Pom文件

*<!-- 集中定义依赖版本号 -->*<properties>

<!-- 单元测试版本 -->  
 <junit.version>4.12</junit.version>  
 <!-- spring版本 -->

<spring.version>5.2.5.RELEASE</spring.version>  
 <!-- mybatis版本 -->

<mybatis.version>3.5.1</mybatis.version>  
 <!-- mybatis和spring整合版本 -->

<mybatis.spring.version>1.3.1</mybatis.spring.version>  
 <!-- mybatis支持分页插件的版本 -->

<mybatis.paginator.version>1.2.15</mybatis.paginator.version>  
 <!-- mysql版本 -->

<mysql.version>8.0.22</mysql.version>

<!-- 日志处理 -->  
 <slf4j.version>1.6.4</slf4j.version>  
 <!-- 德鲁伊连接池版本 -->

<druid.version>1.1.12</druid.version>  
 <!-- 分页插件自己的版本 -->

<pagehelper.version>5.1.2</pagehelper.version>

<!-- 页面jstl需要 -->  
 <jstl.version>1.2</jstl.version>

<!-- servlet和API版本 -->  
 <servlet-api.version>3.0.1</servlet-api.version>  
 <jsp-api.version>2.0</jsp-api.version>

<!-- jackson版本 -->  
 <jackson.version>2.9.6</jackson.version>  
</properties>  
<dependencies>  
  
 *<!-- spring -->* <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-beans</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jdbc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aspects</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jms</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context-support</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-test</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
 *<!-- Mybatis -->* <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis</artifactId>  
 <version>${mybatis.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis-spring</artifactId>  
 <version>${mybatis.spring.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>com.github.miemiedev</groupId>  
 <artifactId>mybatis-paginator</artifactId>  
 <version>${mybatis.paginator.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>com.github.pagehelper</groupId>  
 <artifactId>pagehelper</artifactId>  
 <version>${pagehelper.version}</version>  
 </dependency>  
 *<!-- MySql -->* <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>${mysql.version}</version>  
 </dependency>  
 *<!-- 连接池 -->* <dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>druid</artifactId>  
 <version>${druid.version}</version>  
 </dependency>  
  
 *<!-- junit -->* <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>${junit.version}</version>  
 <scope>test</scope>  
 </dependency>  
  
  
 *<!-- JSP相关 -->* <dependency>  
 <groupId>jstl</groupId>  
 <artifactId>jstl</artifactId>  
 <version>${jstl.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>3.0.1</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jsp-api</artifactId>  
 <scope>provided</scope>  
 <version>${jsp-api.version}</version>  
 </dependency>  
 *<!-- Jackson Json处理工具包 -->* <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-databind</artifactId>  
 <version>${jackson.version}</version>  
 </dependency>  
 <dependency>  
 <groupId>commons-io</groupId>  
 <artifactId>commons-io</artifactId>  
 <version>2.4</version>  
 </dependency>  
 <dependency>  
 <groupId>commons-fileupload</groupId>  
 <artifactId>commons-fileupload</artifactId>  
 <version>1.3.1</version>  
 </dependency>  
</dependencies>  
  
*<!-- 插件配置 -->*<build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 <encoding>UTF-8</encoding>  
 </configuration>  
 </plugin>  
 </plugins>  
 *<!--识别所有的配置文件-->* <resources>  
 <resource>  
 <directory>src/main/java</directory>  
 <includes>  
 <include>\*\*/\*.properties</include>  
 <include>\*\*/\*.xml</include>  
 </includes>  
 <filtering>false</filtering>  
 </resource>  
 <resource>  
 <directory>src/main/resources</directory>  
 <includes>  
 <include>\*\*/\*.properties</include>  
 <include>\*\*/\*.xml</include>  
 </includes>  
 <filtering>false</filtering>  
 </resource>  
 </resources>  
</build>

jdbc.properties文件

//mysql8 xiaomissm数据库名

jdbc.driver=com.mysql.cj.jdbc.Driver  
jdbc.url=jdbc:mysql://localhost:3306/xiaomissm?useSSL=false&serverTimezone=Asia/Shanghai&allowPublicKeyRetrieval=true  
jdbc.username=root  
jdbc.password=123456

//mysql5 springdb数据库名

jdbc.driver=com.mysql.jdbc.Driver  
jdbc.url=jdbc:mysql://localhost:3306/springdb?characterEncoding=utf-8  
jdbc.username=root  
jdbc.password=123456

Mybatis配置文件

sqlMapConfig.xml //开头字母大写会出问题，

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE configuration  
 PUBLIC "-//mybatis.org//DTD Config 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-config.dtd">  
<configuration>  
 *<!-- 分页插件的设置 -->* <plugins>  
 <plugin interceptor="com.github.pagehelper.PageInterceptor"></plugin>  
 </plugins>  
</configuration>

Spring接管mybatis框架的配置信息

ApplicationContext\_dao.xml //

<?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"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd">  
  
 *<!--  
 把数据库的配置信息，写在单独的文件，便于修改数据库的配置内容  
 location：spring知道jdbc.properties文件的位置  
 -->* <context:property-placeholder location="classpath:jdbc.properties"/>  
  
 *<!-- 创建数据库 -->* <bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">  
 <property name="driverClassName" value="${jdbc.driver}"></property>  
 <property name="url"  
 value="${jdbc.url}"/> *<!-- setUrl设值注入，${配置文件中key值} -->* <property name="username" value="${jdbc.username}"/>  
 <property name="password" value="${jdbc.password}"/>  
 </bean>  
  
