武汉纺织大学

Web应用开发课程设计

**新闻页面设置**

**学 院： 数学与计算机学院**

**班 级： 物联网11801**

**姓 名： 许志刚**

**学 号： 1804240515**

**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](#_Toc22456)

[2 系统设计](#_Toc21868)

[2.1用户用例图](#_Toc16712)

[2.2 ER图](#_Toc13261)

[2.3 UML类图（Class Diagram）](#_Toc12060)

[2.4 UML时序图（Sequence Diagram）](#_Toc9481)

[2.4.1 系统登录](#_Toc10858)

[2.4.2 \*\*\*\*\*\*模块](#_Toc18620)

[2.4.3 \*\*\*\*\*\*模块](#_Toc30352)

[2.5 UML活动图（Activity Diagram）](#_Toc22054)

[2.5.1 登陆](#_Toc12767)

[2.5.2 收银](#_Toc20020)

[2.5.3 \*\*\*模块](#_Toc19945)

[2.5.4 \*\*\*\*模块](#_Toc26400)

[3 系统实现](#_Toc20261)

[3.1 项目结构](#_Toc17930)

[3.2 配置文件](#_Toc14003)

[3.2.1 jdbc.properties文件](#_Toc32289)

[3.2.2 log4j.properties文件](#_Toc20299)

[3.2.3 mybatis-config.xml文件](#_Toc19038)

[3.3 VO类User.java](#_Toc7838)

[3.4 DAO接口类IUserDAO.java](#_Toc24038)

[3.5 接口映射文件UserMapper.xml](#_Toc1705)

[3.6 工具包Util](#_Toc18558)

[3.6.1 MybatisUtils.java](#_Toc16083)

[3.6.2 MD5Util.java](#_Toc6085)

[3.7 服务层UserService.java](#_Toc19150)

[3.8 用户界面Driver.java](#_Toc25352)

[3.9 \*\*\*\*\*\*.java](#_Toc1416)

[4 系统测试](#_Toc7918)

[5 系统总结](#_Toc18618)

