后台-用户管理/文章管理

# 13.1 文章管理

## 13.1.1 前端

#### 文章列表

文章状态下拉列表直接使用固定的四个状态即可，在根据状态查询后，状态下拉框数据能够正确的回显。默认状态下显示为审核。具体代码参考如下：

显示状态下拉列表：

<div>  
  
<label>文章状态</label>  
  
<select class="form-control-sm" id="articleStatus" >  
  
<option value="-1">全部</option>  
  
<option value="0">待审核</option>  
  
<option value="1">已审核</option>  
  
<option value="2">审核未通过</option>  
  
</select>  
  
</div>

下拉列表状态回显以及为状态下拉框绑定change事件。

$(function(){  
  
// 下拉框数据改变后触发查询  
  
$(".form-control-sm").change(function(){  
  
//根据下拉框的数据查询数据  
  
$("#content-wrapper").load("/admin/manArticle?status="+$(this).val())  
  
})// end change  
  
//下拉框回显  
  
$("#articleStatus").val('${status}')  
  
})// end of function

分页的实现

后端利用分页工具类生成分页部分的html代码，在这里直接引用，代码如下一行即可  
<div>${page}</div>。这里html代码实际上是如下所示：

<ul class="pagination">  
  
 <li class="page-item">  
  
 <a class="page-link" href="javascript:void(0)" data="xxxx...xxx"  
 aria-label="Previous">  
  
 <span aria-hidden="true">n</span></a>  
  
 </li>  
  
 。。。。。。。。  
  
 </ul>  
  
<div>

代码中data存放的数据是某页的链接地址，n保存的是页码。为了让页码的超链接点击能够获取翻页数据，需要在页面加载完成后利用如下js代码实现：

//获取分页连接的URL .并进行加载  
 $('.page-link').off("click");  
$(function(){  
 $('.page-link').click(function (e) {  
 //获取点击的的url  
 var url = $(this).attr('data');  
 //在中间区域显示地址的内容  
 $('#content-wrapper').load(url);  
 });  
   
})

显示文章列表部分显示数据中日期、状态和需要需要进行转换以方便使用人员阅读；每条数据之后的链接能够打开文章详情，列表代码如下所示：

<table class="table table-sm table-hover table-bordered ">  
 <thead class="thead-light">  
 <tr align="center">  
 <td>序号</td>  
 <td>作者</td>  
 <td>标题</td>  
 <td>当前状态</td>  
 <td>发布时间</td>  
 <td>栏目</td>  
 <td>分类</td>  
 <td>操作</td>  
 </tr>  
 </thead>  
 <c:forEach items="${pageInfo.list}" var="article" varStatus="index">  
 <tr align="center">  
 <td>${index.index+1 }</td>  
 <td>${article.user.username}</td>  
 <td>${article.title}</td>  
 <td>${article.status==0?"待审核":article.status==1?"已审核":"未通过" }</td>  
 <td><fmt:formatDate value="${article.created}" pattern="yyyy年MM月dd日 HH:mm:ss"/> </td>  
 <td>${article.channel.name}</td>  
 <td>${article.cat.name}</td>  
 <td><button type="button" class="btn btn-info" onclick="toDetail(${article.id})">详情</button> </td>  
 </tr>  
  
 </c:forEach>  
  
 </table>

点击详情打开文章详情的代码：

//查看文章详情  
function toDetail(id){  
 $("#content-wrapper").load("/admin/getArticle?id="+id)  
}

#### 文章详情以及状态修改

文章详情显示直接根据后端传的数据利用c标签展示即可，在该页面的操作功能包含：通过/不通过/设置热门/设置非热门/返回。

功能按钮的代码如下所示：

<button type="button" onclick="pass(1)" class="btn btn-info">通过</button>  
<button type="button" onclick="pass(2)" class="btn btn-warning">不通过</button>  
   
<button type="button" onclick="hot(1)" class="btn btn-info">设置热门</button>  
<button type="button" onclick="hot(0)" class="btn btn-warning">设置不热门</button>  
   
