武汉纺织大学

Web应用开发课程设计

**基于SSM和Bootstrap的系统主页设计**

**学 院： 数学与计算机学院**

**班 级： 物联网11801**

**姓 名： 王润鹏**

**学 号： 1814290221**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月15日**

目 录

[1 需求分析 1](#_Toc59227682)

[1.1菜单设计 1](#_Toc59227683)

[1.2轮播图设计 1](#_Toc59227684)

[2 系统设计 1](#_Toc59227685)

[2.1轮播图、菜单与数据库交互 1](#_Toc59227686)

[2.2 UML类图（Class Diagram） 2](#_Toc59227687)

[2.4 整体流程 3](#_Toc59227688)

[3 系统实现 4](#_Toc59227689)

[3.1 项目结构 4](#_Toc59227690)

[3.2 配置文件 4](#_Toc59227691)

[3.2.1 jdbc.properties文件 4](#_Toc59227692)

[3.2.2 web.xml 4](#_Toc59227693)

[3.2.3 mybatis.xml文件 6](#_Toc59227694)

[3.2.4 spring-mybatis 6](#_Toc59227695)

[3.2.4 spring-mvc.xml 8](#_Toc59227696)

[3.2.5porm.xml 8](#_Toc59227697)

[3.3 VO类 13](#_Toc59227698)

[3.3.1 Menu.java 13](#_Toc59227699)

[3.3.2Carousel.java 14](#_Toc59227700)

[3.4 DAO接口类 15](#_Toc59227701)

[3.4.1 IMenuDAO.java 15](#_Toc59227702)

[3.4.2 ICarouselDAO.java 15](#_Toc59227703)

[3.5 接口映射文件 15](#_Toc59227704)

[3.5.1menuMapper.xml 16](#_Toc59227705)

[3.5.2 carouselMapper.xml 16](#_Toc59227706)

[3.6 服务层Service 16](#_Toc59227707)

[3.6.1 ICarouselService.java 16](#_Toc59227708)

[3.6.2 IMenuService.java 16](#_Toc59227709)

[3.6.3 MenuSerivceImpl.java 16](#_Toc59227710)

[3.6.4CarouselServiceImpl.java 17](#_Toc59227711)

[3.7 controller类 17](#_Toc59227712)

[3.7.1 MainPageController.java 17](#_Toc59227713)

[3.7.2 MainPageCarouselController.java 18](#_Toc59227714)

[3.8 前端代码mainpage.java 18](#_Toc59227715)

[3.8.1前端js部分 18](#_Toc59227716)

[3.8.2 html部分（其中有js的代码用来查询轮播图） 20](#_Toc59227717)

[4 系统测试 23](#_Toc59227718)

[4.1网页总体图一览 23](#_Toc59227719)

[4.2菜单功能和轮播图功能一览 24](#_Toc59227720)

[5 系统总结 24](#_Toc59227721)

# 1 需求分析

大多数公司、学校的都需要有方便便于管理的系统主页。方便的系统主页有利于网站后序的建设和发展。首先系统主页的导航栏的连接应该具有实时更新，实时修改的功能。其次轮播图可以更好的展示一个网站的特色，吸引进入网站的用户，也可以更加方便的传达出想表达的事情。接下来一个好的系统主页也应该在整体设计是上具有简洁性和美观性。