# 1 需求分析

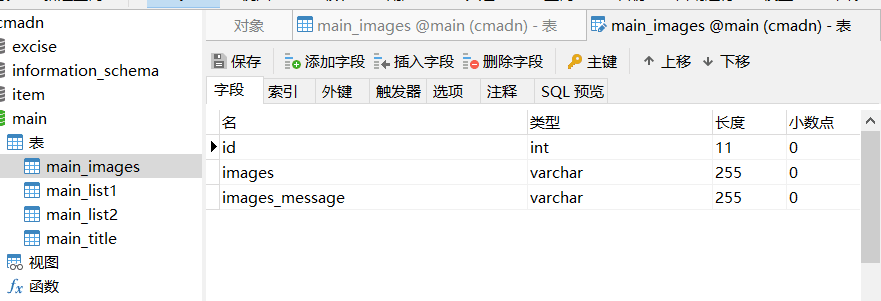


按上图设计实现新闻显示页面，页面所需图片，文字均来自数据库

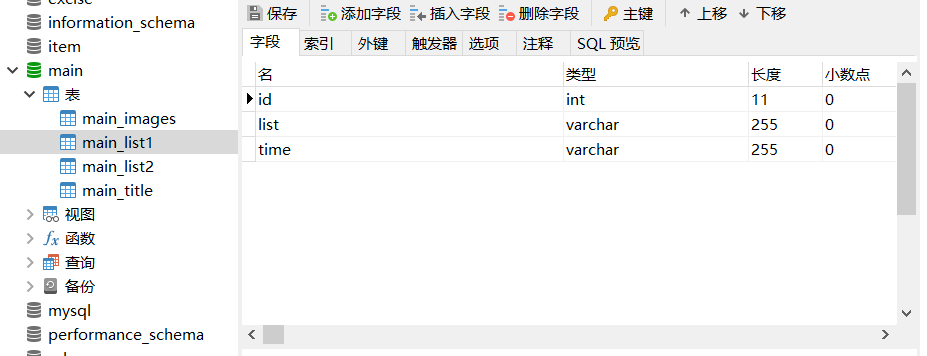
# 2 系统设计

## 2.1 ER图

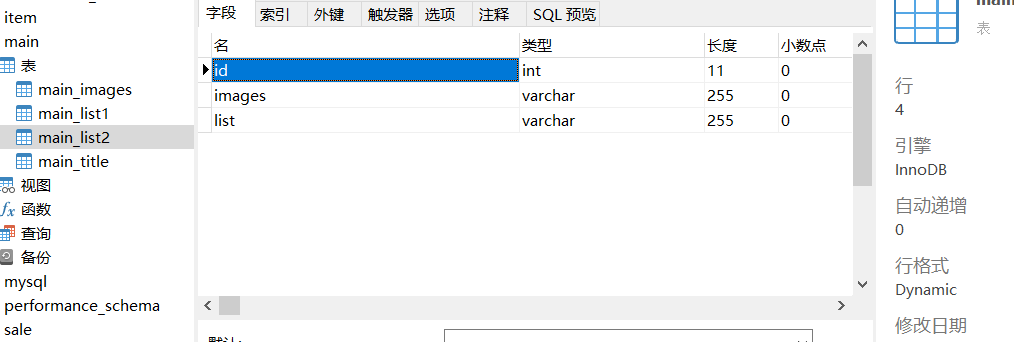
数据库设计四张表：main\_images,main\_list1,main\_list2,main\_title



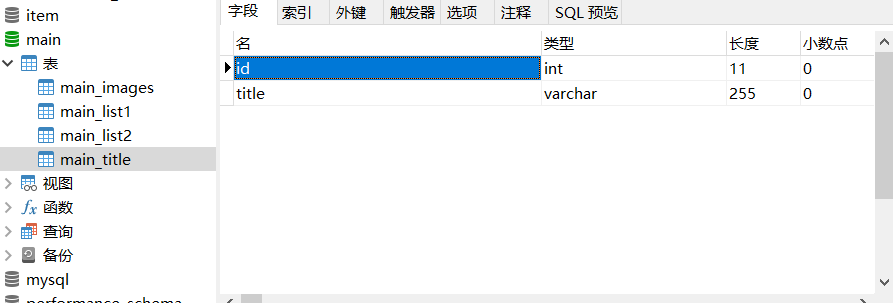
main\_images表，存放轮播图片



main\_list1表，存放左边新闻



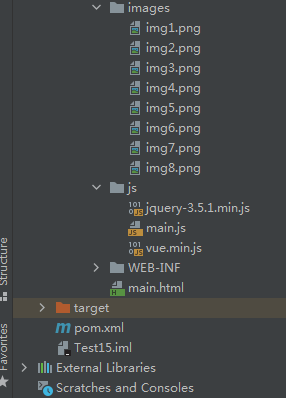
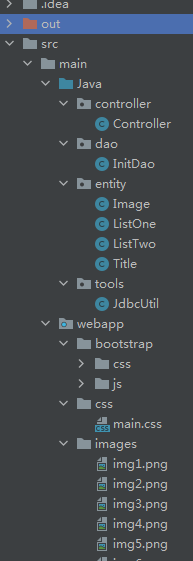
main\_list2表，存放右边新闻



Main\_title表，存放最上面头部

# 3 系统实现

## 3.1 项目结构



## 3.2 配置文件

### 3.2.1pom.xml文件

<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>com.test15</groupId>  
 <artifactId>test15</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>war</packaging>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
  
 </properties>  
  
 <dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.11</version>  
 <scope>test</scope>  
 </dependency>  
 <!--日志-->  
 <dependency>  
 <groupId>org.slf4j</groupId>  
 <artifactId>slf4j-log4j12</artifactId>  
 <version>1.7.10</version>  
 </dependency>  
 <!--J2EE-->  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.0</version>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet.jsp</groupId>  
 <artifactId>jsp-api</artifactId>  
 <version>2.2</version>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
 </dependency>  
 <!--mysql驱动包-->  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>5.1.47</version>  
 </dependency>  
 <!--springframework-->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-web</artifactId>  
 <version>5.1.0.RELEASE</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-core</artifactId>  
 <version>5.1.0.RELEASE</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.1.0.RELEASE</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.1.0.RELEASE</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-test</artifactId>  
 <version>5.1.0.RELEASE</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-jdbc</artifactId>  
 <version>5.1.0.RELEASE</version>  
 </dependency>  
 <dependency>  
 <groupId>com.github.stefanbirkner</groupId>  
 <artifactId>system-rules</artifactId>  
 <version>1.16.1</version>  
 <scope>test</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.8.9</version>  
 </dependency>  
 <!--其他需要的包-->  
 <dependency>  
 <groupId>org.apache.commons</groupId>  
 <artifactId>commons-lang3</artifactId>  
 <version>3.6</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis</artifactId>  
 <version>3.4.6</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis-spring</artifactId>  
 <version>1.3.1</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/com.alibaba/druid -->  
 <dependency>  
 <groupId>com.alibaba</groupId>  
 <artifactId>druid</artifactId>  
 <version>1.1.0</version>  
 </dependency>  
  