<button type="button" onclick="goBack()" class="btn btn-green">返回</button>

上述代码中函数pass、hot 利用ajax技术异步请求实现。代码参考如下：

* /\*\*
  + 审核文章
    - paramter： status 1 审核通过 2 审核不通过
    - return   
      \*/  
       function pass(status){  
       //提交审核请求  
       $.post("/admin/checkArticle",{status:status,articleId:'${article.id}'},function(obj){  
       if(obj.result==1){  
       alert("处理成功")  
       $("#content-wrapper").load("/admin/manArticle")  
       }else{  
       alert(obj.errorMsg);  
       }  
       })//end post

}//end function  
  
/\*\*  
  
- 设置文章是否热门   
- paramter： status 0 非热门 1 热门  
- return   
 \*/  
 function hot(status){  
 //设置热门请求  
 $.post("/admin/sethot",{status:status,articleId:'${article.id}'},function(obj){  
 if(obj){  
 alert("操作成功!")  
 $("#content-wrapper").load("/admin/manArticle")  
 }  
 })//end post  
  
}//end function  
  
function goBack(){  
 $("#content-wrapper").load("/admin/manArticle")  
}

## 13.1.2 后端

#### 控制层

###### 获取文章列表

代码参考如下：

/\*\*

管理员审核文章列表

* @param request
* @param page 页码
* @param status 查询数据的状态
* @return  
  \*/  
  @RequestMapping("manArticle")  
  public String adminArticle(HttpServletRequest request,  
   @RequestParam(defaultValue="1") Integer page  
   ,@RequestParam(defaultValue="0") Integer status  
   ) {

// 根据状态和页码获取文章列表数据  
 PageInfo<Article> pageInfo= articelService.getAdminArticles(page,status);  
 request.setAttribute("pageInfo", pageInfo);  
 request.setAttribute("status", status);  
   
 //生成分页html代码  
 String pageStr = PageUtils.pageLoad(pageInfo.getPageNum(),pageInfo.getPages() ,   
 "/admin/manArticle?status="+status, 10);  
 request.setAttribute("page", pageStr);  
 return "admin/article/list";  
   
}

###### 审核文章

/\*\*  
 \* 审核文章  
 \* @param request  
 \* @param articleId 文章的id  
 \* @param status 审核后的状态 1 审核通过 2 不通过  
 \* @return  
 \*/  
 @RequestMapping("checkArticle")  
 @ResponseBody  
 public ResultMsg checkArticle(HttpServletRequest request,Integer articleId,int status) {

User login = (User)request.getSession().getAttribute(ConstClass.SESSION\_USER\_KEY);  
 if(login == null) {  
 return new ResultMsg(2, "对不起，您尚未登录，不能审核文章", null);  
 }  
 if(login.getRole()!= ConstClass.USER\_ROLE\_ADMIN) {  
 return new ResultMsg(3, "对不起，您没有权限审核文章", null);  
 }  
 Article article = articelService.findById(articleId);  
 if(article==null) {  
 return new ResultMsg(4, "哎呀，没有这篇文章！！", null);  
 }  
 if(article.getStatus()==status) {  
 return new ResultMsg(5, "这篇文章的状态就是您要审核的状态，无需此操作！！", null);  
 }  
 int result = articelService.updateStatus(articleId,status);  
 if(result>0) {  
 return new ResultMsg(1, "恭喜，审核成功！！", null);  
 }else {  
 return new ResultMsg(5, "很遗憾，操作失败，请与管理员联系或者稍后再试！！", null);  
 }  
}

###### 设置热门/非热门

/\*\*  
 \* 设置热门  
 \* @param request  
 \* @param articleId 文章的id  
 \* @param status 热门状态 1 审核通过 2 不通过  
 \* @return  
 \*/  
 @RequestMapping("sethot")  
 @ResponseBody  
 public ResultMsg sethot(HttpServletRequest request,Integer articleId,int status) {

