

Spring框架的事务管理应用分析[1] PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/167/2021\\_2022\\_Spring\\_E6\\_A1\\_86\\_E6\\_c104\\_167011.htm](https://www.100test.com/kao_ti2020/167/2021_2022_Spring_E6_A1_86_E6_c104_167011.htm) 引言在软件开发中出现过各种各样的框架，开源软件的兴起，使得各种各样的框架纷纷出现，例如，Apache组织下就拥有诸多的框架类产品。框架就是一组协同工作的类，它们为特定类型的软件构筑了一个可重用的设计。然而，传统的框架使得应用程序组件过分依赖于框架中的类，这种耦合度的提高降低了组件的复用性。Spring框架的出现，使得组件之间更松散的耦合成为了可能。Spring框架简介Spring框架是一个2003年2月才出现的开源项目，该开源项目起源自Rod Johnson在2002年末出版的《Expert One-on-One J2EE Design and Development》一书中的基础性代码。在该书中，Rod Johnson倡导J2EE实用主义的设计思想，而Spring框架正是这一思想的更全面和具体的实现。Spring框架由一个容器，一个配置和组织组件的框架，和一组内置的为事务、持久化和Web用户接口提供的服务组成。作为一种轻量级的J2EE框架，Spring提供了一种有效的方式来建立和组织J2EE应用程序。

- 1、Spring特性IoC(Inversion of Control；控制反转)；又称DI(Dependency Injection；依赖注入)；是面向对象领域新兴的编程思想；也是Spring的精髓所在。简单地说；IoC就是指程序之间的关系由容器来控制；而不是传统实现中由程序代码直接操控。这也就是所谓“控制反转”的概念所在：控制权由应用代码转到外部容器，控制权的转移，也就是所谓的反转。IoC将控制创建的职责搬进了框架中；并把它从应用代码脱离开来。当使用Spring的IoC容器时只需指出

组件需要的对象，在运行时Spring的IoC容器会根据XML配置数据提供给它。Spring IoC，借助于依赖注入设计模式，使得开发者不用理会对象自身的生命周期及其关系，而且能够改善开发者对模式的使用。对于一个对象的管理不是什么困难，难就难在对整个对象群的管理。依赖注入可以让容器管理对象，即“Don't call me, I will call you”。这样对象本身的生命周期以及对象之间的关系就不再让开发者费神了。Spring AOP，借助于Spring实现拦截器，开发者能够实现以声名方式使用企业级服务，比如安全性服务、事务服务。AOP合理的补充了OOP，借助于Spring AOP，开发者能够高效的使用J2EE服务。Spring服务抽象，借助于各种J2EE API抽象，使得开发者能够一致地使用J2EE技术，而不管具体是使用什么J2EE API，借助于Spring服务抽象，使代码大大减少，满足“更少代码，更少BUG”的软件设计原则。Spring IoC Spring AOP Spring服务抽象，一起形成Spring，这样一个有机体，使构建轻量级J2EE成为可能。Spring提供的事务管理Spring提供的事务管理可以分为两类：编程式的和声明式的。编程式的，比较灵活，但是代码量大，存在重复的代码比较多；声明式的比编程式的更灵活方便。

1、传统使用JDBC的事务管理

以往使用JDBC进行数据操作，使用DataSource，从数据源中得到Connection，我们知道数据源是线程安全的，而连接不是线程安全的，所以对每个请求都是从数据源中重新取出一个连接。一般的数据源由容器进行管理，包括连接池。例如TOMCAT，WEBSPPHERE，WEBLOGIC等这些J2EE商业容器都提供了这个功能。以往的我们使用JDBC在写代码时，事务管理可能会是这样：

```
Connection conn = null.try{ conn =
```

DBConnectionFactory.getConnection.

```
conn.setAutoCommit(false). //do something conn.commit().
```

```
//commit transcation} catch(Exception e){ conn.rollback().} finally{  
try{conn.close(). } catch(SQLException se){ //do sth.} //close
```

ResultSet,PreparedStatement,Connection //notice:Maybe occur

Exception when u close rs,pstmt,conn} 按照以往的思路来写代码

，代码量比较长，而且容易疏忽，忘掉一些try/catch，引发一些

异常无法catch，虽然有时候我们会写DBTool类，来关闭这些

资源，并且保证在关闭这些资源时，不向外抛异常，但是

这样做会导致额外的麻烦。2、Spring提供的编程式的事务处理

Spring提供了几个关于事务处理的类：TransactionDefinition

//事务属性定义 TransactionStatus //代表了当前的事务，可以提

交，回滚。 100Test 下载频道开通，各类考试题目直接下载。

详细请访问 [www.100test.com](http://www.100test.com)