ORM框架

Mybatis

Hibernate

Spring JdbcTemplate

JTA

Spring中如何实现事务管理

1. 在spring配置文件根据spring整合的orm框架(mybatis,JTA,Hibernate),添加相应的事务管理 器(将事务管理 器纳入spring容器管理 )

如myBatis框架的事务管理器是 DataSourceTransactionManager

修改spring配置文件:,添加tx命名空间及事务管理 配置,看黄底部分标记

|  |
| --- |
| *<?*xml version="1.0" encoding="UTF-8"*?>* <beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:context="http://www.springframework.org/schema/context"  xmlns:aop="http://www.springframework.org/schema/aop"  xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"  xmlns:tx="http://www.springframework.org/schema/tx"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd  http://www.springframework.org/schema/aop  http://www.springframework.org/schema/aop/spring-aop.xsd  http://mybatis.org/schema/mybatis-spring  http://mybatis.org/schema/mybatis-spring.xsd  http://www.springframework.org/schema/tx  http://www.springframework.org/schema/tx/spring-tx.xsd">  *<!--开始spring的AOP功能-->* <aop:aspectj-autoproxy></aop:aspectj-autoproxy>     *<!--context:annotation-config代表开启注册配置,在类中可以使用@Component,@Service,@Respository,@Controller-->* <context:annotation-config></context:annotation-config>  *<!--表示扫描com.servlet包中的所有类，看为上是否有@Component,@Service,@Respository,@Controller其中之一的注解，有将该类纳入spring容器管理 -->* <context:component-scan base-package="com.service"></context:component-scan>  <context:component-scan base-package="com.mapper"></context:component-scan>  <context:component-scan base-package="com.aop"></context:component-scan>      <bean id="placeholderConfigurer" class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">  *<!--给location属性赋值 ，在此类的内部就会去读取classpath:properties配置文件 -->* <property name="location" value="classpath:jdbc.properties"></property>  </bean>   *<!--把commons dbcp2 纳入spring容器管理 XML DOM解析 生成XML   ${driverClassName}表示引用 jdbc.properties配置文件的driverClassName  -->   <!--整合MyBatis框架  org.mybatis.spring.SqlSessionFactoryBean是spring+mybatis整合包中提供的工具类，用于获得SqlSession对象的  -->* <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">  *<!--为SqlSessionFactoryBean注入属性-->* <property name="dataSource" ref="datasource"></property>  <property name="mapperLocations" value="classpath:com/mapper/\*.xml"></property>  <property name="configLocation" value="classpath:mybatis-config.xml"></property>  </bean>   *<!--将mybatis的com.mapper包也的所有的接口实现类纳入spring容器管理  在代码中可以用@Autowired注解注入依赖的Mapper  -->* <mybatis-spring:scan base-package="com.mapper"></mybatis-spring:scan>  *<!--配置mybtis的事务管理器  可以理解为一个aop的切面类  -->* <bean id="datasource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="closeConnectionPool">  <property name="driverClassName" value="${driver}"></property>  <property name="password" value="${password}"></property>  <property name="username" value="${username}"></property>  <property name="url" value="${url}"></property>  <property name="maxTotal" value="100"></property>  <property name="initialSize" value="10"></property>  </bean>   *<!—定义mybatis事务管理器 -->*  <bean id="tx" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">  *<!--注入 BasicDataSource-->* <property name="dataSource" ref="datasource"></property>  </bean>  *<!--启用@Transactional注解 然后引用 id=tx的事务管理 器 -->* <tx:annotation-driven transaction-manager="tx"></tx:annotation-driven> </beans> |
|  |

1. 在service包中的类需要实施事务管理的类上添加 @Transactional 注解启用事务

