

# Spring Entegrasyon Testleri ve Transaction Yönetimi



# Entegrasyon Testleri ve Transaction Yönetimi

- **@Transactional** anotasyonu test sınıfı veya metot düzeyinde tanımlanabilir
- Container'da tanımlı bir **PlatformTransactionManager** bean mevcut olmalıdır
- Test metodu başladığında bir **TX başlatılır**
- Test metodu sonlandığında da bu TX **rollback edilir**
- **@Rollback(false)** ile TX **commit** ettirilebilir

# Entegrasyon Testleri ve Transaction Yönetimi

```
@ContextConfiguration
@Transactional
public class TransactionalTests {
    @Test
    public void testWithRollback() {
        // ...
    }

    @Rollback(false)
    @Test
    public void testWithoutRollback() {
        // ...
    }
}
```

# Entegrasyon Testleri ve Transaction Yönetimi

- **@TransactionConfiguration** ile sınıf düzeyinde TX konfigürasyonu özelleştirilebilir
- **@Before** ve **@After** metotları TX içinde çalıştırılır
- **@BeforeTransaction**, **@AfterTransaction** ile de TX dışında setup ve destroy işlemleri yapılabilir

# Entegrasyon Testleri ve Transaction Yönetimi

@ContextConfiguration

@Transactional

@TransactionConfiguration(transactionManager="txMgr",  
defaultRollback=false)

public class CustomConfiguredTransactionalTests {

@Before

public void setUp() {  
}

} @Before anotasyonu ile işaretlenmiş setUp metodu TX içerisinde çalıştırılır. TX dışında çalışması için @BeforeTransaction kullanılmalıdır

@Rollback(true)

@Test

public void testProcessWithoutRollback() {  
}

@After

public void tearDown() {  
}

} @After anotasyonu ile işaretlenmiş tearDown metodu da TX içerisinde çalıştırılır. TX dışında çalışması için @AfterTransaction kullanılmalıdır

# İletişim

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

