1.首先我们要明白打开浏览器输入项目所在的地址（如下，默认进来的就是index.jsp界面）

http://localhost:8080/myshop

**原因是看web.xml中设置了自动开始的界面就是开始界面**

2.

<style type=*"text/css"*>

.\*{font-family:*楷体*;align-text:*center*}

</style>

</head>

<body><center>

<h1>欢迎访问128书籍城</h1>

<table border=*"1"*cellspacing=*"0"*> 边框

<tr align=*"center"*>

<td>书名</td><td>价格</td><td>出版社</td><td>点击购买</td>

</tr>

<tr align=*"center"*>

<td style="width: *123px*; ">jsp入门到精通</td>

<td>20</td>

<td>128出版社</td>

<td><a href=*"#"*>点击购买</a></td>

</tr>

<tr align=*"center"*>

<td style="width: *123px*; ">java入门到精通</td>

<td>21</td>

<td>128出版社</td>

<td><a href=*"#"*>点击购买</a></td>

</tr >

<tr align=*"center"*>

<td style="width: *208px*; ">android入门到精通</td>

<td>21</td>

<td>128出版社</td>

<td><a href=*"#"*>点击购买</a></td>

</tr>

<tr>

</tr>

</table>

</center> </body>

3.现在的界面是



4.加一个购物车按钮

<tr align=*"center"*>

<td colspan=*"4"*><input type=*"button"* value=*"购物车"* style="width:*300*;font-size:*20*"/>

<td>

</tr>

5.接下来我们点击了购物车就会到了另一个界面（购物车showmyCar）

<h1>128购物车</h1>

<form method=*"post"* action=*"??"*>

<table border=*"1"*cellspacing=*"0"*width=*"611"*height=*"200"* >

<tr align=*"center"*>

<td>id</td><td>书名</td><td>价格</td><td>出版社</td><td>数量</td><td>删除</td>

</tr>

<tr align=*"center"*>

<td>3</td>

<td style="width: *143px*; ">android入门到精通</td>

<td>20</td>

<td>128出版社</td>

<td><input type=*"text"* name=*"booknum"*/></td>

<td><a href=*"#"*>删除</a></td>

</tr>

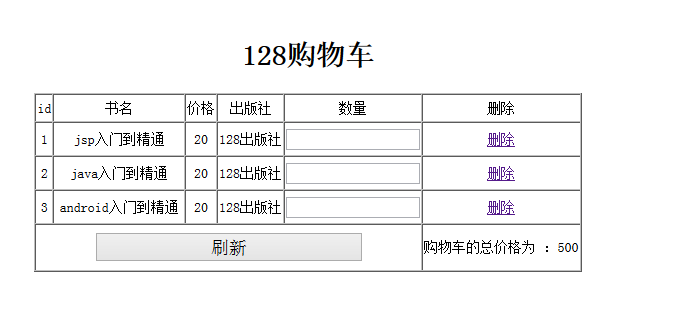
<tr align=*"center"*>

<td colspan=*"5"*><input type=*"button"* value=*"刷新"* style="width:*300*;font-size:*20*"/>

<td>购物车的总价格为 ：500</td>

</tr>

</table>

6. 

# 1.删除用户制作一个提醒功能

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

**function** ok(){

**var** pageNow= parseInt(document.getElementById("tiao").value);

**if**(pageNow!=0){

/\* \*/

// window.open("/myshop/ok.jsp?pageNow="+pageNow);

window.confirm("ha");

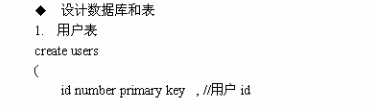
}

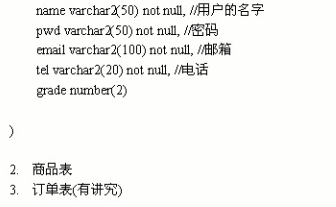
}

</script>

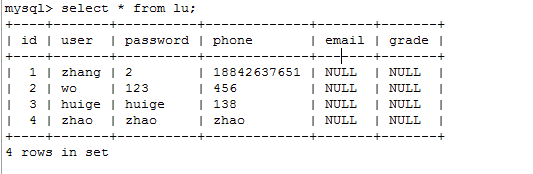
1.<a href=””+onClick = “return window.confirm(“确认删除该用户吗？”)”></a>

