

JPA ve Transaction Yönetimi



JPA ve Transaction Yönetimi

- JPA da TX yönetimi **iki farklı modda** gerçekleştirilir
 - Java SE modu (non-managed)
 - Java EE modu (managed)
- **Java SE** modunda **sadece resource-local** transaction kullanılabilir
- **EntityManager** ve **Transaction** oluşturma/yönetme görevi de **uygulamanın kendisinin**dir

JPA ve Transaction Yönetimi

- **Java EE** modunda ise genellikle **JTA transaction** kullanılır, ancak **istenirse resource-local TX** ile de çalışılabilir
- JavaEE modunda **EntityManager** ve **Transaction** genellikle **container tarafından** oluşturulur ve yönetilir

JPA ve Transaction Yönetimi

- **Transaction** demarcation işlemi de **uygulamanın veya container'ın yönetmesine** göre ikiye ayrılır
 - BMT (bean managed/programatik)
 - CMT (container managed)
- **Java SE** modunda **sadece BMT** yapılabilir
- TX uygulama içerisinde programatik biçimde başlatılır ve sonlandırılır
- **Java EE** modunda ise **hem CMT, hem de BMT** yapılabilir

Java SE Modunda Resource Local Transaction Yönetimi

Her createEntityManager()
metot çağırısı yeni bir
EntityManager yaratır

```
EntityManager entityManager = null;  
EntityTransaction transaction = null;  
try {  
    entityManager = entityManagerFactory  
        .createEntityManager();  
    transaction = entityManager.getTransaction();  
    transaction.begin();  
  
    Pet kedi = new Pet();  
    kedi.setName("boncuk");  
  
    entityManager.persist(kedi);  
  
    transaction.commit();  
} catch (Exception ex) {  
    transaction.rollback();  
}  
finally {  
    entityManager.close();  
}
```

Persistence işlemi
mutlaka açık bir TX
bloğu içerisinde
gerçekleştirilmelidir



aksi takdirde JPA
hata fırlatır

EntityManager'ı kapatmak uygulamanın
sorumluluğundadır

Java SE Modunda Resource Local Transaction Yönetimi


```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" ...>
  <persistence-unit name="testjpa"
    transaction-type="RESOURCE_LOCAL">

    <provider>
      org.hibernate.jpa.HibernatePersistenceProvider
    </provider>

    <non-jta-data-source>jdbc/myDS</non-jta-data-
source>

    <properties>
      <property name="hibernate.ejb.cfgfile"
        value="/hibernate.cfg.xml" />
    </properties>

  </persistence-unit>
</persistence>
```



Lokal TX yönteminde tek bir DB söz konusudur, transaction-type RESOUCE_LOCAL şeklinde belirtilirse DataSource ise non-jta-data-source ile belirtilmelidir

@PersistenceUnit & @PersistenceContext

- Her iki anotasyon da sadece **JavaEE modunda** kullanılabilir
- **@PersistenceUnit** anotasyonu **EntityManagerFactory** nesnesini enjekte etmek için kullanılır
- Container tarafından yönetilen **transactional EntityManager**'a erişmek için ise **@PersistenceContext** anotasyonu kullanılır

JPA Exception'ları

- PersistenceContext üzerinde gerçekleştirilen işlemler sırasında hata meydana geldiğinde **PersistenceException** ve türevleri fırlatılır
- NoResultException, NonUniqueResultException, LockTimeoutException, QueryTimeoutException dışındaki **bütün exception'lar TX rollback nedenidir**
- TX rollback yapılması allocate edilmiş **DB resource'larının beklemeden saliverilmesi** için önemlidir

İletişim

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