## 1.1菜单设计

菜单设计，采用bootstrap4框架的nav，每一个菜单中的选项都可以生成下拉菜单，当鼠标悬浮时，下拉菜单会出现，其中每个链接都来自于数据库当中。样式如下



生成下拉菜单如下



## 1.2轮播图设计

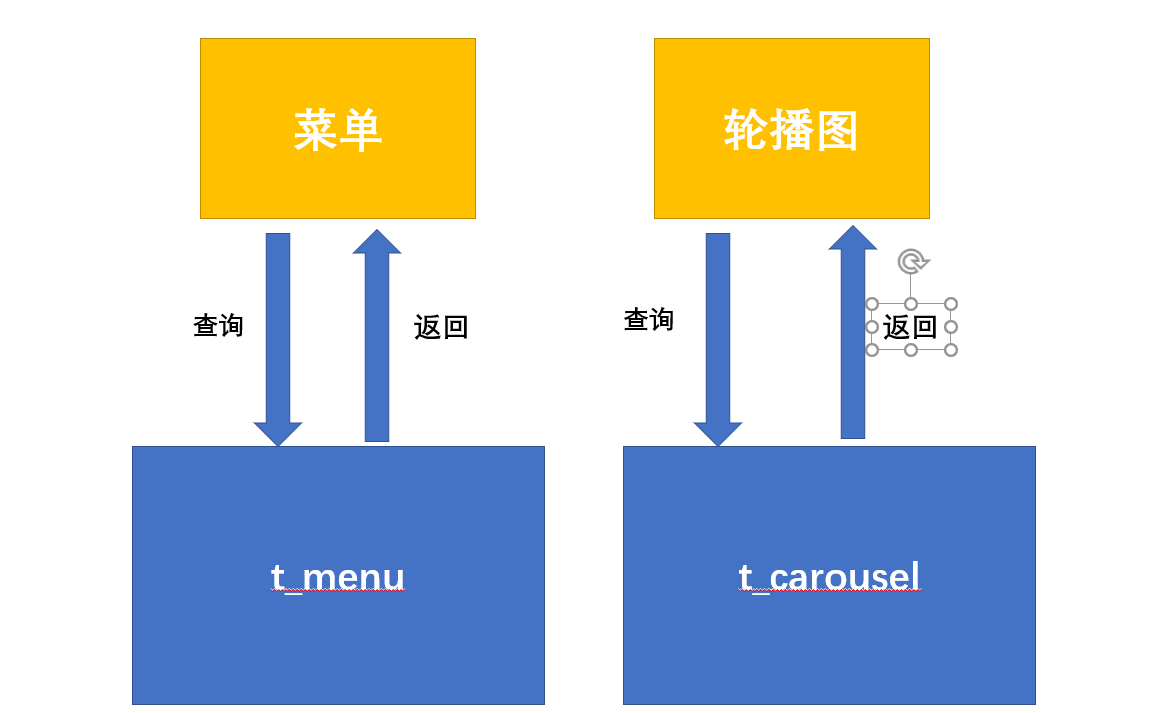
轮播图采用bootstrap4的轮播图组件，其中的描述和图片来自于数据库里面，动态生成