## 6. 数据库到了这个时候我们就要想到制作一个数据库了，让数据库与我们的购物车挂钩

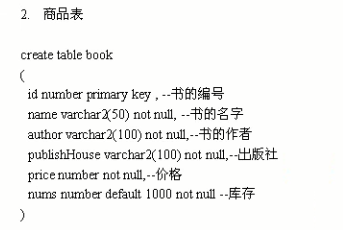


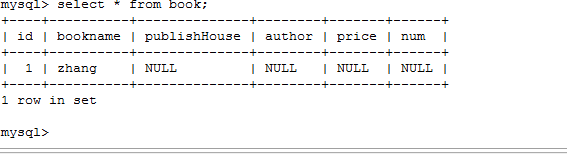


7.然后我们到了数据库中建立即可

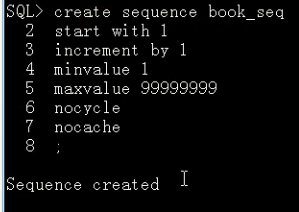


8.上面的那个是我们建立的用户的表，下面这是商品名字

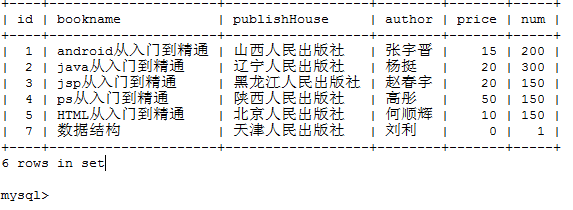




9.我们为了让上面的主键id能够自动增长我们创建下面的序列



10.然后我们建立下商品表



## 11程序到了现在我们的商品表就已经建立好了，开始写我们的程序了

1.首先我们想到的就是分页显示了，这里我们依然使用servlet中我们学习的分页的方法，甚至我们可以讲我们servlet调用进来也是可以的，因为我们的jsp是可以调用servlet的

2.我们建立一个servlet的项目，然后在里面依次得到 数据库中的总数，rowCount和能够分成几页pageCout

3.不妨使用下面的两个方法得到

第一个方法得到rowCont

**public** **class** FenYe{

**int** pageNow = 1; //显示第几页的内容

**int** pageSize = 3; //每页显示几条记录

**static** **int** *pageCount* = 0; //总过有多少页 （这个事通过计算得出的）

**int** rowCount = 0;//总共有多少条记录（就是说我们的数据库中有多少条内容）

Connection cn = **null**;

ResultSet rs = **null**;

PreparedStatement ps= **null**;

//得到我们的rowCount的数目

**public** **int** getRowCount(){

**int** rowCount = 0;

**try** {

//设计一个方法得到我们的 页数 （）

MysqlMY mysqlMY = **new** MysqlMY();

cn = mysqlMY.getConnection();

ps = cn.prepareStatement("select count(\*) from book");

rs =ps.executeQuery();

**if**(rs.next()){

rowCount = rs.getInt(1);

}

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

关闭相应的资源

**}**

2.得到我们的页数 pageCount

//得到我页数pageCount

**public** **int** getPageCount(){

FenYe fenYe = **new** FenYe();

**int** a,pageCount=0;

a = fenYe.getRowCount();

**if**(a%pageSize==0){

pageCount = a/pageSize;

}**else**{

pageCount = a/pageSize+1; }

**return** pageCount;

}

12.上面的那个还没有完成，（之后会进行讲解）

## 这里需要一个javaBean,其实就是我们在Android中学的那个person

## 创建了数据库中的user和 book，，这里讲一下user的 Javabean

**package** JavaBean;

//这时一个javaBean 和数据库中的user表对应的

**public** **class** User {

**int** id;

String user;

String email;

String phone;

