


Spring Transaction Management

Test your understanding about Spring transaction management. Some questions are based on code samples and some are based on on Spring documentation

Category	Technical Skill / Interview questions
Keywords	spring , spring certification test, spring mock test, spring transaction
Tags	java spring
Questions	10 - Free questions
Test takers	1091
Average score	52.3
Rating	 (4/5)
Guru	j2eeGuru
View questions of this test	
<div><div>Start the Test</div><div>Go Back</div></div>	

Test : *Spring Transaction Management*



9:57

Question :1

What kind of servers can support Spring Transaction ?

Select Your Answer :

- ☐ Only Application servers
- ☐ Only Application servers which support J2EE specification
- ☐ Spring's transactional services are supported on web servers also

Question :2

In service class UserAdminService, there is a method updateUser().

```
public interface UserAdminService {  
    public void updateUser() throws UserInvalidException ;  
}
```

UserInvalidException is a checked exception.

This is how it is defined in applicationContext-Service.xml

```
<bean id="userAdminService, " parent="txProxyTemplate">  
    <property name="target">  
        <bean  
            class="com.abc.impl.UserAdminServiceImpl"  
            autowire="byName" />  
        </property>  
    <property name="transactionAttributes">  
        <props>  
            <prop  
key="internalLoadTreatment">PROPAGATION_NESTED</prop>  
            <prop key="*">PROPAGATION_REQUIRED</prop>  
        </props>  
    </property>  
</bean>
```

If UserInvalidException is thrown by method updateUser, will the transaction be rolled back ?

Select Your Answer :

-
- ☐ Yes, always.
-
- ☐ No since UserInvalidException is checked exception.
-
- ☐ Yes if UserInvalidException is made a subclass of runtime exception.
-
- ☐ Data is incomplete
-

Question :3

In service class UserAdminService, there is a method updateUser().

```
public interface UserAdminService {  
    public void updateUser() throws UserInvalidException ;  
}
```

UserInvalidException is a checked exception.

This is how it is defined in applicationContextService.xml

```
<bean id="userAdminService, " parent="txProxyTemplate">  
    <property name="target">  
        <bean  
            class="com.abc.impl.UserAdminServiceImpl"  
            autowire="byName" />  
        </property>  
    <property name="transactionAttributes">  
        <props>  
            <prop  
key="internalLoadTreatment">PROPAGATION_NESTED</prop>  
            <prop key="*">PROPAGATION_REQUIRED</prop>  
        </props>  
    </property>  
</bean>
```

If UserInvalidException is thrown by method updateUser, will the transaction be rolled back ?

Select Your Answer :

- ☐ If prop key in applicationContextService.xml is changed to
<prop key="*">PROPAGATION_REQUIRED,-UserInvalidException </prop>
- ☐ If prop key in applicationContextService.xml is changed to
<prop key="*">PROPAGATION_REQUIRED,+UserInvalidException </prop>
- ☐ Only if UserInvalidException is made a subclass of RuntimeException , transaction will be rolled back

Question :4

Spring's transaction behavior can be applied to any class not just service or JDBC classes.
True or False

Select Your Answer :

- ☐ True
- ☐ False

Question :5

In Spring Framework's declarative transaction implementation , is it sufficient to annotate the classes with the @Transactional annotation ?

Select Your Answer :

- ☐ No
- ☐ Yes
- ☐ No . <tx:annotation-driven/> has to be added in configuration file

Question :6

The key to the Spring transaction abstraction is the notion of a transaction strategy. A transaction strategy is defined by the `org.springframework.transaction.PlatformTransactionManager` interface. The `TransactionException` that can be thrown by any of the `PlatformTransactionManager` interface's is what kind of exception ?

Select Your Answer :

- ☐ Checked Exception
- ☐ Unchecked exception

Question :7

One of the unique features of spring transaction management which is not provided by the EJB is

Select Your Answer :

- ☐ Annotations
- ☐ JPA Support
- ☐ Declarative rollback rules

Question :8

If for a method updateUser , we do not want the rollback for UserNotFoundException , what changes are to be done in the file configuration below ?

```
<tx:advice id="txAdvice">
  <tx:attributes>
    <tx:method name="updateUser"/>
    <tx:method name="*/>
  </tx:attributes>
</tx:advice>
```

Select Your Answer :

- ☐ replace <tx:method name="updateUser"/> with
<tx:method name="updateUser" no-rollback-for="UserNotFoundException"/>
- ☐ replace <tx:method name="updateUser"/> with
<tx:method
name="updateUser,-UserNotFoundException/>
- ☐ replace <tx:method name="updateUser"/> with
<tx:method name="updateUser,+UserNotFoundException/>

Question :9

Spring recommends that you only annotate concrete classes (and methods of concrete classes) with the @Transactional annotation, as opposed to annotating interfaces.
True or False.

Select Your Answer :

- ☐ True
- ☐ False

Question :10

In code example below , which transactional advice will take precedence ?

```
@Transactional(readOnly = true)
public class DefaultFooService implements FooService {

    public Foo getFoo(String fooName) {
        // do something
    }

    @Transactional(readOnly = false, propagation = Propagation.REQUIRES_NEW)
    public void updateFoo(Foo foo) {
        // do something
    }
}
```

Select Your Answer :

-
- ☐ Transaction settings in class takes precedence over the transactional settings defined in the method
-
- ☐ Transaction settings in method in the same class takes precedence over the transactional settings defined at the class level
-
- ☐ Spring considers it mismatch and throws exception
-