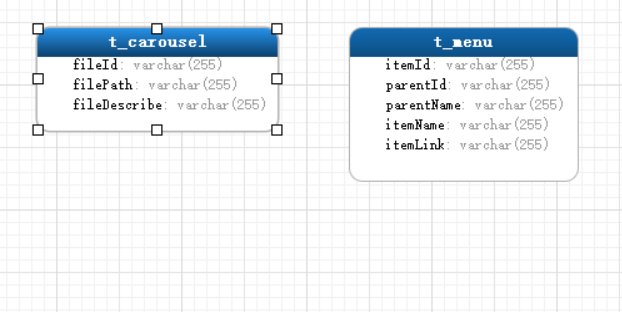
# 2 系统设计

## 2.1轮播图、菜单与数据库交互

交互模式如下



数据库ER图如下



## 2.2 UML类图（Class Diagram）

本次设计所用的类如下

1.Menu类和Carousel类

vo类，每一个成员变量和数据库中每一个量相同

2.IMenuDAO和ICarouselDAO类

于mybatis的映射文件相对应，是接口类

3.IMenuService和ICarouselService类

接口类，处理业务逻辑服务

4.MenuServiceImpl和CarouselSeriveImpl类

接口实现类

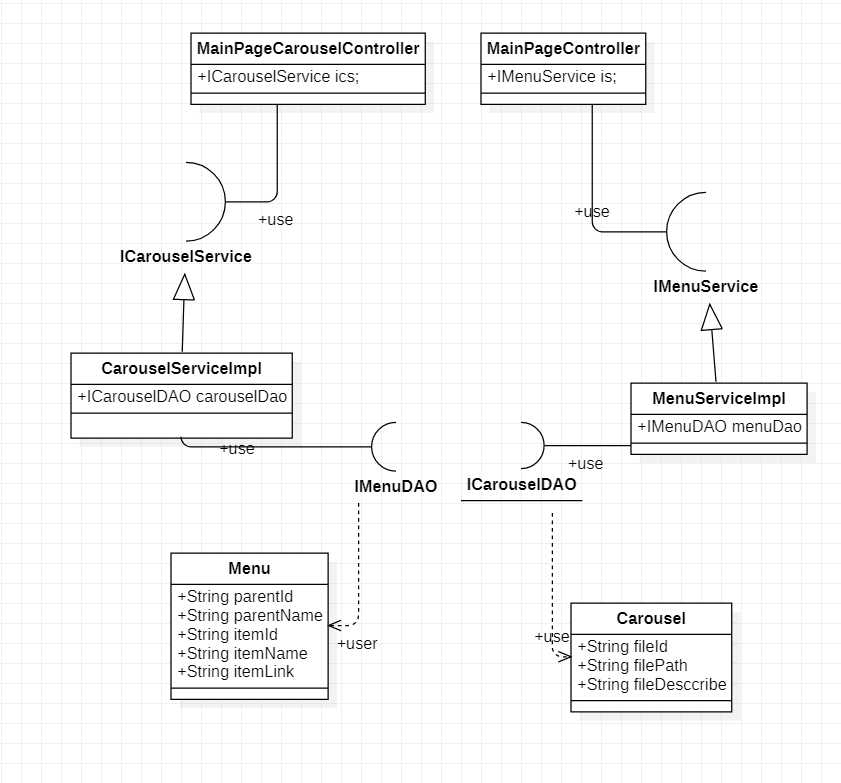
5.MainPageController类

负责将菜单和连接以json的方式发送到前端的ajax的success中继续执行操作

6.MainPageCarouselcontroller类

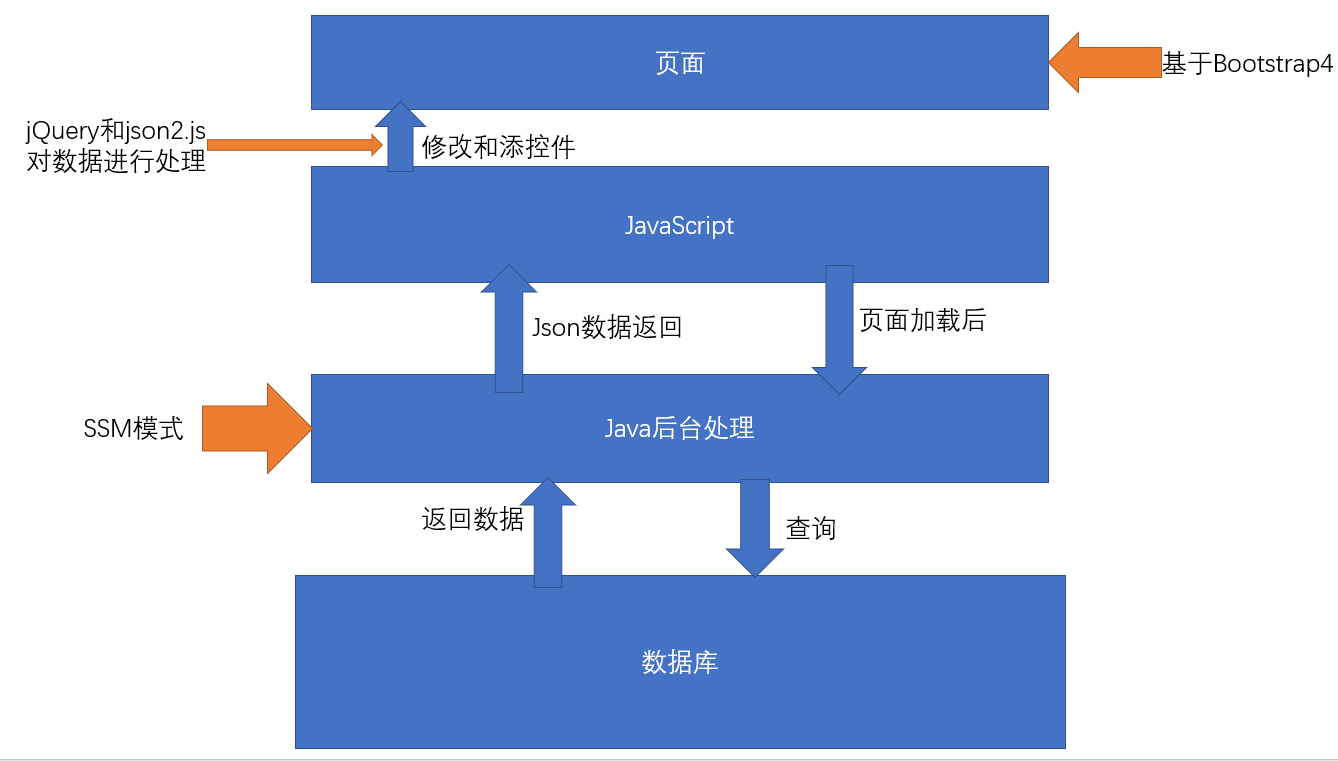
负责将轮播图的图片内容以json方式发送到前端

各类的结构及类之间的关系如下：



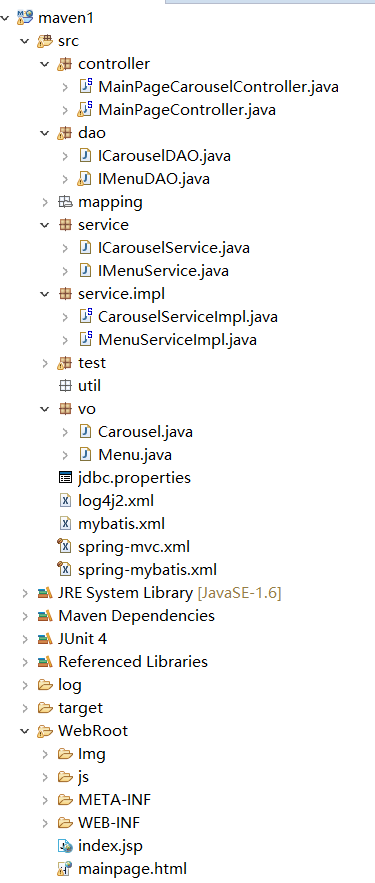
## 2.4 整体流程

整体流程如图所示，通过前端页面接收来自后端的json数据，进行处理后，达到效果



# 3 系统实现

## 3.1 项目结构



## 3.2 配置文件

### 3.2.1 jdbc.properties文件

配置连接数据库

driver=com.mysql.cj.jdbc.Driver

url=jdbc:mysql://localhost:3306/mainpage?useUnicode=true&characterEncoding=utf-8&useSSL=false&serverTimezone=UTC

username=root

password=\*\*\*\*\*\*

initialSize=0

maxActive=20

minIdle=1

maxWait=60000

validationQuery =SELECT 1

### 3.2.2 web.xml

Webapp的配置文件

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>maven1</display-name>

<welcome-file-list>

<welcome-file>index.html</welcome-file>

<welcome-file>mainpage.html</welcome-file>

</welcome-file-list>

<!-- Spring和mybatis的配置文件 -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring-mybatis.xml</param-value>

</context-param>

<!-- log4j2的配置文件 -->

<context-param>

<param-name>log4jConfigLocation</param-name>

<param-value>classpath:log4j2.xml</param-value>

</context-param>

<!-- 编码过滤器 -->

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<async-supported>true</async-supported>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- Spring监听器 -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- 防止Spring内存溢出监听器 -->

<listener>

<listener-class>org.springframework.web.util.IntrospectorCleanupListener</listener-class>

</listener>

<servlet>

<servlet-name>SpringMVC</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:spring-mvc.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

<async-supported>true</async-supported>

</servlet>

<servlet>

<servlet-name>TestController</servlet-name>

<servlet-class>controller.TestController</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SpringMVC</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

