



注解方式的事务管理开发步骤

- 1.添加Spring开发必要的jar包
commons-logging-1.1.3.jar
spring-beans-4.0.0.RELEASE.jar
spring-context-4.0.0.RELEASE.jar
spring-core-4.0.0.RELEASE.jar
spring-expression-4.0.0.RELEASE.jar
- 2.在applicationContext.xml添加context命名空间
- 3.配置自动扫描包的注解：
<context:component-scan base-package = ""/>

- 1.在IOC容器（即applicationContext.xml文件）中配置事务管理器
<bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
 <!-- 加载数据源，因为事务管理的对象是数据源,数据源是c3p0-->
 <property name="dataSource" ref = "dataSource"/>
</bean>
- 2.在IOC容器中加载tx命名空间，启用事务注解
<tx:annotation-driven transaction-manger = "transactionManager" />
- 3.在需要添加事务的方法上添加事务注解
@Transactional

xml方式的事务管理开发步骤

- 1.配置Bean
<bean id="bookShopDao"
 class="com.spring.dao.impl.BookShopDaoImpl">
 <property name="jdbcTemplate"
ref="jdbcTempalte"/>
</bean>

<bean id="bookShopService"
 class="com.spring.service.impl.BookShopDaoImpl">
 <property name="bookShopDao"
ref="bookShopDao"/>
</bean>

- 1.配置事务管理器
<bean id="transactionManager"
 class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
 <!-- 加载数据源，因为事务管理的对象是数据源,数据源是c3p0-->
 <property name="dataSource" ref = "dataSource"/>
</bean>
- 2.配置事务属性，比如隔离级别，回滚等 ---类似切面
<tx:advice id="txAdvice" transaction-manager = "transactionManager">
 <tx:attribute>
 <tx:method name="get*" propagation = ""/>
 <tx:method name="*" />
 </tx:attribute>
</tx:advice>
- 3.配置事务切入点，以及把事务切入点和事务关联起来
<aop:config>
 <aop:pointcut id="txPointCut" expression="execution(* com.spring.tx.service.impl.*(..))">
 <aop:advisor advice-ref="txAdvice" pointcut-ref="txPointCut"/>
 </aop:config>