@Override

**public** String toString() {

**return** "User [id=" + id + ", user=" + user + ", email=" + email

+ ", phone=" + phone + ", grade=" + grade + "]";

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getUser() {

**return** user;

}

**public** **void** setUser(String user) {

**this**.user = user;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getPhone() {

**return** phone;

}

**public** **void** setPhone(String phone) {

**this**.phone = phone;

}

**public** String getGrade() {

**return** grade;

}

**public** **void** setGrade(String grade) {

**this**.grade = grade;

}

String grade;

}

# 13.继续实现前面的那个数据库表格

1.我们首先新建一个servlet类，然后我们将需要的变量写到我们的全局变量中去

**int** pageNow = 1;

**int** pageSize = 3;

**int** rowCount = 0;

**int** pageCount = 0;

Connection cn;

PreparedStatement ps;

ResultSet rs;

MysqlMY mysqlMY ; //这个是我自己建立的一个连接类

2.然后开始一步一步实现我们的分页，首先想到的就是我们的 数据库中的记录的数目

//设计一个方法得到我们的到rowCout

mysqlMY = **new** MysqlMY();

cn = mysqlMY.getConnection();

**try** {

ps = cn.prepareStatement("select count(\*) from book");

rs =ps.executeQuery();

**if**(rs.next()){

rowCount = rs.getInt(1);

out.println(rowCount);

}

}**catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}**finally**{

**try** {

**if**(rs!=**null**){rs.close();}

**if**(ps!=**null**){ps.close();}

**if**(cn!=**null**){cn.close();}

} **catch** (Exception e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

3.然后通过计算得到我们的pageCount页 的数目

//得到页数pageCount

**if**(rowCount%pageSize==0){

pageCount = rowCount/pageSize;

}**else**{

pageCount = rowCount/pageSize+1;

}

4.到了现在我们就已经得到了所需要的 页的一些属性了。然后我们开始显示第一页的内容（这个是测试）

**try** {

cn =mysqlMY.getConnection();

ps = cn.prepareStatement("select \* from book limit "+((pageNow)\*3-3)+","+pageSize);

rs = ps.executeQuery();

//分页显示千万不能写成if

**while**(rs.next()){

out.println(" <tr>");

out.println("<td>"+rs.getInt(1)+"</td>");

out.println("<td>"+rs.getString(2)+"</td>");

out.println("<td>"+rs.getString(3)+"</td>");

out.println("<td>"+rs.getString(4)+"</td>");

out.println("<td>"+rs.getInt(5)+"</td>");

out.println("<td>"+rs.getInt(6)+"</td>");

out.println(" </tr>");

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

5.然后我们观察下界面（这里的所有的代码全部在doGet中，其他的简单的代码不写了）

## 这里就是制作上一页和下一页的东西了，

out.println("<br/>");

**if**(pageNow!=1){

out.println("<a href=FenYe?pageNow="+(pageNow-1)+">前一页<a>");

}

**for** (**int** i = 1; i <=pageCount; i++) {

out.println("<a href='FenYe?pageNow="+i+"'>"+i+"</a>");

}

**if**(pageNow!=pageCount){

out.println("<a href=FenYe?pageNow="+(pageNow+1)+">下一页<a>");

}

## 6.然后我们到了上面接收，这里特别注意的是我们这里一定要检测，但是一定要保证我们的当前的页面是小于总页数的

String now = request.getParameter("pageNow");

**if**(now!=**null**){

**int** a= Integer.parseInt(now);

**if**(a<=pageCount){

pageNow = a;

}

}

## 7.然后我们添加一个跳转的按钮

out.println("当前页："+pageNow+"总页数:"+rowCount);

out.println("跳转到 ：<input typw='text' name='tiaot' id='tiao' /><input type='button' name='tiaob' value='跳转'onClick = 'ok()'>");

8.然后我们到了javascript中去修改，下面的self就是表示可以再自己的页面打开'\_self'}

out.println("<script type='text/javascript' laguage='javascript'>");

out.println("function ok(){var "

+ "pageNow = document.getElementById('tiao')"