</web-app>

### 3.2.3 mybatis.xml文件

Mybatis配置文件，不过主要依赖spring-mybatis.xml，没有这个文件会报错

<?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>

<settings>

<setting name="logImpl" value="LOG4J2" />

</settings>

</configuration>

### 3.2.4 spring-mybatis

负责配置数据库，整合spring和mybatis，实现自动

<?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*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd"*>

<!-- 扫描类包，将标注Spring注解的类自动转化Bean，同时完成Bean的注入 -->

<context:component-scan base-package=*"service"* />

<!-- 引入jdbc配置文件 -->

<bean id=*"propertyConfigurer"*

class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"location"* value=*"classpath:jdbc.properties"* />

</bean>

<!-- 数据库连接池 -->

<bean id=*"dataSource"* class=*"com.alibaba.druid.pool.DruidDataSource"*

destroy-method=*"close"*>

<property name=*"driverClassName"* value=*"${driver}"* />

<property name=*"url"* value=*"${url}"* />

<property name=*"username"* value=*"${username}"* />

<property name=*"password"* value=*"${password}"* />

<!-- 初始化连接大小 -->

<property name=*"initialSize"* value=*"${initialSize}"*></property>

<!-- 连接池最大数量 -->

<property name=*"maxActive"* value=*"${maxActive}"*></property>

<!-- 连接池最小空闲 -->

<property name=*"minIdle"* value=*"${minIdle}"*></property>

<!-- 获取连接最大等待时间 -->

<property name=*"maxWait"* value=*"${maxWait}"*></property>

<!-- 验证连接是否可用，使用的SQL语句 -->

<property name=*"validationQuery"* value=*"SELECT 1"*></property>

</bean>

<!-- spring和MyBatis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"* />

<property name=*"configLocation"* value=*"classpath:mybatis.xml"*></property>

<!-- 自动扫描mapping.xml文件 -->

<property name=*"mapperLocations"* value=*"classpath: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=*"dao"* />

<property name=*"sqlSessionFactoryBeanName"* value=*"sqlSessionFactory"*></property>

</bean>

<!-- (事务管理)transaction manager-->

<bean id=*"transactionManager"*

class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>

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

</bean>

</beans>

### 3.2.4 spring-mvc.xml

Spring配置文件

<?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:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"* 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/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd*

