

失物大厅模块

需求描述：与首页过多的信息不同的是，失物大厅是展示“失物”启示列表，而且用户可以按照类别做进一步筛选，同时支持全局模糊搜索（按照内容）。

注意：寻物大厅功能基本一致，这里不再给出详细介绍。

1 服务器端接口的实现

传入参数：map 对象（搜索的条件）

返回值：map 集合

```
/**
 * 用以渲染失物大厅
 */
@RequestMapping("/queryLossList")
public Map queryLossList(@RequestBody Map<String,String>map){
    //创建map 对象 用以返回前端
    Map m = new HashMap();
    Post post = new Post();
    if(map.containsKey("typeName")){
        String typeName = map.get("typeName");
        post.setTypeName(typeName);
    }
    //失物的flag 为1
    post.setFlag(1);
    //从post 列表查询为失物的帖子
    PageInfo<Post> postPageInfo = postService.findAll(1,100,post);
    List<Post> postList = postPageInfo.getList();
    //遍历列表 放入评论数
    for (Post p:
        postList) {
        //获取id
        Integer id = p.getId();
        int nums = commentService.getCommentCounts(id);
        p.setCounts(nums);
    }

    //将结果返回前端
    m.put("postlist",postList);
    //查询分类信息 并统计分类相关的post 数量
    List<TypeVo> typeVoList = typeService.getCountsByFlag(1);
    m.put("typeVoList",typeVoList);

    return m;
}
```

```
//将结果返回前端
m.put("postlist",postList);
//查询分类信息 并统计分类相关的post数量
List<TypeVo> typeVoList = typeService.getCountsByFlag(1);
m.put("typeVoList",typeVoList);

return m;
```

注意上面三个红色框：

①TypeVo 是根据需求设计的 Vo 数据类。

```
public class TypeVo {
    private String typeName;
    private Integer counts;
```

②getCountsByFlag(int flag);查询失物启示下的分类，并统计每个分类下的启示数量。

Service 层：

```
/**
 * 失物寻物大厅下的类型数量
 * @param flag
 * @return
 */
List<TypeVo> getCountsByFlag(int flag);
```

Service 实现：

```
@Override
public List<TypeVo> getCountsByFlag(int flag) {
    return typeDao.getCountsByFlag(flag);
}
```

Dao 层：

```
/**
 * 失物寻物大厅下的类型数量
 * @param flag
 * @return
 */
List<TypeVo> getCountsByFlag(int flag);
```

Dao 层对应 xml 文件:

```
<!-- 根据flag统计数量 即统计失物下的分类数量和寻物下的分类数量-->
<select id="getCountsByFlag" resultType="com.zhanghk.vo.TypeVo" parameterType="int">
    select type_name,count(id) as counts from post where flag=#{flag} group by type_name
</select>
```

