


### Spring Transaction Management

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

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

## Test : Spring Transaction Management

Question :1 What kind of servers can support Spring Transaction ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	Only Application servers	
	<input type="radio"/>	Only Application servers which support J2EE specification	
✓	<input type="radio"/>	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?

Correct Answer	Your Selection	Answer	Explanation
	<input type="checkbox"/>	Yes, always.	
✓	<input type="checkbox"/>	No since UserInvalidException is checked exception.	
✓	<input type="checkbox"/>	Yes if UserInvalidException is made a subclass of runtime exception.	
	<input type="checkbox"/>	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

Correct Answer	Your Selection	Answer	Explanation
✓	<input checked="" type="radio"/>	If prop key in applicationContextService.xml is changed to <prop key="">PROPAGATION_REQUIRED,- UserInvalidException </prop>	- implies the transaction is to be rolled back for this exception
	<input type="radio"/>	If prop key in applicationContextService.xml is changed to <prop key="">PROPAGATION_REQUIRED,+UserInvalidException </prop>	
	<input type="radio"/>	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

Correct Answer	Your Selection	Answer	Explanation
✓	<input checked="" type="radio"/>	True	You can apply the Spring Framework declarative transaction management to any class, not merely special classes such as EJBs
	<input type="radio"/>	False	

Question :5 In Spring Framework's declarative transaction implementation , is it sufficient to annotate t classes with the @Transactional annotation ?

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="checkbox"/>	No	
	<input type="checkbox"/>	Yes	
✓	<input type="checkbox"/>	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 ?

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	Checked Exception	
✓	<input checked="" type="radio"/>	Unchecked exception	in keeping with Spring's philosophy, the TransactionException that can be thrown by any of the PlatformTransactionManager interface's methods is unchecked (that is, it extends the java.lang.RuntimeException class). Transaction infrastructure failures are almost invariably fatal. In rare cases where application code can actually recover from a transaction failure, the application developer can still choose to catch and handle TransactionException. The salient point is that developers are not forced to do so

Question :7 One of the unique features of spring transaction management which is not provided by the J is

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	Annotations	
	<input type="radio"/>	JPA Support	
✓	<input type="radio"/>	Declarative rollback rules	The Spring Framework offers declarative rollback rules, a feature with no EJB equivalent. Both programmatic and declarative support for rollback rules is provided.

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>
```

Correct Answer	Your Selection	Answer	Explanation
✓	<input type="radio"/>	replace <tx:method name="updateUser"/> with <tx:method name="updateUser" no-rollback-for="UserNotFoundException"/>	
	<input type="radio"/>	replace <tx:method name="updateUser"/> with <tx:method name="updateUser,- UserNotFoundException"/>	
	<input type="radio"/>	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 class) with the @Transactional annotation, as opposed to annotating interfaces.  
True or False.

Correct Answer	Your Selection	Answer	Explanation
✓	<input checked="" type="radio"/>	True	
	<input type="radio"/>	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
    }
}
```

Correct Answer	Your Selection	Answer	Explanation
	<input type="radio"/>	Transaction settings in class takes precedence over the transactional settings defined in the method	
✓	<input checked="" type="radio"/>	Transaction settings in method in the same class takes precedence over the transactional settings defined at the class level	
	<input type="radio"/>	Spring considers it mismatch and throws exception	