 <!-- https://mvnrepository.com/artifact/org.mybatis.generator/mybatis-generator-core -->  
 <dependency>  
 <groupId>org.mybatis.generator</groupId>  
 <artifactId>mybatis-generator-core</artifactId>  
 <version>1.3.6</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-databind</artifactId>  
 <version>2.9.5</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-core -->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-core</artifactId>  
 <version>2.9.5</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-annotations -->  
 <dependency>  
 <groupId>com.fasterxml.jackson.core</groupId>  
 <artifactId>jackson-annotations</artifactId>  
 <version>2.9.5</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/commons-fileupload/commons-fileupload -->  
 <dependency>  
 <groupId>commons-fileupload</groupId>  
 <artifactId>commons-fileupload</artifactId>  
 <version>1.4</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/commons-io/commons-io -->  
 <dependency>  
 <groupId>commons-io</groupId>  
 <artifactId>commons-io</artifactId>  
 <version>2.6</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/commons-logging/commons-logging -->  
 <dependency>  
 <groupId>commons-logging</groupId>  
 <artifactId>commons-logging</artifactId>  
 <version>1.1.1</version>  
 </dependency>  
  
 <!--加密依赖-->  
 <!-- https://mvnrepository.com/artifact/commons-codec/commons-codec -->  
 <dependency>  
 <groupId>commons-codec</groupId>  
 <artifactId>commons-codec</artifactId>  
 <version>1.9</version>  
 </dependency>  
 <!--mybatis分页插件-->  
 <!-- https://mvnrepository.com/artifact/com.github.jsqlparser/jsqlparser -->  
 <dependency>  
 <groupId>com.github.jsqlparser</groupId>  
 <artifactId>jsqlparser</artifactId>  
 <version>0.9.5</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/com.github.pagehelper/pagehelper -->  
 <dependency>  
 <groupId>com.github.pagehelper</groupId>  
 <artifactId>pagehelper</artifactId>  
 <version>5.0.0</version>  
 </dependency>  
  
  
 </dependencies>  
  
 <build>  
 <finalName>esquel</finalName>  
 <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->  
 <plugins>  
  
 <plugin>  
 <artifactId>maven-clean-plugin</artifactId>  
 <version>3.1.0</version>  
 </plugin>  
 <!-- see http://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_war\_packaging -->  
 <plugin>  
 <artifactId>maven-resources-plugin</artifactId>  
 <version>3.0.2</version>  
 </plugin>  
 <plugin>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.0</version>  
 </plugin>  
 <plugin>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>2.22.1</version>  
 </plugin>  
 <plugin>  
 <artifactId>maven-war-plugin</artifactId>  
 <version>3.2.2</version>  
 </plugin>  
 <plugin>  
 <artifactId>maven-install-plugin</artifactId>  
 <version>2.5.2</version>  
 </plugin>  
 <plugin>  
 <artifactId>maven-deploy-plugin</artifactId>  
 <version>2.8.2</version>  
 </plugin>  
  
 </plugins>  
 </pluginManagement>  
 </build>  
</project>

应用下载第三方库到本地

### 3.2.2web.xml

<?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">  
 <welcome-file-list>  
 <welcome-file>main.html</welcome-file>  
 </welcome-file-list>  
</web-app>

修改Tomcat启动时首页

## 3.3 js文件

### main.js

let mainData = {  
 title:[],  
 images:[],  
 listOne:[],  
 listTwo:[]  
}  
let loadNews = function () {  
 $.ajax({  
 type:"GET",  
 url:"init.do",  
 contentType : "application/x-www-form-urlencoded;charset=utf-8",  
 dataType:"json",  
 success:function (response) {  
 mainData.images = response.images;  
 mainData.title = response.title;  
 mainData.listOne = response.lists;  
 mainData.listTwo = response.activities;  
 }  
 })  
}  
new Vue({  
 el:"#mainContent",  
 data:mainData,  
 method:{  
  
