

Transaction Sonrasında İşlem Yapmak



Transaction Sonrasında İşlem Yapmak

- Transaction sonrası için ise **TransactionSynchronization** arayüzü implement edilmelidir
- Ardından bu sınıftan bir nesne aktif bir TX içerisinde **TransactionSynchronizationManager** ile register edilmelidir
- Eğer aktif TX olmadan bu işlem yapılırsa **IllegalStateException** fırlatılır

Transaction Sonrasında İşlem Yapmak

- **Birden fazla** tx synchronization nesnesi kayıt ettirilebilir
- Aralarındaki çalışma sırası **Ordered arayüzü** implement edilerek belirtilebilir
- **TransactionSynchronizationAdapter** isimli bir sınıf ile tx synchronization sınıfı sadece ilgili metotlar override edilerek yazılabilir

Transaction Sonrasında İşlem Yapmak

```
public class MyTransactionSynchronization extends
    TransactionSynchronizationAdapter {

    @Override
    public void beforeCommit(boolean readOnly) {
        System.out.println("tx is about to be committed");
    }

    @Override
    public void afterCommit() {
        System.out.println("tx committed");
    }

    @Override
    public void afterCompletion(int status) {
        System.out.println("tx result :" + status);
    }
}
```

Transaction Sonrasında İşlem Yapmak

```
public interface TransactionSynchronization extends Flushable {  
  
    int STATUS_COMMITTED = 0;  
    int STATUS_ROLLED_BACK = 1;  
    int STATUS_UNKNOWN = 2;  
  
    void suspend();  
    void resume();  
    void flush();  
  
    void beforeCommit(boolean readOnly);  
    void beforeCompletion();  
    void afterCommit();  
    void afterCompletion(int status);  
  
}
```

Transactional resource'u, örneğin Hibernate Session, TransactionSynchronizationManager'a bind/unbind etmek ve resource içindeki işlemleri DB'ye yansıtmak için kullanılırlar

Transaction Sonrasında İşlem Yapmak

```
@Service
public class TransactionalService {

    @Transactional(propagation=Propagation.REQUIRED)
    public void transactionalMethod() {

        TransactionSynchronizationManager
            .registerSynchronization(
                new MyTransactionSynchronization());

        //perform transaction logic...
    }
}
```

Register işlemi mutlaka aktif bir transaction içerisinde gerçekleşmelidir
PlatformTransactionManager konfigürasyonunda synchronization
Kabiliyetinin ne zaman devrede olacağı kontrol edilebilir

Transaction Sonrasında İşlem Yapmak

```
<bean id="transactionManager"  
class="org.springframework.orm.hibernate4.HibernateTransactionManager">  
  
    <property name="sessionFactory" ref="sessionFactory" />  
  
    <property name="transactionSynchronization" value="0"/>  
  
    <property name="transactionSynchronizationName"  
              value="SYNCHRONIZATION_ALWAYS" />  
</bean>
```

SYNCHRONIZATION_ALWAYS=0
SYNCHRONIZATION_ON_ACTUAL_TRANSACTION=1
SYNCHRONIZATION_NEVER=2

ALWAYS (default) değerinde transaction propagation kuralı ne olursa olsun synchronization yapılabilir. Eğer değer ACTUAL_TRANSACTION ise sadece REQUIRED, REQUIRES_NEW ve MANDATORY'de synchronization kod register edilebilir.

İletişim

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

