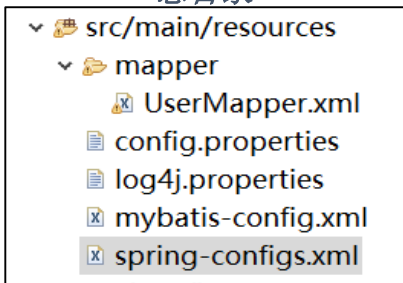


# 天津商会云项目开发手册

- 开发要求: 实现每张数据库表的增删改查功能。
- 使用框架: SSM (Spring + SpringMVC + Mybatis)
- 文件配置:

## 总目录



## config.properties

```
1 driver=com.mysql.jdbc.Driver
2 url=jdbc:mysql:///integral_shop
3 username=root
4 password=root
```

## Web.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID">
3   <display-name>integral-shop-V1.0</display-name>
4   <servlet>
5     <servlet-name>dispatcherServlet</servlet-name>
6     <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
7     <init-param>
8       <param-name>contextConfigLocation</param-name>
9       <param-value>classpath:spring-configs.xml</param-value>
10    </init-param>
11    <load-on-startup>1</load-on-startup>
12  </servlet>
13  <servlet-mapping>
14    <servlet-name>dispatcherServlet</servlet-name>
15    <url-pattern>*.do</url-pattern>
16  </servlet-mapping>
17 </web-app>
```

## log4j.properties

```
log4j.rootLogger = INFO,stdout
log4j.appender.stdout = org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout = org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern = [%-5p] %d{yyyy-MM-dd HH:mm:ss,SSS} %m%n
log4j.logger.com.mybatis3=DEBUG
log4j.logger.com.project=DEBUG
```

## spring-configs.xml

```
<!-- 配置bean扫描 -->
<context:component-scan base-package="com.pb" />
<!-- 启用mvc注解功能 -->
<mvc:annotation-driven />
<!-- 配置视图解析器(解析modelAndView对象中封装的view信息) -->
<bean id="viewResolver"
      class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/WEB-INF/pages/" />
  <property name="suffix" value=".html" />
</bean>
<!-- 配置DRUID连接池 -->
<util:properties id="cfg" location="classpath:config.properties" />
<bean id="myDataSource" class="com.alibaba.druid.pool.DruidDataSource">
  <property name="driverClassName" value="#{cfg.driver}" />
  <property name="url" value="#{cfg.url}" />
  <property name="username" value="#{cfg.username}" />
  <property name="password" value="#{cfg.password}" />
</bean>
<!-- 配置mybatis依赖 -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="myDataSource" />
  <property name="configLocation" value="classpath:mybatis-config.xml"/>
</bean>
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
  <property name="basePackage" value="com.pb.dao" />
</bean>
</beans>
```

## mybatis-config.xml

```
<?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">
<!-- 这是MyBatis的核心配置文件 -->
<configuration>
  <settings>
    <!-- mybatis控制台输出 -->
    <setting name="logImpl" value="LOG4J" />
  </settings>
  <typeAliases>
    <package name="com.pb.pojo"/>
  </typeAliases>
  <mappers>
    <mapper resource="mapper/UserMapper.xml"/>
  </mappers>
</configuration>
```

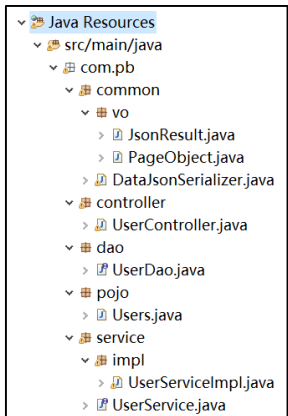
## mapper/Mappper.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.pb.dao.UserDao">

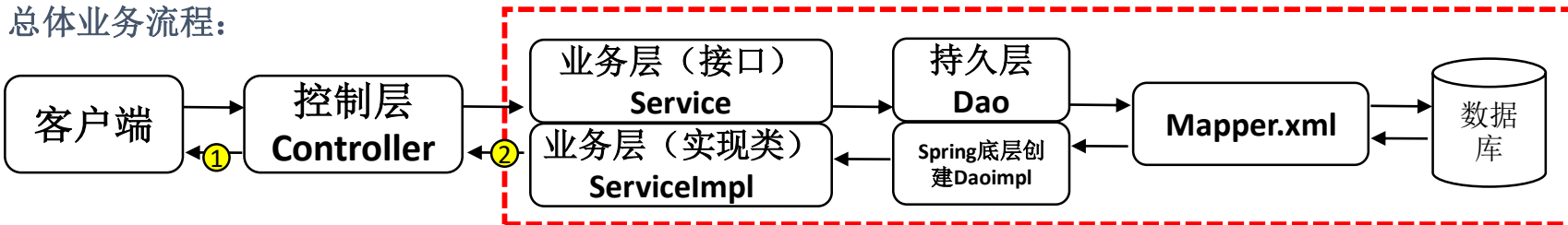
  相应增删改查的SQL语句

</mapper>
```

## • 目录配置



总体业务流程:



Note: 1.数据通过com.pb.pojo.Users封装。

2. ① 为控制层通过封装JsonResult返回给客户端包括: 状态码、消息、具体封装的数据。

3. ② 为业务层封装页面信息, 用于分页显示查询结果。

## • 设计要求

1. 仿照案例, 正确配置SSM相应文件, 要求使用“DRUID连接池, Log4J”, 要求使用Maven, 具体依赖版本参考项目中的Pom文件。
2. 仿照案例, 正确配置目录: “业务层→持久层 → Mapper → 数据库对应表单”。
3. 能够通过JUnit测试, 实现业务层“增删改查”。