*http://www.springframework.org/schema/tx*

*http://www.springframework.org/schema/tx/spring-tx.xsd*

*http://www.springframework.org/schema/aop*

*http://www.springframework.org/schema/aop/spring-aop.xsd"*>

<mvc:annotation-driven />

<!-- 扫描类包，将标注Spring注解的类自动转化Bean，同时完成Bean的注入 -->

<context:component-scan base-package=*"controller"* />

<context:component-scan base-package=*"service"* />

<!--配置视图解析器 -->

<!-- ViewResolver 视图解析器 用于将返回的ModelAndView对象进行分离

InternalResourceViewResolver：用于支持Servlet、JSP视图解析；

viewClass：JstlView表示JSP模板页面需要使用JSTL标签库，classpath中必须包含jstl的相关jar包； prefix和suffix：查找视图页面的前缀和后缀（前缀[逻辑视图名]后缀），

比如传进来的逻辑视图名为hello，则该该jsp视图页面应该存放在“/hello.jsp”； -->

<bean

class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/"*></property> <!-- 视图放在webroot/下 -->

<property name=*"suffix"* value=*".jsp"*></property>

</bean>

</beans>

### 3.2.5porm.xml

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>maven1</groupId>

<artifactId>maven1</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>maven1</name>

<description/>

<properties>

<webVersion>3.0</webVersion>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<springversion>5.0.0.RELEASE</springversion> <!-- spring版本号 -->

<mybatisversion>3.3.0</mybatisversion> <!-- mybatis版本号 -->

<log4jversion>2.11.2</log4jversion> <!-- log4j版本号 -->

</properties>

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.0.1</version>

<!-- 用于编译和测试的环境，将不会被打包到lib目录下。 -->

<scope>provided</scope>

</dependency>

<!--springframework -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${springversion}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${springversion}</version>

</dependency>

<!-- log4j2，用于日志管理 -->

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-api</artifactId>

<version>${log4jversion}</version>

</dependency>

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-core</artifactId>

<version>${log4jversion}</version>

</dependency>

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-web</artifactId>

<version>${log4jversion}</version>

</dependency>

<!-- junit单元测试 -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.12</version>

<!-- 表示开发的时候引入，发布的时候不会加载此包 -->

<scope>test</scope>

</dependency>

<!-- 用于spingMVC中的java对象和json自动转换 -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.9.1</version>

</dependency>

<!-- mybatis核心包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>${mybatisversion}</version>

</dependency>

<!-- mybatis/spring包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<!--mysql驱动包 -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.20</version>

</dependency>

<!-- 连接池，用来加载DruidDataSource -->

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>druid</artifactId>

<version>1.0.9</version>

</dependency>

<!-- JSTL标签库 -->

<dependency>

<groupId>jstl</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src</sourceDirectory>

<resources>

<resource>

<directory>src</directory>

<excludes>

<exclude>\*\*/\*.java</exclude>

</excludes>

</resource>

</resources>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>2.3.2</version>

<configuration>

<source>1.6</source>

<target>1.6</target>

</configuration>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>2.6</version>

<configuration>

<warSourceDirectory>${basedir}/WebRoot</warSourceDirectory>

<failOnMissingWebXml>false</failOnMissingWebXml>

</configuration>

</plugin>

</plugins>

</build>

</project>

## 3.3 VO类

## 3.3.1 Menu.java

**public** **class** Menu {

**private** String parentName;

**private** String itemName;

**private** String itemLink;

**private** String itemId;

**private** String parentId;

**public** Menu(String parentName, String itemName, String itemLink, String itemId, String parentId) {

**super**();

**this**.parentName = parentName;

**this**.itemName = itemName;

**this**.itemLink = itemLink;

**this**.itemId = itemId;

**this**.parentId = parentId;

}

**public** Menu() {

**super**();

}

**public** String getParentName() {

**return** parentName;

}

**public** **void** setParentName(String parentName) {

**this**.parentName = parentName;

}

**public** String getItemName() {

**return** itemName;

}

**public** **void** setItemName(String itemName) {

**this**.itemName = itemName;

}

**public** String getItemLink() {

**return** itemLink;

}

**public** **void** setItemLink(String itemLink) {

**this**.itemLink = itemLink;

}

**public** String getItemId() {

**return** itemId;

}

**public** **void** setItemId(String itemId) {

**this**.itemId = itemId;

}

**public** String getParentId() {

**return** parentId;

}

**public** **void** setParentId(String parentId) {

**this**.parentId = parentId;

}

@Override

**public** String toString() {

**return** "Menu [parentName=" + parentName + ", itemName=" + itemName + ", itemLink=" + itemLink + ", itemId="

+ itemId + ", parentId=" + parentId + "]";