+ "window.alert(pageNow.value);"

+ "window.open('FenYe?pageNow='+pageNow.value+,");

out.println("</script>");

9.跳转做出来了，现在就是希望做出（上一页和下一页中间的不是固定的，它也是变化），中间的页数要求显示两页，但是不可超过 最大的页数

**for** (**int** i = pageNow+1; (i <=pageNow+2)&&(i<=pageCount); i++) {

out.println("<a href='/myshop/ok.jsp?pageNow="+i+"'>"+i+"</a>");

}

## 15 下面这个实在jsp中出现的东西我们使用下面这个来进行跳转，下面的这个强制转型和java中不一样

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

**function** ok(){

**var** a = document.getElementById("tiao").value;

**if**(a !=""){

**var** pageNow= parseInt(a);

window.open("/myshop/ok.jsp?pageNow="+pageNow);

}**else**{

window.alert("请正确输入");

}

}

**function** shan(){

window.confirm("你确定删除吗");

}

</script>

# 16.现在我们就开始制作用户的处理了，先制作一个删除用户的，和修改用户的链接，然后进行完成，这里住主要是用到了 Sqlhelper.java 和我们自己制作的关于用户的一个java文件，UserService.java

1.下面这个是我们在UserService.java文件中出现的删除的代码,其实主要是通过id来进行传播的

out.println(" <td><a href='/myshop/delete.jsp?type=del&id="+rs.getInt(1)+"' onClick = 'shan()'>删除用户</a></td>");

out.println(" <td><a href='/myshop/delete.jsp?type=update&id="+rs.getInt(1)+"'>修改用户</a></td>");

//制作一个删除记录的方法，用来删除我们的东西

**public** **boolean** DelUser(**int** id){

**boolean** a = **true**;

String sql = "delete from lu where id ="+id;

**try** { SqlHelper.*executeUpdate*(sql,**null**);

} **catch** (Exception e) {

a = **false**; } **return** a;

}

2.然后我们到了点击链接删除用户进入的界面delete.jsp界面，下面的这个我们千万不能使用Integer.parseInt(idd)来完成，下面这个得到的是一个integer的类型，这里写的这个只是一个int型的，不可以用来的到一对象

<% String type = request.getParameter("type");

String idd = request.getParameter("id");

**int** id = Integer.valueOf(idd);

String table = "lu";

out.println(type);

out.println(""+id);

UserService userService = **new** UserService();

User user ;

**if**(type.equals("del")){

**if**(userService.DelUser(id)){

response.sendRedirect("ok.jsp");

}

}

3.开始制作修改用户了，首先我们通过id进行查找有关user的所有的属性（包括密码，电话等）这个查找过程我们也是在UserService.java中完成的,下面的这个id是必须有的，我们要根据这个id来进行更新，也就是修改用户的东西

//通过得到用户来进行修改

**public** User getUser(**int** id){

User user = **null**;

ResultSet rs = **null**;

**try** {

String sql = "select \* from lu where id = ?";

String parameters[] = {""+id};

rs = SqlHelper.*executeQuery*(sql, parameters);

**if** (rs.next()) {

user = **new** User(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6));

}

} **catch** (Exception e) {

e.printStackTrace();

}**finally**{

SqlHelper.*close*(rs, SqlHelper.*getPs*(), SqlHelper.*getCt*());

}

**return** user;

}

4.下面这个是进行处理这个东西，学习使用在两个jap中一个request,通过 request.setAttribute("user", user);

**if**(type.equals("del")){

**if**(userService.DelUser(id)){

out.println("删除成功");

response.sendRedirect("ok.jsp");

}

}**else** **if**(type.equals("update")){

user = userService.getUser(id);

//将下面这个参数 和下一个界面的额request 公用一个

request.setAttribute("user", user);

//下面这个就是实现了跳转(但是这里需要知道的是下面的这个路径)

request.getRequestDispatcher("/update.jsp").forward(request, response);

