武汉纺织大学

Web应用开发课程设计

**新闻展示页面**

**学 院： 数学与计算机学院**

**班 级： 物联网11803**

**姓 名： 徐一帆**

**学 号： 1804280319**

**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](#_Toc59211922)

[2 系统设计 1](#_Toc59211923)

[3 系统实现 2](#_Toc59211924)

[3.1 项目结构 2](#_Toc59211925)

[3.2 相关代码文件 3](#_Toc59211926)

[3.2.1 NewsController.java文件 3](#_Toc59211927)

[3.2.2 NewsService.java文件 3](#_Toc59211928)

[3.2.3 Newdao.java文件 4](#_Toc59211929)

[3.3.4 News.java 4](#_Toc59211930)

[3.3.5NewsDao.xml 5](#_Toc59211931)

[3.3.6 application-spring.xml 6](#_Toc59211932)

[3.3.7 mybatis-config.xml 7](#_Toc59211933)

[3.3.8 spring-mvc.xml 7](#_Toc59211934)

[3.3.9 web.xml 8](#_Toc59211935)

[3.3.10 前端用户界面news.html 9](#_Toc59211936)

[4 系统测试 13](#_Toc59211937)

[5 系统总结 13](#_Toc59211938)

# 1 需求分析

按下图设计实现新闻显示页面，页面所需图片，文字均来自数据库。



# 2 系统设计

如图所示就是个新闻显示页面，整体的流程就是从前端打开页面，然后发送请求获取后端的数据，然后从数据库中获取到数据后到前端进行渲染。前端发出请求过后，进入到NewsController这一层，通过NewsController进入到NewsService方法，随后调用NewService的queryNews，通过queryNews中的NewsDao去查询数据库里的数据，随后返回到前端，也就是NewsController。前端获取到新闻页面的数据后进行渲染。

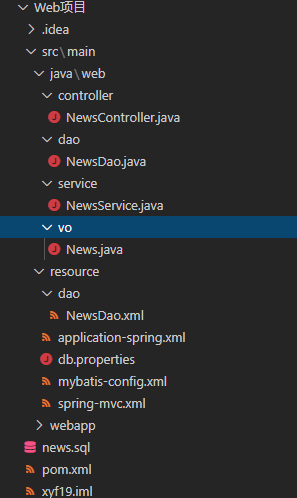
整个项目中用到的前端技术就是一个bootstrap，它是一个前端UI库，里面封装了很多css的样式，还用到了一个jquery，它是一个js库。然后还使用了一个vue.js的一个框架，这个框架在页面中就是进行绑定的功能，将数据和页面上的元素进行绑定，这样就可以通过后端返回的数据进行渲染。

后端是使用了spring，spring mvc，mybatis三个框架，spring mvc的作用就是在controller这一层，接受前端的请求后，通过service的等等一系列的处理，然后返回数据给前端。Spring的作用在整个后端，它参与了整个后端的运作。Mybatis是作用在Dao层，通过NewsDao.java和NewsDao.xml去获取数据或者更改数据。

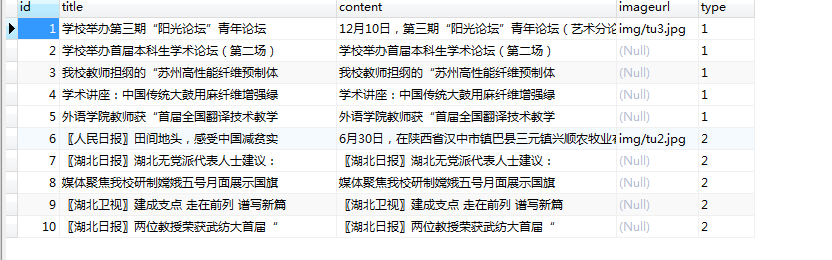
# 3 系统实现

## 3.1 项目结构

Web项目由前端代码，后端代码，配置文件，第三方库组成，典型目录如下。src目录：java源代码及配置文件的存放目录。可根据需要在该目录下创建包，如controller，model，tools，config等。



数据库界面如下图。



## 3.2 相关代码文件

## 3.2.1 NewsController.java文件

该文件是新闻显示页面的主要控制方法。

package web.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import web.service.NewsService;

import web.vo.News;

import java.util.List;

@Controller

public class NewsController {

    @Autowired

    private NewsService newsService;

    // 查询新闻

    @RequestMapping("/queryNews")

    @ResponseBody

    public List<News> queryNews(String type){

        List<News> news = newsService.queryNews(type);

        return news;

    }

}

## 3.2.2 NewsService.java文件

描述该文件作用，并对里面的配置进行解释

package web.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import web.dao.NewsDao;

import web.vo.News;

import java.util.List;