}

}

### 3.3.2Carousel.java

**package** vo;

**public** **class** Carousel {

**private** String fileId;

**private** String filePath;

**private** String fileDescribe;

**public** Carousel(String fileId, String filePath, String fileDescribe) {

**super**();

**this**.fileId = fileId;

**this**.filePath = filePath;

**this**.fileDescribe = fileDescribe;

}

**public** Carousel() {

**super**();

}

**public** String getFileId() {

**return** fileId;

}

**public** **void** setFileId(String fileId) {

**this**.fileId = fileId;

}

**public** String getFilePath() {

**return** filePath;

}

**public** **void** setFilePath(String filePath) {

**this**.filePath = filePath;

}

**public** String getFileDescribe() {

**return** fileDescribe;

}

**public** **void** setFileDescribe(String fileDescribe) {

**this**.fileDescribe = fileDescribe;

}

@Override

**public** String toString() {

**return** "Carousel [fileId=" + fileId + ", filePath=" + filePath + ", fileDescribe=" + fileDescribe + "]";

}

}

## 3.4 DAO接口类

与mybatis的映射文件对应

### 3.4.1 IMenuDAO.java

**public** **interface** IMenuDAO {

**public** ArrayList<Menu> getAll() ;

}

### 3.4.2 ICarouselDAO.java

**public** **interface** ICarouselDAO {

**public** ArrayList<Carousel> getAll();

}

## 3.5 接口映射文件

负责实现对数据库的增删改查

### 3.5.1menuMapper.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=*"dao.IMenuDAO"*>

<select id=*"getAll"* resultType=*"vo.Menu"*>

select \* from t\_menu

</select>

</mapper>

### 3.5.2 carouselMapper.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=*"dao.ICarouselDAO"*>

<select id=*"getAll"* resultType=*"vo.Carousel"*>

select \* from t\_carousel

</select>

</mapper>

## 3.6 服务层Service

业务逻辑层的操作

### 3.6.1 ICarouselService.java

**public** **interface** ICarouselService {

**public** ArrayList<Carousel> getAllCarouselMsg();

}

### 3.6.2 IMenuService.java

**public** **interface** IMenuService {

**public** ArrayList<Menu> getAllMenuMsg() ;

}

### 3.6.3 MenuSerivceImpl.java

@Service

**public** **class** MenuServiceImpl **implements** IMenuService {

@Autowired

**private** IMenuDAO menuDao;

**public** ArrayList<Menu> getAllMenuMsg()

{

**return** menuDao.getAll();

}

}

### 3.6.4CarouselServiceImpl.java

@Service

**public** **class** CarouselServiceImpl **implements** ICarouselService{

@Autowired

**private** ICarouselDAO carouselDao;

@Override

**public** ArrayList<Carousel> getAllCarouselMsg() {

// **TODO** Auto-generated method stub

**return** carouselDao.getAll();

}

}

## 3.7 controller类

实现对前端页面的控制

### 3.7.1 MainPageController.java

将查询到的导航栏内容返回给前端

@Controller

**public** **class** MainPageController {

@Autowired

**private** IMenuService is;

@RequestMapping(value = "/getMenuContent.do", method = RequestMethod.***POST***)

@ResponseBody

**public** ArrayList<Menu> getToNav(HttpSession session, HttpServletResponse response){

ArrayList<Menu> mlist=is.getAllMenuMsg();

**return** mlist;

}

}

### 3.7.2 MainPageCarouselController.java

将查询的轮播图内容返回给前端，用json类型

@Controller

**public** **class** MainPageCarouselController {

@Autowired

**private** ICarouselService ics;

@RequestMapping(value = "/getCarouselContent.do", method = RequestMethod.***POST***)

@ResponseBody

**public** ArrayList<Carousel> getToCarousel(HttpSession session, HttpServletResponse response)

{

ArrayList<Carousel> clist=ics.getAllCarouselMsg();

**return** clist;

}

}

## 3.8 前端代码mainpage.java

### 3.8.1前端js部分

<script type=*"text/javascript"*>

window.onload=**function**(){

//做导航的处理

$.ajax({

type:"post",

url:"getMenuContent.do",

data:{},

dataType:"json",

success:**function**(response)

{

doMenu(response);

}

});

**function** bindLink(tool,str)

{

tool.click(

**function**()

{

//false来阻止冒泡，动态修改href没用，只能添加点击事件来跳转

window.location.href=str;

**return false**;

}

);

}

**function** doMenu(res)

{

**for**(**var** i=0;i<res.length;i++)

{

console.log(res[i].parentId);

**if**(res[i].itemId==res[i].parentId)

{

**var** tool=$("#"+res[i].parentId.toString()).find("a");

**var** str=res[i].itemLink;

bindLink(tool,str);

