# 尚马教育 JAVA WEB课程

# 分页与查询

文档编号：B17

创建日期： 2017-06-13

最后修改日期：2019-09-23

版 本 号：V2.0

电子版文件名：尚马教育-第二阶段-7.分页技术.docx

**文档修改记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| 更新日期 | 更新作者 | 更新说明 | 版本号 |
| 2018-08-01 | 王绍成 | 初始版本 | V1.0 |
| 2019-09-13 | 郭雪岩 | 内容更新 | V2.0 |

**主讲人：**

**郭雪岩**

目录

[1. 为什么分页](#_Toc25855_WPSOffice_Level1) [3](#_Toc25855_WPSOffice_Level1)

[2. 分页实现思路](#_Toc20287_WPSOffice_Level1) [3](#_Toc20287_WPSOffice_Level1)

[3. 分页与条件查询](#_Toc14770_WPSOffice_Level1) [4](#_Toc14770_WPSOffice_Level1)

[4. Ajax分页](#_Toc15839_WPSOffice_Level1) [5](#_Toc15839_WPSOffice_Level1)

## 为什么分页

* 显示数据时，因为数据库中数据较多，一般不会一次把所有数据都查出给用户显示

所以需要通过分页的方式，把数据分开给用户显示。

## 分页实现思路

* 1.查询列表时默认显示固定数量的数据
* 2.算出当前符合条件的所有数据的总页码数
* 3.显示上一页下一页按钮
* 4.每次点击按钮是判断是否是首尾页
* 分页功能需要页面显示与数据查询同步进行。

分页的功能需要由数据库语句体现，mysql数据库中分条显示数据语句：

|  |
| --- |
| select uid,uname,upwd,unick\_name,uage,uaddr from my\_user limit 0 , 5  如果page从1开始，那么这两个值的关系 (page-1)\*pageSize , pageSize; |

在dao中 可以接收参数 来拼接查询语句。

|  |
| --- |
| public interface UserDao {  List<MyUser> queryUser(int page,int pageSize);  } |

在给用户显示页面时，需要把将要访问的页数写进页面，当用户点击翻页时，后台可以接收到数据并根据页数查找数据

|  |
| --- |
| <a class="btn" href="${pageContext.request.contextPath }/queryUser?page=${page+1 }">下一页  </a> |

判断按钮是否可用时需要知道以当前的每页显示记录数总共有多少页数据，此数据是通过总记录数和每页显示记录数算出的

|  |
| --- |
| int totalpage =  totalnum%pageSize==0?totalnum/pageSize:(totalnum/pageSize)+1;  如果总记录数能被每页显示记录数整除，那总页数直接相除得到，  如果不能整除，剩余部分需要多占一页 |

总结：在后台计算页数，并把上一页、下一页要查询的参数提前在后台生成好，生成在页面中，用户点击生成好的链接，在后台查询新的数据，同时生成新的上一页、下一页的参数。以此完成页数前端与后台的同步。

## 分页与条件查询

* 查询功能中经常有通过特定条件对记录进行筛选。

多条件时，用户如果填写有内容，则数据库语句需要拼接上此条件，如果没有时此条件不拼接。条件查询经常需要和分页功能同时存在

* 多条件查询拼接

|  |
| --- |
| List<Object> params = new ArrayList<Object>(); //存放参数列表  boolean isWhere = true; //通过开关判断是否要拼接where  if(quser.getUname()!=null&&!"".equals(quser.getUname())) {//判断条件是否存在  if(isWhere) { //判断是否拼接where  sql+=" where "+ " uname = ? "; //拼接where  isWhere = false; //把拼接where的标记改为false  }else { //否则拼接and  sql+=" and "+ " uname = ? ";  }  params.add(quser.getUname()); //把此条参数放入保存参数的list中  } |
| psta = conn.prepareStatement(sql);  int paramIndex = 1;  for(Object param: params) { //设置查询参数时 把list列表中的参数直接读出  psta.setObject(paramIndex, param);//一个个按位设置给prepareStatement对象  paramIndex++;  }  rs = psta.executeQuery(); |

注意：查总记录数的语句需要与分条查询语句查询条件相同

* 页面中翻页时，查询条件需要一起发送到后台，页面中可以利用表单中的隐藏域

|  |
| --- |
| <form id="qform" action="${pageContext.request.contextPath }/queryUser" method="post" >  <!--其他查询条件 -->  <input id="page" name="page" type="hidden">  <input id="pageSize" name="pageSize" type="hidden" name="pageSize">  <!--用隐藏域存放page pagesize -->  <input type="submit" class="btn btn-success" value="查询">  </form> |
| $("#page").val(gopage); //js操作时 给隐藏域中的属性设置值  $("#qform").submit(); //通过js触发表单提交 |

## Ajax分页

当条件查询与分页合在一起使用时，为了处理查询条件与页数同时同步，代码处理上较为麻烦，如果改成页面不刷新，只改变部分数据，逻辑结构上会更加清晰

* 页面不跳转，查询数据都保留在页面中
* 页面中通过js创建变量，保存当前翻到第几页
* 每次查询请求，将页面上的查询数据和页数发送到后台，后台返回数据和最新的页数，页面接收到数据，把数据的部分替换到页面中，并更新js中保存页数相关的变量

服务器部分进行配合ajax响应的改造，改为向页面输出json格式字符串

|  |
| --- |
| Map<String,Object> returnMap = new HashMap<String,Object>();  returnMap.put("returnCode", "10000"); //状态码（公司自己定义）  returnMap.put("returnData", lu); //数据库查询出的数据  returnMap.put("pageInfo", pif); //页数相关数据  String returnstr = JSON.toJSONString(returnMap);//转成json格式字符串  pw.print(returnstr); |

前端部分改造成ajax请求

|  |
| --- |
| $.ajax({  url:"ajaxQueryUser",//地址  type:"post",//请求类型  data:mydata, //请求的参数 键值对字符串 json格式  dataType:"json",//返回数据的类型  success:function(data){//响应成功时的回调函数  //console.log(data.returnData);  $("#mainData").html("");//清空之前的数据  //遍历返回的数据，填入到页面  $.each(data.returnData,function(i,d){  var newdata = "<tr><td>"+d.uid+"</td><td>"+d.uname+"</td>"  +"<td>"+d.age+"</td><td>"+d.uNickname+"</td>"  +"<td>"+d.uaddr+"</td></tr>"  $("#mainData").append(newdata);  })  //更新页数相关变量，更新页面显示页码  page = data.pageInfo.page;  pagesize = data.pageInfo.pageSize;  totalpage = data.pageInfo.totalPage;  //$("#myMsg").html(data);  $("#showPage").html(page);  $("#showSize").html(pagesize);  $("#showTotal").html(totalpage);  //重置按钮禁用状态  initBtn();  }  }); |