

客户端发布启示

需求：用户在客户端填写启示字段后，读取对应的信息，然后发送请求到后台，后台添加启示信息后，前端刷新跳转至 index 首页面。

问题：用户在填写启示信息时，首先需要从数据库渲染出分类，这样用户才能选择一个类别进行填写。

因此接下来的首要任务就是向后台请求分类

1 后台的查类型的接口

```
@RequestMapping("/queryTypeList")
public List<Type> queryTypeList(){
    PageInfo<Type> typePageInfo = typeService.findAll(1,15,null);
    return typePageInfo.getList();
}
```

2 后台的添加启示接口

```
/**
 * 客户端发布一条启示的接口
 * 上传的图片单独处理
 * MultipartFile 这个类一般是用来接受前台传过来的文件(处理图片)
 */

@RequestMapping("/addPost")
public R addPost(@RequestParam("post") Post post,
                 @RequestParam("upload") MultipartFile upload) throws
IOException {
    // 设置路径信息
    String path = "d://images/";
    // 判断是否存在这个目录, 如果不存在则创建
    File file = new File(path);
    if (!file.exists()){
        file.mkdir();
    }

    String fileName = upload.getOriginalFilename();
    // 防止上传的图片文件名重复, 前面加个随机生成的 uuid
    fileName = UUID.randomUUID().toString().replace("-", "") + "_" + fileName;
    upload.transferTo(new File(path, fileName));

    post.setPicture("/images/" + fileName);
    post.setCtime(new Date());
    post.setViewCount(1);
    post.setStatus(1); // 默认进行中
}
```

//后期添加用户名

```
int num = postService.insert(post);
if (num!=0){
    return R.ok();
}
return R.fail("发布启示失败了！请重新发布");
}
```

注意下面几个地方

```
@RequestMapping("/addPost")
public R addPost(@RequestParam("post") Post post,
                 @RequestParam("upload") MultipartFile upload) throws IOException {
    //设置路径信息
    String path = "d://images/";
    //判断是否存在这个目录, 如果不存在则创建
    File file = new File(path);
    if (!file.exists()){
        file.mkdir();
    }

    String fileName = upload.getOriginalFilename();
    //防止上传的图片文件名重复, 前面加个随机生成的uuid
    fileName = UUID.randomUUID().toString().replace( target: "-", replacement: "")+"_"+fileName;
    upload.transferTo(new File(path,fileName));

    post.setPicture("/images/"+fileName);
    post.setCtime(new Date());
    post.setViewCount(1);
    post.setStatus(1);//默认进行中
    //后期添加用户名

    int num = postService.insert(post);
    if (num!=0){
        return R.ok();
    }
    return R.fail("发布启示失败了！请重新发布");
}
```

MultipartFile 类: https://blog.csdn.net/Peter_S/article/details/84951978

3 后前端页面的处理

首先是 form 表单.

```
<form class="ui attached segment" name="uploadForm" id="uploadForm"
method="post" enctype="multipart/form-data">
<div class="ui padded vertical segment m-padded-tb-large">
    <div class="ui mobile reversed stackable grid">
        <!-- 发布-->

        <div class="ui form" >
            <div class="field">
                <label ><span class="ui header">类型:</span></label>
```

```

<div class="inline fields">
  <div class="field">
    <div class="ui radio checkbox">
      <input type="radio" name="flag" value="0" >
      <label>寻物</label>
    </div>
  </div>
  <div class="field">
    <div class="ui radio checkbox">
      <input type="radio" name="flag" value="1" >
      <label>失物</label>
    </div>
  </div>
</div>
<div class="field" id="typeDiv">
  <script type="text/template" id="typeTmp">

    <label><span class="ui header">分类:</span></label>
    <select class="ui search dropdown" name="typeName">
      <option value="">- 请选择一个类别-</option>
      {{#data}}
      <option value="{{typeName}}">{{typeName}}</option>
      {{/data}}
    </select>
  </script>

</div>
<div class="field"></div>
</div>

<div class="ui form" style="width: 100%; height: 100%; min-
height:100px">
  <div class="required field">
    <label><span class="ui header">标题:</span></label>
    <input name="title" type="text" maxlength="20">
    <div class="ui pointing label">
      标题最多 20 字符
    </div>
  </div>
  <div class="field ">
    <label><span class="ui header">详情:</span></label>
    <div id="md-content" style="z-index: 1 !important;">
      <!-- markdown 区域 -->
      <textarea name="content" placeholder="启示详情" style="display:
none"></textarea>
    </div>
    <div class="ui pointing label">
      对物品进行简要描述
    </div>
  </div>
  <div class="field">
    <label><span class="ui header">图片:</span></label>
    <div id="imgPreview">