}

**else**

{

//说明要绑定下拉菜单

**if**($("#"+res[i].parentId.toString()).find("div").length==0)

{

$("#"+res[i].parentId.toString()).append('<div class="dropdown-menu" style="color: #000000;">'+

'<a class="dropdown-item" href="'+res[i].itemLink+'">'+res[i].itemName+'</a>'+

'</div>');

}

**else**

{

$("#"+res[i].parentId.toString()).find("div").append('<a class="dropdown-item" href="'+res[i].itemLink+'">'+res[i].itemName+'</a>');

console.log(res[i].itemLink);

}

}

}

**return true**;

}

**function** fuwuClikeEvent(srcStr1,srcStr2,content1,content2)

{

$("#fuwu1pic").attr('src',srcStr1);

$("#fuwu2pic").attr('src',srcStr2);

$("#fuwu1text").text(content1);

$("#fuwu2text").text(content2);

}

$("#hjjiance").click(**function**(){

fuwuClikeEvent("Img/jiance1.jpg","Img/jiance2.jpg","服务检测包您满意","高效率高性价比");

}

);

$("#swfenxi").click(**function**(){

fuwuClikeEvent("Img/fenxi1.jpg","Img/fenxi2.jpg","高新技术前沿检测","顶级团队专业设备");

});

$("#wrjiance").click(

**function**(){

fuwuClikeEvent("Img/wuran1.jpg","Img/wuran2.jpg","实时监控保证安全","服务到位业界好评");

}

);

};

### 3.8.2 html部分（其中有js的代码用来查询轮播图）

<body>

<div id=*"header"* class=*"align-items-lg-center row bccolorThree p-2 text-white fontOne font-weight-light"*>

<div class=*"col-md-2 ml-3"*>

当前登录:

<a id=*"login"* href=*"#"*><small>点我登录</small></a>

</div>

<div class=*"col-md-3 offset-md-6 mr-1 input-group h-75"*>

<input type=*"text"* class=*"form-control"*/>

<div class=*"input-group-append"*>

<button class=*"btn bccolorOne"* type=*"submit"*>Q</button>

</div>

</div>

</div>

<nav class=*"navbar navbar-expand-sm fontOne pl-5 pr-5 bccolorTwo navbar-dark"*>

<a class=*"navbar-brand"* href=*"#"*>

<img src=*"Img/LOGO.png"* alt=*"logo"* style="width:*90px*;">

</a>

<ul class=*"navbar-nav"* id=*"navDiv"*>

<li class=*"nav-item"* id=*"shouye"*>

<a class=*"nav-link"* href=*"javaScript:;"* id=*"navbardrop"* data-toggle=*"dropdown"*>

首页</a>

</li>

<li class=*"nav-item dropdown"* id=*"gsgaikuang"*>

<a class=*"nav-link "* href=*"#"* id=*"navbardrop"* data-toggle=*"dropdown"*>

公司概况

</a>

</li>

<li class=*"nav-item dropdown"* id=*"zhkeji"*>

<a class=*"nav-link "* href=*"#"* id=*"navbardrop"* data-toggle=*"dropdown"*>

智慧科技

</a>

</li>

<li class=*"nav-item dropdown"* id=*"fwzhichi"*>

<a class=*"nav-link "* href=*"#"* id=*"navbardrop"* data-toggle=*"dropdown"*>

服务支持

</a>

</li>

<li class=*"nav-item dropdown"* id=*"rczhaopin"*>

<a class=*"nav-link "* href=*"#"* id=*"navbardrop"* data-toggle=*"dropdown"*>

人才招聘

</a>

</li>

<li class=*"nav-item dropdown"* id=*"gywomen"*>

<a class=*"nav-link"* href=*"#"* id=*"navbardrop"* data-toggle=*"dropdown"*>

关于我们

</a>

</li>

</ul>

</nav>

<script type=*"text/javascript"*>

**function** doCarousel(res)

{

**for**(**var** i=0;i<res.length;i++)

{

**if**($("#picInner").find("div").eq(0).length==0)

{

//首次

$("#picInner").append('<div class="carousel-item active" '+res[i].fileId+'>'+

'<img src="'+res[i].filePath+'">'+

'<div class="carousel-caption">'+

'<p class="fontThree">'+res[i].fileDescribe+'</p>'+

'</div>'+

'</div>');

$("#zhishifu").append('<li data-target="#demo" data-slide-to='+i+' class="active"></li>');

}

**else**

{

$("#picInner").append('<div class="carousel-item" '+res[i].fileId+'>'+

'<img src="'+res[i].filePath+'">'+

'<div class="carousel-caption">'+

'<p class="fontThree">'+res[i].fileDescribe+'</p>'+

'</div>'+

'</div>');