@Service

public class NewsService {

    @Autowired

    private NewsDao newsDao;

    // 查询新闻

    public List<News> queryNews(String type) {

        return newsDao.selectNews(type);

    }

}

## 3.2.3 Newdao.java文件

描述该文件作用，并对里面的配置进行解释

package web.dao;

import web.vo.News;

import java.util.List;

public interface NewsDao {

    // 查询新闻

    List<News> selectNews(String type);

}

## 3.3.4 News.java

描述该文件作用，并对里面的关键代码进行解释

package web.vo;

public class News {

    private Integer id;

    private String title;

    private String content;

    private String imageurl;

    private String type;

    public Integer getId() {

        return id;

    }

    public void setId(Integer id) {

        this.id = id;

    }

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public String getContent() {

        return content;

    }

    public void setContent(String content) {

        this.content = content;

    }

    public String getImageurl() {

        return imageurl;

    }

    public void setImageurl(String imageurl) {

        this.imageurl = imageurl;

    }

    public String getType() {

        return type;

    }

    public void setType(String type) {

        this.type = type;

    }

}

## 3.3.5NewsDao.xml

描述该文件作用，并对里面的关键代码进行解释

<?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="web.dao.NewsDao">

    <resultMap id="BaseResultMap" type="web.vo.News">

        <id column="id" jdbcType="INTEGER" property="id" />

        <result column="title" jdbcType="VARCHAR" property="title" />

        <result column="content" jdbcType="VARCHAR" property="content" />

        <result column="imageurl" jdbcType="VARCHAR" property="imageurl" />

        <result column="type" jdbcType="VARCHAR" property="type" />

    </resultMap>

    <select id="selectNews" resultMap="BaseResultMap">

        select \* from `news` where `type` = #{type}

    </select>

</mapper>

## 3.3.6 application-spring.xml

描述该文件作用，并对里面的关键信息进行解释

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

<beans xmlns="http://www.springframework.org/schema/beans"

       xmlns:p="http://www.springframework.org/schema/p"

       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

       xmlns:context="http://www.springframework.org/schema/context"

       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/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">

        <!--注解扫描-->

        <context:component-scan base-package="web">

                <context:exclude-filter type="annotation"

                                        expression="org.springframework.stereotype.Controller"/>

        </context:component-scan>

        <context:property-placeholder location="classpath:db.properties"/>

        <!--连接池-->

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

                <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>

        <!--Mybatis工厂-->

        <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

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

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

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

        </bean>

        <!--配置Mapper扫描-->

        <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">

                <property name="basePackage" value="web.dao"/>

        </bean>

</beans>

## 3.3.7 mybatis-config.xml

描述该文件作用，并对里面的关键代码进行解释

<?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>

    <!--打印sql-->

    <settings>

        <setting name="logImpl" value="STDOUT\_LOGGING"/>

    </settings>

    <typeAliases>

        <!--包扫描-->

        <package name="web.vo"/>

    </typeAliases>

</configuration>

## 3.3.8 spring-mvc.xml

描述该文件作用，并对里面的关键代码进行解释

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

<beans xmlns="http://www.springframework.org/schema/beans"

       xmlns:p="http://www.springframework.org/schema/p"

       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

       xmlns:context="http://www.springframework.org/schema/context"

       xmlns:aop="http://www.springframework.org/schema/aop"

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

       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/aop

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

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

        http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd

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

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

    <!--开启注解扫描-->

    <context:component-scan base-package="web">

        <!--只扫描控制器-->

        <context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>

    </context:component-scan>

    <mvc:default-servlet-handler/>

    <mvc:annotation-driven/>

</beans>

## 3.3.9 web.xml

描述该文件作用，并对里面的关键代码进行解释

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

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"

         version="4.0">

  <listener>

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

  </listener>

  <context-param>

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

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

  </context-param>

  <!--定义Spring MVC前端控制器-->

  <servlet>

    <servlet-name>spring-mvc</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>

  </servlet>

  <!--  让springmvc的前端控制器拦截所有请求-->

  <servlet-mapping>

    <servlet-name>spring-mvc</servlet-name>

    <url-pattern>/</url-pattern>

  </servlet-mapping>

  <!--编码过滤器-->

  <filter>

    <filter-name>characterEncodingFilter</filter-name>

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

    <init-param>

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

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

    </init-param>

  </filter>

  <filter-mapping>

    <filter-name>characterEncodingFilter</filter-name>

    <url-pattern>/\*</url-pattern>

  </filter-mapping>

  <welcome-file-list>

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

  </welcome-file-list>

</web-app>

## 3.3.10 前端用户界面news.html

新闻显示界面的布局文件

<!DOCTYPE html>

<html lang="en" xmlns:v-on="http://www.w3.org/1999/xhtml">

<head>

    <meta charset="UTF-8">

    <title>Title</title>

    <link rel="stylesheet" href="./bootstrap/css/bootstrap.min.css">

    <script type="text/javascript" src="./js/jquery-3.4.1.min.js"></script>

    <script type="text/javascript" src="./bootstrap/js/bootstrap.min.js"></script>

    <script type="text/javascript" src="./js/vue.js"></script>

</head>

<style>

    #app{

        width: 820px;

        height: 400px;