User login = (User)request.getSession().getAttribute(ConstClass.SESSION\_USER\_KEY);  
 if(login == null) {  
 return new ResultMsg(2, "对不起，您尚未登录，不能修改文章热门状态", null);  
 }  
 if(login.getRole()!= ConstClass.USER\_ROLE\_ADMIN) {  
 return new ResultMsg(3, "对不起，您没有权限修改文章热门状态", null);  
 }  
 Article article = articelService.findById(articleId);  
 if(article==null) {  
 return new ResultMsg(4, "哎呀，没有这篇文章！！", null);  
 }  
 if(article.getHot() == status) {  
 return new ResultMsg(5, "这篇文章的状态就是您要修改的状态，无需此操作！！", null);  
 }  
 int result = articelService.updateHot(articleId,status);  
 if(result>0) {  
 return new ResultMsg(1, "恭喜，审核成功！！", null);  
 }else {  
 return new ResultMsg(5, "很遗憾，操作失败，请与管理员联系或者稍后再试！！", null);  
 }  
}

####

#### 服务层

##### 获取管理文章列表

/\*\*

\*

\* **\@param** page 页码

\* **\@param** status 审核的状态

\* **\@return**

\*/

\@Override

**public** PageInfo<Article> getAdminArticles(Integer page,Integer status)  
{

//设置分页信息

PageHelper.*startPage*(page, 10);

**return new** PageInfo<Article>(articleMapper.listAdmin(status));

}

##### 审核文章

/\*\*

\* 审核文章

\* **\@param** articleId

\* **\@param** status 要审核的状态

\* **\@return**

\*/

\@Override

**public int** updateStatus(Integer articleId, **int** status) {

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

**return** articleMapper.updateStatus(articleId,status);

}

##### 设置热门

/\*\*

\*

\* 修改热门

\* **\@param** articleId

\* **\@param** status

\* **\@return**

\*/

\@Override

**public int** updateHot(Integer articleId, **int** status) {

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

**return** articleMapper.updateHot(articleId,status);

}

#### 数据层

##### 获取文章列表

/\*\*

\* 获取需要管理的文章

\* **\@return**

\*/

List<Article> listAdmin(\@Param("status") Integer status);

对应sql语句

<!-- 获取需要管理的文章 -->

<select id=*"listAdmin"* resultMap=*"articleMapper"*>

SELECT id,title,picture,channel*id,category*id,user\_id,

hits,hot,status,created,updated

FROM cms\_article

WHERE deleted=0

<if test=*"status != -1 "*>

AND status=#{status}

</if>

ORDER BY created DESC

</select>

##### 审核文章

/\*\*

\* 修改文章状态

\* **\@param** articleId

\* **\@param** status

\* **\@return**

\*/

\@Update("UPDATE cms\_article set status=#{status},updated=now() "

+ " WHERE id=#{articleId}")

**int** updateStatus(\@Param("articleId") Integer articleId,

\@Param("status") **int** status);

##### 设置热门文章

/\*\*

\* 修改文章热门状态

\* **\@param** articleId

\* **\@param** status

\* **\@return**

\*/

\@Update("UPDATE cms\_article set hot=#{status},updated=now() "

+ " WHERE id=#{articleId}")

**int** updateHot(\@Param("articleId") Integer articleId, \@Param("status")  
**int** status);

# 13.2 用户管理

用户管理用于用户的查找以及注册用户账号的封禁与解禁，显示界面如下：



实现功能包含根据用户名称进行模糊查询、分页、注册账号的解禁和封禁等。

## 13.2.1 前端

#### 列表的显示

后端数据存放在request中，在jsp中渲染。代码参考如下：

<table class=*"table table-sm table-hover table-bordered "*>

<thead class=*"thead-light"*>

<tr align=*"center"*>

<td>序号</td>

<td>姓名</td>

<td>当前状态</td>

<td>性别</td>

<!-- <td>出生日期</td> -->

<td>注册日期</td>

<td>更新日期</td>

</tr>

</thead>

<c:forEach items=*"*${pageuser.list}*"* var=*"u"* varStatus=*"index"*>

<tr align=*"center"*>

<td>${index.index+1 }</td>

<td>${u.username}</td>

<td><button type=*"button"* class=*"btn btn-success"*  
onclick=*"moption('*${u.id}*',this)"*>${u.locked=="0"?"正常":"禁用"}</button></td>

