üzerinde
değişiklik
yapılırsa
true
dönülür

Hibernate Interceptor Konfigürasyonu

- Interceptor konfigürasyonu **Configuration** nesnesi üzerinden programatik yapılır
- Configuration düzeyinde tanım **SessionFactory genelinde aktif** olacaktır
- Dolayısı ile bu Interceptor **bütün Session'larda devreye girecektir**

```
Configuration cfg = new Configuration();  
  
cfg.setInterceptor(new AuditLogInterceptor());  
  
cfg.configure();  
SessionFactory sessionFactory = cfg.buildSessionFactory();
```

Hibernate Interceptor Konfigürasyonu ve JPA

```
<persistence-unit name="jpa-hibernate">  
  <properties>  
    <property name="hibernate.ejb.interceptor"  
value="com.javaegitimleri.AuditLogInterceptor"/>  
    ...  
  </properties>  
</persistence-unit>
```

↓
Hibernate Interceptor **JPA** konfigürasyonunda
EntityManagerFactory/SessionFactory düzeyinde
tanıtılabilir

Hibernate Interceptor Konfigürasyonu ve JPA

```
<persistence-unit name="jpa-hibernate">  
  <properties>  
    <property  
name="hibernate.ejb.interceptor.session_scoped"  
value="com.javaegitimleri.AuditLogInterceptor"/>  
    ...  
  </properties>  
</persistence-unit>
```

Hibernate Interceptor **JPA** konfigürasyonunda
EntityManager/Session düzeyinde de tanımlanabilir
Bu durumda her EntityManager/Session oluşturulduğunda
bu PersistenceContext'e özgü yeni bir Interceptor
instance'ı yaratılacaktır

Hibernate Interceptor Kullanımında Dikkat Edilecek Noktalar

- Interceptor içerisinde bir nesnenin **save()** işlemi sırasında aynı **Session** üzerinden yeni bir **save()** metot çağrısı önerilmez
- Böyle bir durumda ikinci save işlemi **farklı bir Session** üzerinden gerçekleştirilmelidir

İletişim

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