2 前端页面的处理

// 默认调用

```
getPost();

function getPost(flag,status,type,title){
    values={} //传入前端的map
    if(flag!=null){
        values.flag=flag;
        values.status=status;
    }
    if(type!=null){
        values.typeName=type;
    }
    if(title!=null){
        values.title=title;
    }

    // 发送请求
    axios.post('http://127.0.0.1:8888/post/queryLossList',values).then(function
(response) {

        var data = response.data;
        console.log(data);

        // 渲染
        var typeVoList = data.typeVoList;
        var statusDiv = document.getElementById("statusDiv");
        var statusContent = `<div class="ui labeled button m-margin-tb-tiny">
            <a href="javascript:void(0)" onclick="getPost()" class="ui basic teal
button">全部</a>
        </div>`;
        // 遍历列表
        for(let type of typeVoList){
            statusContent+=`
            <div class="ui labeled button m-margin-tb-tiny">
                <a href="javascript:void(0)"
onclick="getPost(null,null,'${type.typeName}')" class="ui basic teal
button">${type.typeName}</a>
                <div class="ui basic teal left pointing
label">${type.counts}</div>
            </div>`;
        }
        statusDiv.innerHTML = statusContent;

        // 渲染失物招领帖子
        var lossMainDiv = document.getElementById("lossMainDiv");
```

```

let lossMainContent = '';
//拿到后台返回前端的数据
var postlist = data.postlist;

//遍历 postlist
for(let obj of postlist){
  lossMainContent+=`
<div class="ui padded vertical segment m-padded-tb-large m-mobile-
lr-clear">
  <div class="ui mobile reversed stackable grid">
    <div class="eleven wide column">
      <div class="item">
        <div ${obj.status==0 && obj.flag==1 ? '' :
'style="display:none;"' } class="ui yellow label horizontal" data-tooltip="已经认
领">失物招领</div>
        <div ${obj.status==1 && obj.flag==1 ? '' :
'style="display:none;"' } class="ui blue label horizontal" data-tooltip="正在寻找">
失物招领</div>
        <i class="tags icon"></i>${obj.typeName}
        <i class="map marker icon"></i>${obj.address}
      </div>
      <h3 class="ui header">${obj.title}</h3>
      <p class="m-text">${obj.content}</p>
      <div class="ui stackable grid">
        <div class="eleven wide column">
          <div class="ui mini horizontal link list">
            <div class="item">
              <!--用
网站给的图片填充-->
              <div class="content"><a href="#"
class="header">${obj.uname}</a></div>
              </div>
              <div class="item">
                <i class="calendar icon"></i>${obj.ctime}
              </div>
              <div class="item">
                <i class="eye icon"></i>${obj.viewCount}
              </div>
              <div class="item">
                <i class="comment
icon"></i>${obj.counts}
              </div>
            </div>
          </div>
          <div class="right aligned five wide column">
            <a href="detail.html?postId=${obj.id}" class="ui
teal basic label m-padded-tiny m-text-thin">详情</a>
          </div>
        </div>
      </div>
    </div>
    <div class="five wide column">
      <a href="#" target="_blank">
         <!--用网站给的图片填充-->
      </a>
    </div>
  </div>
`
}

```

```

        </div>

        </div>`
    }
    lossMainDiv.innerHTML = lossMainContent;

    }).catch(function (error){})

}

```

特别注意下面的地方，要理解明白：

```

function getPost(flag,status,type,title){
    values={}//传入前端的map
    if(flag!=null){
        values.flag=flag;
        values.status=status;
    }
    if(type!=null){
        values.typeName=type;
    }
    if(title!=null){
        values.title=title;
    }
}

//发送请求
axios.post('http://127.0.0.1:8888/post/queryLossList',values).then(function (response) {

//渲染
var typeVolist = data.typeVolist;
var statusDiv = document.getElementById("statusDiv");
var statusContent = `<div class="ui labeled button m-margin-tb-tiny">
<a href="javascript:void(0)" onclick="getPost()" class="ui basic teal button">全部</a>
</div>`;
//遍历列表
for(let type of typeVolist){
    statusContent+=`
    <div class="ui labeled button m-margin-tb-tiny">
    <a href="javascript:void(0)" onclick="getPost(null,null,'${type.typeName}')" class="ui basic teal button">${type.typeName}</a>
    <div class="ui basic teal left pointing label">${type.counts}</div>
    </div>`;
}
statusDiv.innerHTML = statusContent;

```

①

②

```

//渲染失物招领帖子
var lossMainDiv = document.getElementById("lossMainDiv");
let lossMainContent = '';
//拿到后台返回前端的数据
var postlist = data.postlist;

//遍历postlist
for(let obj of postlist){
    lossMainContent+=`
    <div class="ui padded vertical segment m-padded-tb-large m-mobile-lr-clear">
    <div class="ui mobile reversed stackable grid">
        <div class="eleven wide column">
            <div class="item">
                <div ${obj.status==0 && obj.flag==1 ? '' : 'style="display:none;" } class="ui yellow label horizontal" data-
                <div ${obj.status==1 && obj.flag==1 ? '' : 'style="display:none;" } class="ui blue label horizontal" data-t
                <i class="tags icon"></i>${obj.typeName}
                <i class="map marker icon"></i>${obj.address}
            </div>
            <h3 class="ui header">${obj.title}</h3>
            <p class="m-text">${obj.content}</p>
            <div class="ui stackable grid">
                <div class="eleven wide column">
                    <div class="ui mini horizontal link list">
                        <div class="item">

```

在图①中，values 作为传入后台服务器端的 map 参数。

在图②中，注意 javascript:void(0)和 onclick 的含义。

图③与图②比较相似。