<td>${u.gender==0?"女":"男"}</td>

<%-- <td>${u.birthday}</td> --%>

<td>

<fmt:formatDate value=*"*${u.createTime}*"* />

<%-- <c:if test="${u.createTime!=null}">*</c:if>* --%>

</td>

<td><fmt:formatDate value=*"*${u.updateTime}*"* /></td>

</tr>

</c:forEach>

</table>

#### 搜索

搜索输入的显示使用如下代码

<div class=*"form-inline"*>

<label for=*"username"*>用户名:</label> <input id=*"username"*  
type=*"text"*

name=*"username"* value=*"*${username }*"* class=*"form-control "*>

<button type=*"button"* class=*"btn btn-info"*  
onclick="query()">查询</button>

</div>

搜索js代码实现如下：

/\*\*

\* 根据用户名模糊查找用户

\*/

**function** query(){

//在中间区域显示地址的内容

$('#content-wrapper').load("/user/list?name="+$("[name='username']").val());

}

#### 分页

利用工具类在后台实现分页的html代码，前台直接引用。

<div>

${pageStr}

</div>

#### 工具类

参考工具类一节。

#### 封禁与解禁

解禁或封禁需要使用js发送ajax请求，处理成功刷新当前求改的数据。

Js代码参考如下：

/\*\*

\* 解禁或封禁

\* userid 用户id

\* obj dom对象

\*/

**function** moption(userid,obj){

$.ajax({

type:"post",

data:{id:userid,locked:$(obj).text()=="正常"?1:0},

url:"/user/update",

success:**function**(flag){

**if**(flag){

$(obj).text($(obj).text()=="正常"?"禁用":"正常");

}

}

})

}

## 13.2.2 后端

#### 控制层

##### 查询列表

代码参考如下

/\*\*

\*

\* **\@param** request

\* **\@param** name 模糊查询条件

\* **\@param** pageNumber 页码

\* **\@param** pageSize 每页大小

\* **\@return**

\*/

\@RequestMapping("list")

**public** String list(HttpServletRequest request,

\@RequestParam( defaultValue="") String name,

\@RequestParam(value="page",defaultValue="1") **int** pageNumber,

\@RequestParam(defaultValue="3") **int** pageSize) {

PageInfo<User> users = userService.search(pageNumber,pageSize,name);

String pageStr = PageUtil.*page*(users.getPageNum(), users.getPages(),  
"/user/list", users.getPageSize());

request.setAttribute("pageStr", pageStr);

request.setAttribute("pageuser", users);

**return** "admin/user/list";

}

##### 解禁与封禁

代码参考如下：

/\*\*

\* 修改用户的状态

\* **\@param** request

\* **\@param** id

\* **\@param** locked 是否封禁

\* **\@return**

\*/

\@RequestMapping("update")

\@ResponseBody

**public boolean** update(HttpServletRequest request,Integer id,Integer  
locked) {

**return** userService.updateLocked(id,locked)>0;

}

#### 业务层

##### 查询列表

/\*\*

\* 根据用户名模糊查询

\* **\@param** pageNumber

\* **\@param** pageSize

\* **\@param** name

\* **\@return**

\*/

PageInfo<User> search(**int** pageNumber, **int** pageSize, String name){

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

PageHelper.*startPage*(pageNumber,pageSize);

List<User> users = userMapper.queryList(name);

**return new** PageInfo<User>(users);

}

##### 解禁与封禁

/\*\*

\*

\* **\@param** id 用户的id

\* **\@param** locked 是否锁定 1 表示锁定 0 表示不锁定

\* **\@return**

\*/

**public int** updateLocked(Integer userId, Integer locked) {

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

**return** userMapper.updateLocked( userId, locked);

}

#### 数据层

##### 查询用户列表

\@Select("select id,username,password,nickname,birthday,gender,"

+ "locked,create*time as createTime,update*time as updateTime,url,"

+ "score,role from cms\_user "

+ " where username like concat('%',#{name},'%') "

+ " order by createTime desc ")

\@ResultType(User.**class**)

List<User> queryList(\@Param("name") String name);

##### 封禁与解禁用户

/\*\*

\* 修改用户的锁定状态

\* **\@param** userId

\* **\@param** locked

\* **\@return**

\*/

\@Update("update cms*user set locked=#{locked},update*time=now() WHERE  
id=#{userId}")

**int** updateLocked(\@Param("userId") Integer userId, \@Param("locked")  
Integer locked);