$("#zhishifu").append('<li data-target="#demo" data-slide-to='+i+'></li>');

}

}

}

$.ajax({

type:"post",

url:"getCarouselContent.do",

data:{},

dataType:"json",

success:**function**(response)

{

doCarousel(response);

}

});

</script>

<div id=*"demo"* class=*"carousel slide"* data-ride=*"carousel"*>

<!-- 指示符 -->

<ul class=*"carousel-indicators"* id=*"zhishifu"*>

</ul>

<!-- 轮播图片 -->

<div class=*"carousel-inner"* id=*"picInner"*>

</div>

</div>

<!-- 左右切换按钮 -->

<a class=*"carousel-control-prev"* href=*"#demo"* data-slide=*"prev"*>

<span class=*"carousel-control-prev-icon"*></span>

</a>

<a class=*"carousel-control-next"* href=*"#demo"* data-slide=*"next"*>

<span class=*"carousel-control-next-icon"*></span>

</a>

</div>

<div id=*"introduce"* class=*"mt-5 w-100"*>

<p class=*"text-center fontTwo"*>我们的服务</p>

</div>

<div id=*"introduceNav"* class=*"container mb-5"*>

<div class=*"border border-top-0 border-left-0 border-right-0 border-bottom d-flex text-center w-100 mb-5"*>

<div class=*"p-5 flex-fill"*><a class=*"fontThree"* id=*"hjjiance"* href=*"javaScript:;"*>环境检测</a></div>

<div class=*"p-5 flex-fill"*><a class=*"fontThree"* id=*"swfenxi"* href=*"javaScript:;"*>生物分析</a></div>

<div class=*"p-5 flex-fill"*><a class=*"fontThree"* id=*"wrjiance"* href=*"javaScript:;"*>污染监测</a></div>

<div class=*"p-5 flex-fill"*><a class=*"fontThree"* id=*"hjzhengming"* href=*"javaScript:;"*>环检证明</a></div>

</div>

<div class=*"d-flex flex-row"*>

<div class=*"border mr-5 ml-5 p-1"* style="width:*40%*;">

<img src=*"Img/jiance1.jpg"* id=*"fuwu1pic"* alt=*"John Doe"* class=*"mr-1 mt-1 rounded d-flex mx-auto"* style="width:*90%*; height: *70%*;">

<div class=*"media-body"*>

<p id=*"fuwu1text"* class=*"fontOne text-center text-black"*>环检检测包您满意</p>

</div>

</div>

<div class=*"border mr-5 ml-5 p-1"* style="width:*40%*;">

<img src=*"Img/jiance2.jpg"* id=*"fuwu2pic"* alt=*"John Doe"* class=*"mr-1 mt-1 rounded d-flex mx-auto"* style="width:*90%*; height: *70%*;">

<div class=*"media-body"*>

<p id=*"fuwu2text"* class=*"fontOne text-center text-black"*>高效率高性价比</p>

</div>

</div>

</div>

</div>

<div id=*"bottomDiv"* class=*"row bccolorThree colorOne mb-0 mr-0 ml-0 p-3"*>

<div class=*"col-2"*>

<a>联系我们:</a>

<br>

<a>&nbsp;tele:15975451557</a>

<br>

<a>&nbsp;Email:2077il@kunlun.com</a>

</div>

<div class=*"col-4 border border-secondary border-left-0 border-bottom-0 border-top-0 border-right"*>

</div>

<div class=*"col-3 offset-3"*>

<a>昆仑环检检测科技有限公司版权所有 © 2077 Copyrights all reserved ICP备xxxxx号-0x 网备xxxxxxxxxxxxxx号</a>

</div>

</div>

<div class=*"bccolorTwo text-center text-white-50 mb-0 mr-0 mt-0 ml-0 p-1"*>

仅供个人学习，图标虚构公司不存在，图片来源网络，侵删致歉

</div>

</body>

</html>

# 4 系统测试

## 4.1网页总体图一览



## 4.2菜单功能和轮播图功能一览



# 5 系统总结

这次的选题，难度并不是很大。但是实践过程中也出了很多问题。但是我体会到了，出现问题解决了问题，更加有利于去了解这次涉及到的mybatis、spring和bootstrap以及jQuery。当然这次选题，在后端数据库的操作并不是很多，也比较容易。这次在前端部分花的功夫占比会更多一些。但是后端的mybatis在上学期的java课程设计里面也学到了很多。我感觉从老师这里学到了很多。当然还有很多不足的地方，比如我对maven理解和使用并不是很到位，还有spring的使用，这些是日后需要慢慢加强的部分。