}

## 5.然后我们接收面的那个user，得到它的各项属性，进，r行修改，

<body><center><br/><br/><br/><br/>

<form action=*"/myshop/updateCL.jsp"* method=*"post"*>

<%

User user = (User)request.getAttribute("user");

out.println(" <table border="+1+" cellspacing="+0+">");

out.println(" <tr align= center>");

out.println(" <td >id</td><td><input type = 'text' name = 'id' readonly value = '"+user.getId()+"'/></td>");

out.println(" </tr>");

out.println(" <tr align= center>");

out.println(" <td>姓名</td><td><input type = 'text' name = 'user' value = '"+user.getUser()+"'/></td>");

out.println(" </tr>");

out.println(" <tr align= center>");

out.println(" <td>密码</td><td><input type = 'text' name = 'password' value = '"+user.getPassword()+"'/></td>");

out.println(" </tr>");

out.println(" <tr align= center>");

out.println(" <td>手机</td><td><input type = 'text' name = 'phone' value = '"+user.getPhone()+"'/></td>");

out.println(" </tr>");

out.println(" <tr align= center>");

out.println(" <td>邮箱</td><td><input type = 'text' name = 'email' value = '"+user.getEmil()+"'/></td>");

out.println(" </tr>");

out.println(" <tr align= center>");

out.println(" <td>成绩</td><td><input type = 'text' name = 'grade' value = '"+user.getGrade()+"'/></td>");

out.println(" </tr>");

out.println(" <tr align= center>");

out.println(" <td><input type = 'submit' name = 'grade' value = '"+user.getGrade()+"'/></td><td><input type = 'text' name = 'grade' value = '"+user.getGrade()+"'/></td>");

out.println(" </tr>");

out.println(" </table>");

%>

</form>

</center> </body>

6.下面就是使用两个按钮进行修改和，重新修改

out.println(" </tr>");

out.println(" <tr align= center>");

out.println(" <td><input type = 'submit' value ='修改'/></td>

<td><input type = 'reset' value = '重新修改'/></td>");

out.println(" </tr>");

7.下面就是进行解决修改的问题了，我们让这个from表单直接到了下一个jsp文件中进行处理,通过的到

<%

String idd = request.getParameter("id");

**int** id = Integer.parseInt(idd);

String name = request.getParameter("user");

String password = request.getParameter("password");

String phone = request.getParameter("phone");

String email = request.getParameter("email");

String grade = request.getParameter("grade");

User user = **new** User(id,name,password,phone,email,grade);

UserService userservice = **new** UserService();

**if**(userservice.UpdateUser(user)){

out.println("修改成功");

out.println("<a href = '/myshop/ok.jsp'>返回管理用户</a>");

} **else**{

out.println("'修改失败");

}

8.这个就是在UserServise.java中的东西更新用户的东西

**public** **boolean** UpdateUser(User user){

**boolean** a = **true**;

String sql = "update lu set user = ?,password = ?,email = ?,phone = ?,grade = ? where id = ?";

String parameters[] = {user.getUser(), user.getPassword(),user.getEmil(),user.getPhone(),user.getGrade(),""+(user.getId())};

**try** {

SqlHelper.*executeUpdate*(sql, parameters);

} **catch** (Exception e) {

a = **false**;

}

**return** a ;

}

# 对于跳转做一个总结。

1.关于jsp之间的跳转很随意，/项目名/\*.jsp

2.对于servlet中的跳转问题 （这个是主要是看我们的web.xml中的东西）

如果web.xml中出现的是

**<url-pattern>/FenYe</url-pattern>**

response.sendRedirect("FenYe");

3.对于我们的jsp传播属性user我们一定要知道的是尽管是jsp但是也一定是这样的

request.getRequestDispatcher("/update.jsp").forward(request, response);

# 对于乱码做一个总结

1.我们修改之后，从数据库中进来东西，但是发现有乱码，这个时候我们到了显示清单的地方加上我们的、

，这个是在我们修改用户的时候我们，将一些属性设置为了中文

request.setCharacterEncoding("utf-8");

### 2.对于response我们可以使用

3.response。setContentType(“”;charset=”gbk”)