 *<!-- 创建SqlSessionFactoryBean -->* <bean class="org.mybatis.spring.SqlSessionFactoryBean">  
 *<!-- 配置数据源 //上面已创建 -->* <property name="dataSource" ref="dataSource"/>  
 *<!-- 配置MyBatis的核心配置文件 -->* <property name="configLocation" value="classpath:sqlMapConfig.xml"/>  
 *<!-- 配置实体类 -->* <property name="typeAliasesPackage" value="com.bjpowernode.pojo"/>  
 </bean>  
 *<!-- 创建mapper文件扫描器 -->* <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">  
 <property name="basePackage" value="com.bjpowernode.mapper"/>  
 </bean>  
</beans>

applicationContext\_service.xml

<?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:tx="http://www.springframework.org/schema/tx"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx.xsd http://www.springframework.org/schema/aop https://www.springframework.org/schema/aop/spring-aop.xsd">  
*<!-- <import resource="applicationContext\_dao.xml"></import>-->  
 <!-- 设置业务逻辑层的包扫描器，指定路径下被@Service注解的类 -->* <context:component-scan base-package="com.bjpowernode.service"/>  
 *<!-- 设置事务管理器 -->* <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">

<!-- 可以通过上方注释import导入数据源，不推荐，可能发生循环导入。建议通配导入 -->  
 <property name="dataSource" ref="dataSource"></property>  
 </bean>  
 *<!-- 添加事务切面 缀tx -->* <tx:advice id="myadvice" transaction-manager="transactionManager">  
 <tx:attributes>

<!-- 查 -->  
 <tx:method name="\*select\*" read-only="true"/>  
 <tx:method name="\*find\*" read-only="true"/>  
 <tx:method name="\*get\*" read-only="true"/>  
 <tx:method name="\*search\*" read-only="true"/>

<!-- 增 -->

<!-- SUPPORTS：支持当前事务，如果当前没有事务，就以非事务方式执行（没有事务） -->  
 <tx:method name="\*insert\*" propagation="REQUIRED"/>  
 <tx:method name="\*add\*" propagation="REQUIRED"/>  
 <tx:method name="\*save\*" propagation="REQUIRED"/>

<!-- 删 -->  
 <tx:method name="\*delete\*" propagation="REQUIRED"/>  
 <tx:method name="\*remove\*" propagation="REQUIRED"/>  
 <tx:method name="\*clear\*" propagation="REQUIRED"/>

<!-- 改 -->  
 <tx:method name="\*update\*" propagation="REQUIRED"/>  
 <tx:method name="\*modify\*" propagation="REQUIRED"/>  
 <tx:method name="\*change\*" propagation="REQUIRED"/>  
 <tx:method name="\*set\*" propagation="REQUIRED"/>

<!-- 以上都不匹配，只支持事务 -->  
 <tx:method name="\*" propagation="SUPPORTS"/>  
 </tx:attributes>  
 </tx:advice>

*<!-- 完成切面和切入点的织入 -->* <aop:config>

<!-- 返回类型任意，在com,bjpowernode.service包下的所有的类的所有方法，任意参数 -->  
 <aop:pointcut id="mypointcut" expression="execution(\* com.bjpowernode.service.\*.\*(..))"/>  
 <aop:advisor advice-ref="myadvice" pointcut-ref="mypointcut"></aop:advisor>  
 </aop:config>  
</beans>

Springmvc配置文件

Springmvc.xml

<?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:mvc="http://www.springframework.org/schema/mvc"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/context https://www.springframework.org/schema/context/spring-context.xsd http://www.springframework.org/schema/mvc https://www.springframework.org/schema/mvc/spring-mvc.xsd">  
 *<!-- 包扫描器 -->* <context:component-scan base-package="com.bjpowernode.controller"/>  
 *<!-- 视图解析器 -->* <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<!-- 注意前后都要有/ -->  
 <property name="prefix" value="/admin/"/>  
 <property name="suffix" value=".jsp"/>  
 </bean>  
 *<!-- 设置文件上传核心组件 注：id不可改 可以properties进行文件上传设置-->* <bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver"/>  
 *<!-- 注解驱动 缀mvc -->* <mvc:annotation-driven></mvc:annotation-driven>  
</beans>

字符编码过滤器

@Service service层类

@Controller controller层类

@Autowired

在所有的界面层，一定会有业务逻辑层的对象

在业务逻辑层中，一定会有数据访问层的对象

界面层获取前端页面传入的数据：方法的参数列表用前端页面标签栏的name

后端对象携带数据传递到前端页面：HttpServletRequest request

request.setAttribute("前端页面el定义对象名", 传输的信息/对象)

admin 管理

mapper 映射器

create Criteria 创建标准

And???EqualTo 查询语句追加条件

selectByExample 通过条件进行查询 -- new Example 创建一个空的对象，可以添加

getMD5 获取加密后的MD5密文

distinct 不同的（去重）

Base\_Column\_List 表中列名

//跳转路径：界面层RequestMapper("/..") target：指定位置，防止左边导航栏被覆盖

前端页面<a href="跳转路径" target="myright">

iframe：可以放进任何页面

分页：

1.当前页显示的？条数据的集合

2.页码的导航显示（一共多少页）

3.当前？页

4.当前页页码背景展示不同

5.每页显示？条