武汉纺织大学

Web应用开发课程设计

**新闻显示3**

**学 院： 数学与计算机学院**

**班 级： 物联网11804**

**姓 名： 吴喆**

**学 号： 1804280419**

**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 3](#_Toc59206771)

[2 设计思路 3](#_Toc59206772)

[3 程序实现 4](#_Toc59206773)

[3.1 项目结构 4](#_Toc59206774)

[3.2 com.dao接口类NewsDao.java 4](#_Toc59206775)

[3.3 com.pojo实体类News.java 5](#_Toc59206776)

[3.4 com.servlet类ShowIndexServlet.java 6](#_Toc59206777)

[3.5接口映射文件web.xml 8](#_Toc59206778)

[3.6 工具包com.utils 8](#_Toc59206779)

[DbUtil.java 8](#_Toc59206780)

[4 系统测试 12](#_Toc59206781)

[4.1 访问jsp界面 12](#_Toc59206782)

[4.2 对数据库进行增加新闻条目：爱上对方斯蒂芬，删除：新闻测试 12](#_Toc59206783)

[5 系统总结 13](#_Toc59206784)

# 1 需求分析



设计一个新闻显示页面，页面所需图片，文字均来自数据库。

# 2 设计思路

1）前端传入基本的查询请求，然后通过数据业务层进行转换，底层jdbc查询数据库。

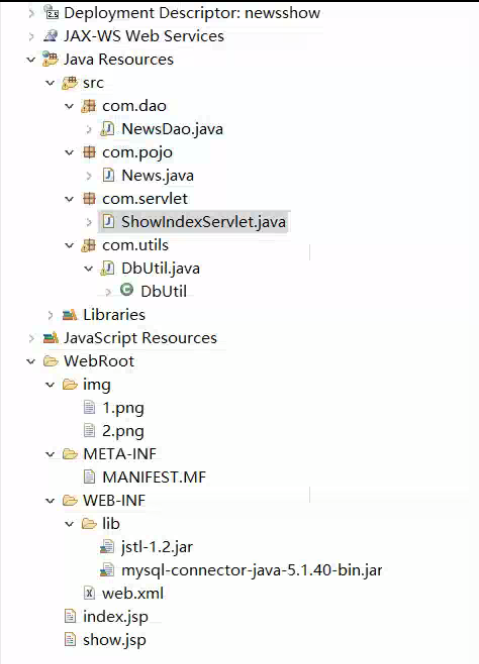
2）返回模型视图。

3）前端jsp页面通过el表达式动态设置域对象值。

4）显示列表和动态div。

# 3 程序实现

## 3.1 项目结构



## 3.2 com.dao接口类NewsDao.java

NewsDdao.java定义方法接口，获取dbUtil类news表中所有信息。

public class NewsDao {

private DbUtil dbUtil = new DbUtil();

public List<News> getAll() {

List<News> collect = dbUtil.list("select \* from news", null);

return collect;

}

}

## 3.3 com.pojo实体类News.java

News.java对要引用的数据库的5个字段进行封装。

**public** **class** News {

**private** Integer id;

**private** String title;

**private** String content;

**private** String photo;

**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 getPhoto() {

**return** photo;

}

**public** **void** setPhoto(String photo) {

**this**.photo = photo;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** News(Integer id, String title, String content, String photo, String type) {

**super**();

**this**.id = id;

**this**.title = title;

**this**.content = content;

**this**.photo = photo;

**this**.type = type;

}

**public** News() {

}

}

## 3.4 com.servlet类ShowIndexServlet.java

ShowIndexServlet.java是一个页面展示servlet，通过doGet方法将页面请求转发至show.jsp。

/\*\*

\* Servlet implementation class ShowIndexServlet

\* 页面展示servlet

\*/

**public** **class** ShowIndexServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** ShowIndexServlet() {

**super**();

// **TODO** Auto-generated constructor stub

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse

\* response)

\*/

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

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

// 设置请求字符集,以便从请求读取中文

request.setCharacterEncoding("utf-8");

// 设置响应字符集,以输出中文到浏览器正常显示

response.setContentType("text/html;charset=utf-8");

List<News> list=**new** NewsDao().getAll();

**if**(list.size()>=2)

{

request.setAttribute("first", list.get(0));

request.setAttribute("second", list.get(1));

}

**int** len=(list.size()-2)/2;

**int** sub=(list.size()-2)%2;

**if**(sub==0){

request.setAttribute("listleft", list.subList(2, len+2));

request.setAttribute("listright", list.subList(len+2, list.size()));

}**else**{

request.setAttribute("listleft", list.subList(2, len+3));

request.setAttribute("listright", list.subList(len+3, list.size()));

}

request.setAttribute("highsize", len\*78);

request.getRequestDispatcher("show.jsp").forward(request, response);

}

/\*\*

\* **@see** HttpServlet#doPost(HttpServletRequest request, HttpServletResponse

\* response)

\*/

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

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

doGet(request, response);

}

}

## 3.5接口映射文件web.xml

web.xml文件是用来初始化配置信息：Welcome页面、servlet、servlet-mapping、filter、listener、启动加载级别等。

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

<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\_2\_5.xsd"* id=*"WebApp\_ID"* version=*"2.5"*>

<display-name>newsshow</display-name>

<welcome-file-list>

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

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

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

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

<welcome-file>default.htm</welcome-file>

<welcome-file>default.jsp</welcome-file>

</welcome-file-list>

<servlet>

<description></description>

<display-name>ShowIndexServlet</display-name>

<servlet-name>ShowIndexServlet</servlet-name>

<servlet-class>com.servlet.ShowIndexServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ShowIndexServlet</servlet-name>