 },  
 created:function () {  
 $.ajax({  
 type:"GET",  
 url:"init.do",  
 contentType : "application/x-www-form-urlencoded;charset=utf-8",  
 dataType:"json",  
 success:function (response) {  
 mainData.images = response.imaes;  
 mainData.title = response.title;  
 mainData.listOne = response.lists;  
 mainData.listTwo = response.activities;  
 }  
 })  
 }  
})前端页面与后端交互

## 3.4Controller.java

package controller;  
  
import com.fasterxml.jackson.databind.ObjectMapper;  
import dao.InitDao;  
  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
import java.io.PrintWriter;  
  
@WebServlet("/init.do")  
public class Controller extends HttpServlet {  
  
 @Override  
 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 this.doPost(request,response);  
 }  
  
 @Override  
 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
 request.setCharacterEncoding("UTF-8");  
 response.setContentType("text/html;charset = UTF-8");  
 PrintWriter out = response.getWriter();  
 ObjectMapper objectMapper = new ObjectMapper();  
  
 String json = objectMapper.writeValueAsString(new InitDao().getAll());  
 out.println(json);  
 out.flush();  
 out.close();  
 }  
}

控制器，主要接受并处理前端请求

## 3.5 InitDao.java

package dao;  
  
import entity.Image;  
import entity.ListOne;  
import entity.ListTwo;  
import entity.Title;  
import tools.JdbcUtil;  
  
import javax.swing.border.TitledBorder;  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
public class InitDao {  
 private Connection con;  
 private PreparedStatement pst;  
  
 public Map<String, Object> getAll(){  
 //连接数据库，获取省份信息  
 JdbcUtil jdbcUtil = new JdbcUtil();  
 con = jdbcUtil.getConnection();  
 Map map = new HashMap();  
  
 try {  
 map.put("title",getTitle());  
 map.put("images",getImages());  
 map.put("lists",getListOne());  
 map.put("activities",getListTwo());  
 jdbcUtil.close();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return map;  
 }  
 public List<Image> getImages() throws SQLException {  
 List<Image> list = new ArrayList<>();  
 String sql ="SELECT id,images,images\_message FROM main\_images";  
 pst = con.prepareStatement(sql);  
  
 ResultSet rs = pst.executeQuery();  
 while (rs.next()){  
 int id = rs.getInt(1);  
 String images = rs.getString(2);  
 String images\_message = rs.getString(3);  
 Image image = new Image(id,images,images\_message);  
 list.add(image);  
 }  
 return list;  
 }  
  
 public List<ListOne> getListOne() throws SQLException {  
 List<ListOne> list = new ArrayList<>();  
 String sql ="SELECT id,list,time FROM main\_list1";  
 pst = con.prepareStatement(sql);  
  
 ResultSet rs = pst.executeQuery();  
 while (rs.next()){  
 int id = rs.getInt(1);  
 String lists = rs.getString(2);  
 String images = rs.getString(3);  
 ListOne listOne = new ListOne(id,lists,images);  
 list.add(listOne);  
 }  
 return list;  
 }  
  
 public List<ListTwo> getListTwo() throws SQLException {  
 List<ListTwo> list = new ArrayList<>();  
 String sql ="SELECT id,images,list FROM main\_list2";  
 pst = con.prepareStatement(sql);  
  
 ResultSet rs = pst.executeQuery();  
 while (rs.next()){  
 int id = rs.getInt(1);  
 String images = rs.getString(2);  
 String lists = rs.getString(3);  
 ListTwo listTwo = new ListTwo(id,images,lists);  
 list.add(listTwo);  
 }  
 return list;  
 }  
  
 public List<Title> getTitle() throws SQLException {  
 List<Title> list = new ArrayList<>();  
 String sql ="SELECT id,title FROM main\_title";  
 pst = con.prepareStatement(sql);  
  
 ResultSet rs = pst.executeQuery();  
 while (rs.next()){  
 int id = rs.getInt(1);  
 String title = rs.getString(2);  
 Title title1 = new Title(id,title);  
 list.add(title1);  
 }  
 return list;  
 }  
}