    }

    #left{

        padding-top: 20px;

        padding-left: 20px;

        width: 400px;

        height: 400px;

        float: left;

    }

    li{

        margin-top: 8px;

        font-size: 18px;

    }

    .headerImage{

        width: 100px; height: 100px

    }

    .headerRight{

        float: right; height: 80px; width: 260px; margin-left: 20px

    }

    #right{

        padding-top: 20px;

        padding-left: 20px;

        width: 400px;

        height: 400px;

        float: right;

     }

</style>

<body>

    <div id="app">

        <ul class="nav nav-tabs">

            <li role="presentation" class="active"><a href="#">组织工作</a></li>

            <li role="presentation"><a href="#">干部工作</a></li>

        </ul>

        <div id="left">

            <div id="header">

                <span><img :src="data1[0].imageurl" class="headerImage"></span>

                <div class="headerRight">

                    <h5>{{data1[0].title}}...</h5>

                    <p>{{data1[0].content}}...</p>

                    <a href="#">详细</a>

                </div>

                <hr>

                <ul class="a">

                    <li v-for="data in data1">{{data.title}}</li>

                </ul>

            </div>

        </div>

        <div id="right">

            <div id="headerRight">

                <span><img :src="data2[0].imageurl" class="headerImage"></span>

                <div class="headerRight">

                    <h5>{{data2[0].title}}...</h5>

                    <p>{{data2[0].content}}...</p>

                    <a href="#">详细</a>

                </div>

                <hr>

                <ul class="a">

                    <li v-for="data in data2">{{data.title}}</li>

                </ul>

            </div>

        </div>

        <!--<button type="button"  v-on:click="test">aaa</button>-->

        <!--<h1>{{message}}</h1>-->

        <!--<h1>{{user.name}}</h1>-->

    </div>

</body>

<script type="text/javascript">

    var vue = new Vue({

        el: '#app',

        data: {

            message: 'Hello Vue!',

            user:{},

            data1:[],

            data2:[]

        },

        methods:{

            test:function () {

                $.ajax({

                    url:"www.baidu.com",

                    type:"POST",

                    data:{name : "test"},

                    success:function(result){

                        vue.data1 = result;

                    }

                });

            },

            initDate1:function () {

                $.ajax({

                    url:"/queryNews",

                    type:"GET",

                    data:{type : "1"},

                    success:function(result){

                        console.log(result);

                        vue.data1 = result;

                    }

                });

            },

            initDate2:function () {

                $.ajax({

                    url:"/queryNews",

                    type:"GET",

                    data:{type : "2"},

                    success:function(result){

                        console.log(result);

                        vue.data2 = result;

                    }

                });

            }

        },

        create:function () {

        },

        mounted:function () {

            this.initDate1();

            this.initDate2()

        }

    })

</script>

</html>

# 4 系统测试



# 5 系统总结

通过此次Web应用开发的课程设计，我对这门课有了更加深入的理解，虽然在此次课程设计中使用的都是比较基础的组件以及语法，但是通过此次课程设计对这门科目的基础知识更加熟练了。这门科目是一门以Java为基础的学科，因为最初学习Java时的基础没有打好，所以刚开始学习这门课程的时候还是有点吃力的，不过在老师和同学们的热情帮助下，逐步对Java熟练了起来，虽然也没有学的很好，但是与当初相比已经进步了很多。

在最开始的时候，学习最大了困难是在于软件，因为软件是纯英文的，所以使用起来特别的吃力，而且在软件的使用方面，有时候需要用到第三方的库，然后因为纯英文又看不懂，所以当时的心态是有点崩溃的，不过这个问题随着越来越多次的使用这个软件就解决了，通过每次上课跟着老师一起操作就逐步熟悉了软件。

当然在实际的编程过程中也总会遇到一些bug不知道如何去解决，但是老师教会了我们如何通过日志去寻找哪里出现了问题。总的来说这门科目说难也难说简单也简单，如果我们想要去深入学习这门课程，就还需要花更多的时间去探索去学习。