<url-pattern>/ShowIndexServlet</url-pattern>

</servlet-mapping>

</web-app>

## 3.6 工具包com.utils

## DbUtil.java

DbUtil.java封装了对JDBC的操作，简化了JDBC操作。

public class DbUtil {

private static final String URL = "jdbc:mysql://localhost:3306/newshow?useUnicode=true&characterEncoding=utf8";

private static final String USERNAME = "root";

private static final String PASSWORD = "123456";

public static Connection getConnection() {

Connection conn = null;

try {

Class.forName("com.mysql.jdbc.Driver");

conn = DriverManager.getConnection(URL, USERNAME, PASSWORD);

return conn;

} catch (ClassNotFoundException e) {

e.printStackTrace();

return null;

} catch (SQLException e) {

e.printStackTrace();

return null;

}

}

/\*\*

\* 增删改通用

\*

\* @param sql

\* @param params

\* @return

\*/

public static boolean update(String sql, Object... params) {

// TODO Auto-generated method stub

// 定义标记

boolean flag = false;

// 获取连接对象

Connection conn = DbUtil.getConnection();

try {

conn.setAutoCommit(false);

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

// 获取操作对象

PreparedStatement pstm = null;

try {

pstm = conn.prepareStatement(sql);

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// 给SQL语句设置值

try {

if (params != null) {

for (int i = 0; i < params.length; i++) {

pstm.setObject(i + 1, params[i]);

}

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

// 执行SQL语句

int state = 0;

try {

state = pstm.executeUpdate();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

if (state > 0) {

// 成功 提交事务

flag = true;

// 提交事务

try {

conn.commit();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

} else {

// 失败 回滚事务

flag = false;

try {

conn.rollback();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return flag;

}

public static List<News> list(String sql, Object... params) {

Connection conn = getConnection();

PreparedStatement stat = null;

List<News> list = new ArrayList<News>();

try {

stat = conn.prepareStatement(sql);

if (params != null) {

for (int i = 0; i < params.length; i++) {

// stat.setString(i, params[i]);

stat.setObject(i + 1, params[i]);

}

}

ResultSet rs = stat.executeQuery();

while (rs.next()) {

/\*\*

\* public News(Integer id, String title, String content, String

\* photo, String type) {

\*/

News stu = new News(rs.getInt("id"), rs.getString("title"), rs.getString("content"),

rs.getString("photo"), rs.getString("newstype"));

list.add(stu);

}

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally {

try {

if (stat != null)

stat.close();

if (conn != null)

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

return list;

}

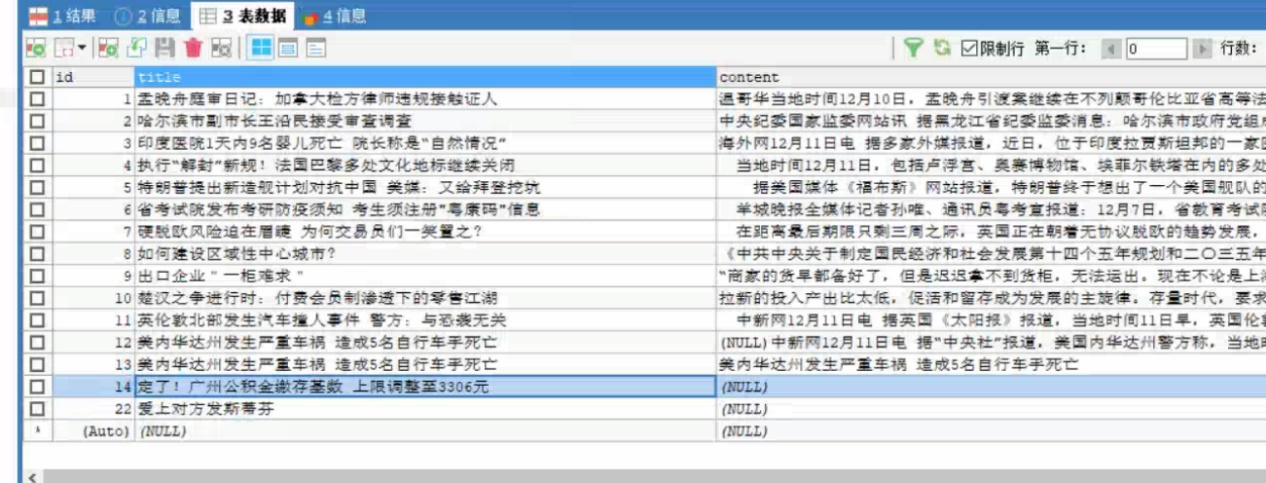
}

# 4 系统测试

## 4.1 访问jsp界面



## 4.2 对数据库进行增加新闻条目：爱上对方斯蒂芬，删除：新闻测试





# 5 系统总结

在这门课的学习中我初步了解了html，css，js，jsp，ajax，jq还有后端的ssm框架，这些都是我之前从没有接触过得东西，在学习中也感觉到了这些东西的有趣，然后在这次课设中通过实战项目加深我对这些技术的学习使我能够更加娴熟的写出代码，也通过调用网络api实现了更多的功能。

通过这次课设我认识到了数据库的重要性，数据库设计的好才能为后续的代码提供更多便利。