配置dao

1.首先获取数据源：

1：直接写：

```
    <bean id="jdbcDataSource"
class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName">
            <value>com.mysql.jdbc.Driver</value>
        </property>
        <property name="url">
            <value>jdbc:mysql://localhost:3306/world?
characterEncoding=utf8&amp;useSSL=true</value>
        </property>
        <property name="username">
            <value>root</value>
        </property>
        <property name="password">
            <value>123456</value>
        </property>
    </bean>
```

2：通过读取外部文件：

```
        <?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:p="http://www.springframework.org/schema/p"
               xmlns:c="http://www.springframework.org/schema/c"
               xmlns:context="http://www.springframework.org/schema/context"
               xmlns:util="http://www.springframework.org/schema/util"
        xmlns:aop="http://www.springframework.org/schema/aop"
               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/util
        http://www.springframework.org/schema/util/spring-util.xsd
        http://www.springframework.org/schema/aop
        http://www.springframework.org/schema/aop/spring-aop.xsd
        http://www.springframework.org/schema/tx
        http://www.springframework.org/schema/tx/spring-tx.xsd">
            <!--使用外部的jdbc.properties配置文件-->
```

```
    <context:property-placeholder location="classpath:jdbc.properties"/>
    <!--配置数据库连接池-->
    <bean id="myDataSource"
class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
        <property name="driverClassName" value="${jdbc.driver}"/>
        <property name="url" value="${jdbc.url}"/>
        <property name="username" value="${jdbc.username}"/>
        <property name="password" value="${jdbc.password}"/>
    </bean>
</beans>
```

2.配置sqlsessionfactory

2.1不需要mybatis的配置映射文件

```
  <!--  spring和MyBatis完美整合，不需要mybatis的配置映射文件  -->
    <bean  id="sqlSessionFactory"  class="org.mybatis.spring.SqlSessionFactoryBean">
        <property  name="dataSource"  ref="dataSource"  />
        <!--  自动扫描mapping.xml文件  -->
        <property  name="mapperLocations"  value="classpath:com/cn/hnust/mapping/*.xml">
</property>
    </bean>
    <!--  DAO接口所在包名，Spring会自动查找其下的类  -->
    <bean  class="org.mybatis.spring.mapper.MapperScannerConfigurer">
        <property  name="basePackage"  value="com.cn.hnust.dao"  />
        <property  name="sqlSessionFactoryBeanName"  value="sqlSessionFactory">
</property>
    </bean>
```

2.2通过扫描包的形式：(需要配置文件)

```
<!--jdbc使用Connection，Mybatis SqlSession = Connection-->
    <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="myDataSource"/>
        <property name="configLocation" value="classpath:/mybatis/mybatis-conf.xml"/>
    </bean>
    <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
        <property name="basePackage" value="com.wanho.java99.rbac.auth.*.mapper"/>
    </bean>
```

3.配置service实现类:

```
   <!--管理service实现类(xml or @Service-->
    <context:component-scan base-package="com.wanho.java99.rbac.auth.*.service"/>
```

4.配置事务：

```
<!--jdbc 事务管理器（由spring提供）-->
    <bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
        <property name="dataSource" ref="myDataSource"/>
    </bean>
            <!--事务代理类-->
    <tx:advice id="txAdvice" transaction-manager="transactionManager">
    <tx:attributes>
        <tx:method name="*"/>
    </tx:attributes>
    </tx:advice>
            <!--基于xml对业务代码进行增强(声明式事务)-->
    <aop:config>
    <!--定义切面（对哪些类的哪些方法进行增强）-->
    <aop:pointcut id="serviceMethod" expression="execution(*
com.wanho.java99.service.impl..*(..))"/>
    <aop:advisor advice-ref="txAdvice" pointcut-ref="serviceMethod"/>
    </aop:config>
```