**配置事务之前要有session**

**这里说几种session的配置**

**先创建一个数据源**

**数据源也有好几种配置 dbcp c3p0**

**<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">**

**<property name="driverClassName" value="com.mysql.jdbc.Driver"/>**

**<property name="url" value="jdbc:mysql://localhost:3306/colthes"/>**

**<property name="username" value="root" />**

**<property name="password" value="06110218" />**

**</bean>**

**再配置SessionFactory**

**这个根据使用的框架不同，也有几种配置**

**下面是hibernate的SessionFactroy**

**<bean id="sessionFactory" class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">**

**<property name="dataSource">**

**<ref bean="dataSource" />**

**</property>**

**<property name="hibernateProperties">**

**<props>**

**<prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect</prop>**

**<prop key="hibernate.hbm2ddl.auto">update</prop>**

**<prop key="javax.persistence.validation.mode">none</prop>**

**<prop key="show\_sql">true</prop>**

**<!-- <prop key="current\_session\_context\_class">thread</prop>**

**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

**<prop key="lazy">false</prop> -->**

**<!-- could not open hibernate session for transactio;nested exception is org.hibernate.transactionexception:jdbc begin failed:**

**Maybe 缓存同步的问题您可以这样获取Session Session session =HibernateSessionFactory.getSessionFactory().getCurrentSession();**

**试试 也可是修改Hibernate \*.hbm.xml中添加修改一个 lazy属性为false 默认为 true -->**

**<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->**

**</props>**

**</property>**

*<!-- 引入映射文件 -->  
   
 <!-- <property name="mappingLocations" value="classpath:com/blog/entry/\*.hbm.xml" ></property> -->* <**property name="mappingDirectoryLocations" value="classpath:com/blog/entry"** ></**property**>  
</**bean**>

**配置事务管理器**

<**bean id="transactionManager" class="org.springframework.orm.hibernate5.HibernateTransactionManager"**>  
 <**property name="sessionFactory" ref="sessionFactory"** />  
</**bean**>

**这个时候就必须在持久层注入session**

**package** com.blog.utils;  
  
**import** java.io.Serializable;  
  
**import** org.hibernate.Session;  
**import** org.hibernate.SessionFactory;  
**import** org.springframework.beans.factory.annotation.Autowired;  
**import** org.springframework.transaction.annotation.Transactional;  
  
@Transactional  
**public class** SessionUtil **implements** Serializable {  
 @Autowired  
 **private** SessionFactory sessionFactory;  
  
*// public void setSessionFactory(SessionFactory sessionFactory) {  
// this.sessionFactory = sessionFactory;  
// }* **public** Session getCurrentSession(){  
 **return** sessionFactory.getCurrentSession();  
 }  
}

**还有就是MyBatis的SqlSessionFactory**

*<!--配置sqlSessionFactory-->*<**bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"  
 p:dataSource-ref="dataSource" p:configLocation="classpath:conf/mybatiscfg.xml"**/>

**然后就是使用这些个SessionFactory来创建开启session**

*<!--扫描mapper文件 并注入sqlSessionFactory-->*<**bean class="org.mybatis.spring.mapper.MapperScannerConfigurer" p:basePackage="com.mvc.dao"**/>

*<!--配置事务-->*<**bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager" p:dataSource-ref="dataSource"**/>

*<!--配置事务的通知-->*<**tx:advice id="txAdvice" transaction-manager="transactionManager"**>  
 <**tx:attributes**>  
 <**tx:method name="save\*" read-only="false" isolation="DEFAULT" propagation="REQUIRED"**/>  
 <**tx:method name="update\*"**/>  
 <**tx:method name="delete\*"**/>  
 <**tx:method name="get\*" read-only="true"** />  
 <**tx:method name="find\*" read-only="true"** />  
 </**tx:attributes**>  
</**tx:advice**>  
  
*<!--配置切面-->*<**aop:config**>  
 <**aop:pointcut id="ponitcut" expression="execution(\* com.mvc.service.impl.\*Impl.\*(..))"** />  
 <**aop:advisor advice-ref="txAdvice" pointcut-ref="ponitcut"** />  
</**aop:config**>