1.创建一个用于封装分页相关属性及操作的类

2.从页面增加分页导航条的功能

3.实现分页查询功能，从页面请求->Servlet->DAO的实现

可能需要的工具类，放在工具包中即可

加粗加下划线的字体即是该工具类的使用

@RequestMapping("search")

//pagenum是要查询的页数。默认是一，

public ModelAndView search(HttpServletRequest request,@RequestParam(value = "pageNum",defaultValue = "1")String pageNum){

//获得请求路径，也就是本controller的路径

String requestURI = request.getRequestURI();

//创建ModelAndView对象并确定处理完业务跳转到那个页面

ModelAndView modelAndView=new ModelAndView("customer");

//webUtils工具类中的getParametersStartingWith方法可以将参数名含有search\_的参数接收，并将

//search\_去掉后的参数名和参数值以map键值对的方式返回，例如search\_id=18,会变成id=18

Map<String, Object> search\_ = WebUtils.getParametersStartingWith(request, "search\_");

调用service层的方法，将集合search\_和要查询的页数传递。

PageInfo<Customer> pageInfo= customerService.search(search\_,pageNum);

//将返回的数据存入ModelAndView

modelAndView.addObject("pageInfo",pageInfo);

//map工具类的此方法的作用是将该集合的所有数据，再以search\_的形式拼接，用于分页之间的跳转

String queryStr = **MapUtils**.parseParamMapToQueryStr(search\_);

//将拼接好的字符串存入

modelAndView.addObject("queryStr",queryStr);

//将分页跳转的路径，也就是本页面的访问路径存入

modelAndView.addObject("requestURI",requestURI);

return modelAndView;

}

控制层还可以使用这种方式返回

@RequestMapping(value = "/page",method = RequestMethod.GET)

@ResponseBody

public ResultEntity getBXPage(HttpServletRequest request, @RequestParam(value="pageNum",defaultValue = "1") String pageNum){

String requestURI = request.getRequestURI();

//Map的key:cid，value就是传过来的值

Map<String, Object> paramMap = WebUtils.getParametersStartingWith(request, "search\_");

PageInfo<Baoxiao> pageInfo = baoXiaoService.search(paramMap,pageNum);

String queryStr = MapUtils.parseParamMapToQueryStr(paramMap);;

ResultEntity put = ResultEntity.success().put("pageInfo", pageInfo).put("queryStr", queryStr).put("requestURI", requestURI);

return put;

}

业务层的实现

@Override

public PageInfo<Customer> search(Map<String, Object> search\_, String pageNum) {

int pageNo=1;

try{

pageNo= Integer.parseInt(pageNum);

}catch (Exception ex){

ex.printStackTrace();

}

//工具类的该方法是为了直接将模糊查询的字符串拼接成%字符串%的形式，使得我们在映射文件中可以

//使用占位符，而不是拼接符

Map<String, Object> mybatisMap= **MapUtils**.parseParamMapToMyBatisMap(search\_);

//ConstantUtils.PAGE\_SIZE手写的枚举类，或者工具类，该常量表示分页一页有多少数据

//PageHelper分页插件中的类，该类的该方法是设置要查询的页数和一页有多少条数据

PageHelper.startPage(pageNo, ConstantUtils.PAGE\_SIZE);

去数据库查询数据

List<Customer> search = customerMapper.search(mybatisMap);

//将查询到的数据交给PageInfo管理，第二个参数ConstantUtils.NAVIGATE\_PAGES代表导航页有几个，

PageInfo<Customer> pageInfo =

new PageInfo<Customer>(search,ConstantUtils.NAVIGATE\_PAGES);

return pageInfo;

}

前端页面

使用c标签遍历${pageInfo.list}，从域中将存入modelandview中的数据取出来，选择它的list属性，这个属性中存储了将要显示的数据，

<c:forEach items="${pageInfo.list}" var="customer" varStatus="index">

<tr name="sign" align='center' bgcolor="#FFFFFF" onMouseMove="javascript:this.bgColor='#FCFDEE';" onMouseOut="javascript:this.bgColor='#FFFFFF';" height="22" >

<td><input name="id" type="checkbox" id="id" value="${customer.id}" class="np"></td>

<td>${index.count}</td>

<td>${customer.companyperson}</td>

<td align="center">${customer.comname}</td>

<td>

<fmt:formatDate value="${customer.addtime}" pattern="yyyy-MM-dd"/>

</td>

<td>${customer.comphone}</td>

</tr>

</c:forEach>

分页的导航条前段显示，最好将导航条写到单独的页面，因为所有导航条基本相似，只要确保参数控制层存进modelandview的属性名一致即可，到时候在导航条的位置上导入

<**jsp:include** page="page.jsp"/>

接下来是page.jsp的内容

<tr align="right" bgcolor="#EEF4EA">

<td height="36" colspan="12" align="center">

<div>

<a href="${requestURI}?pageNum=1${queryStr}">首页</a>

<a href="${requestURI}?pageNum=${pageInfo.pageNum-1}${queryStr}">上一页</a>

<c:forEach items="${pageInfo.navigatepageNums}" var="num">

<c:if test="${num == pageInfo.pageNum}">

【${num}】

</c:if>

<c:if test="${num != pageInfo.pageNum}">

<a href="${requestURI}?pageNum=${num}${queryStr}">${num}</a>

</c:if>

</c:forEach>

<a href="${requestURI}?pageNum=${pageInfo.pageNum+1}${queryStr}">下一页</a>

<a href="${requestURI}?pageNum=${pageInfo.pages}${queryStr}">末页</a>&nbsp;&nbsp;

跳转到<input id="changeNum" size="1px">页

<script type="text/javascript">

$("#changeNum").blur(function () {

var page=$("#changeNum").val();

var reg = /^[1-9]\d\*$/;

if(!reg.test(page)){

alert("请输入正确页码");

return ;

}

window.location.href="${requestURI}?pageNum="+page+"${queryStr}";

});

</script>

</div>

</td>

</tr>