根据前端页面读取数据库信息

## 3.6Image.java

package entity;  
  
public class Image {  
 private int id;  
 private String img;  
 private String img\_message;  
  
 public Image() {  
 super();  
 }  
  
 public Image(int id, String img, String img\_message) {  
 this.id = id;  
 this.img = img;  
 this.img\_message = img\_message;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getImg() {  
 return img;  
 }  
  
 public void setImg(String img) {  
 this.img = img;  
 }  
  
 public String getImg\_message() {  
 return img\_message;  
 }  
  
 public void setImg\_message(String img\_message) {  
 this.img\_message = img\_message;  
 }  
  
 @Override  
 public String toString() {  
 return "Image{" +  
 "id=" + id +  
 ", img='" + img + '\'' +  
 ", img\_message='" + img\_message + '\'' +  
 '}';  
 }  
}对应数据库中的main\_images表

## 3.10 ListOne.java

package entity;  
  
public class ListOne {  
 private int id;  
 private String list;  
 private String time;  
  
 public ListOne() {  
 super();  
 }  
  
 public ListOne(int id, String list, String time) {  
 this.id = id;  
 this.list = list;  
 this.time = time;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getList() {  
 return list;  
 }  
  
 public void setList(String list) {  
 this.list = list;  
 }  
  
 public String getTime() {  
 return time;  
 }  
  
 public void setTime(String time) {  
 this.time = time;  
 }  
  
 @Override  
 public String toString() {  
 return "listOne{" +  
 "id=" + id +  
 ", list='" + list + '\'' +  
 ", time='" + time + '\'' +  
 '}';  
 }  
}对应数据库中main\_list1表

## 3.11 ListTwo.java

package entity;  
  
public class ListTwo {  
 private int id;  
 private String images;  
 private String list;  
  
 public ListTwo() {  
 super();  
 }  
  
 public ListTwo(int id, String images, String list) {  
 this.id = id;  
 this.images = images;  
 this.list = list;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getImages() {  
 return images;  
 }  
  
 public void setImages(String images) {  
 this.images = images;  
 }  
  
 public String getList() {  
 return list;  
 }  
  
 public void setList(String list) {  
 this.list = list;  
 }  
  
 @Override  
 public String toString() {  
 return "listTwo{" +  
 "id=" + id +  
 ", images='" + images + '\'' +  
 ", list='" + list + '\'' +  
 '}';  
 }  
}对应数据库中main\_list2表

## 3.12Title.java

package entity;  
  
public class Title {  
 private int id;  
 private String title;  
  
 public Title() {  
 super();  
 }  
  
 public Title(int id, String title) {  
 this.id = id;  
 this.title = title;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 @Override  
 public String toString() {  
 return "title{" +  
 "id=" + id +  
 ", title='" + title + '\'' +  
 '}';  
 }  
}对应数据库中main\_title表

## 3.13 JdbcUtil.java

package tools;  
  
import java.io.BufferedInputStream;  
import java.io.FileInputStream;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.util.Properties;  
  
*/\*\*  
 \** ***@author*** *13281  
 \*/*public class JdbcUtil {  
 private static final String *DRIVER* = "com.mysql.jdbc.Driver";  
 private static final String *URL* = "jdbc:mysql://localhost:3306/main?useUnicode=true&characterEncoding=utf8&useSSL=false";  
 private static final String *USER* = "root";  
 private static final String *PASSWORD* = "scsf\_Y5L";  
 private Connection con;  
  
  
 static {  
 try {  
 Class.forName(*DRIVER*);  
 } catch (ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public JdbcUtil(){  
 super();  
 }  
  
 public Connection getConnection(){  
 try {  
 con = DriverManager.getConnection(*URL*,*USER*,*PASSWORD*);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return con;  
 }  
  
 public void close(){  
 if(con != null){  
 try {  
 con.close();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

连接数据库

# 4 结果显示



# 5 系统总结

本课程学习综合度较高,对于我来说学习也有些吃力.在学习过程中,也收获了以往所漏掉的内容，也学到了很多新的知识。本次课设需要做一个网页的新闻显示,需要用到数据库，也及时巩固了我有关数据库的知识。