```

```

        <div id="prompt3">
        <span id="imgSpan" class="ui center">
        <p class="m-text-thin m-text-spaced m-opacity-mini" >点击上传
    </p>

        <i class="large upload icon"></i>
        </span>
        <input name="upload" type="file" id="file" class="filepath"
onchange="changePic(this)" accept="image/jpg,image/jpeg,image/png,image/PNG">
        <!-- 当 value 值改变时执行 changePic 函数，规定上传的文件只能是图片-
    ->

        </div>
        <img style="width: 150px;height: 150px" src="" id="img3" />
        </div>
        <div class="ui pointing label">
            图片是必须的哟
        </div>

    </div>
    <div class="field">
        <label><span class="ui header">地点:</span></label>
        <input name="address" type="text" maxlength="25"
placeholder="选填">
        <div class="ui pointing label">
            地点越准，找回的成功率就越大哟
        </div>
    </div>
    <div class="actions">
        <input type="button" onclick="addInfo()" class="ui primary
right floated m-margin-tb-large button" value="确认">
        <input type="button" class="ui right floated m-margin-tb-large
button" value="取消">
    </div>
</div>
</div>

    </form>

```

js 代码:

```

// 添加提示信息
function addInfo() {
    // 获取值信息
    // 并发送信息到后台
    // 后台新增数据后跳转到主页

    var flag = uploadForm.flag.value;
    var title = uploadForm.title.value;
    var typeName = uploadForm.typeName.value;
    var content = uploadForm.content.value;
    var upload = document.getElementById("file").files[0];
    var address = uploadForm.address.value;

```

```

var values = {};
values.flag=flag;
values.title=title;
values.typeName=typeName;
values.content=content;
values.upload=upload;
values.address=address;
//借助 formData 上传
// 将 form 表单元素的 name 与 value 进行组合，实现表单数据的序列化，从而减少表单元素的
// 的拼接，提高工作效率，异步上传文件
//使用方法参考博客 https://www.jianshu.com/p/e984c3619019
var formdata = new FormData();
formdata.append("upload",upload);//拼接图片
formdata.append("post",new
Blob([JSON.stringify(values)],{type:"application/json"}));//拼接普通值
console.log(formdata);

//添加的请求头部信息 enctype="multipart/form-data" multipart/form-data 上面表
单 上传图片的信息 这样才能添加
let config={headers:{'Content-Type':'multipart/form-data'}}

axios.post("http://127.0.0.1:8888/post/addPost",formdata,config).then(function
(response) {
    window.location.href="http://localhost:63342/swzlweb/index.html";
}).catch(function (error) {

})

}
//发送请求
axios.post("http://127.0.0.1:8888/type/queryTypeList",{ }).then(function
(response) {
    console.log(response)

    //创建 type 模板
    var tem = document.getElementById("typeTmp").innerHTML;
    var s = Mustache.render(tem,response);
    document.getElementById("typeDiv").innerHTML = s;

}).catch(function (error) {

})

/*表单验证*/
$('.ui.form').form({
    fields : {
        title : {
            identifier: 'title',
            rules: [{
                type:'empty',
                prompt:'标题：请输入标题'
            }

```

```

    }
  }
});

// 初始化 markdown 编辑器
var contentEditor;
$(function() {
  contentEditor = editormd("md-content", {
    width  : "100%",
    height : 350,
    syncScrolling : "single",
    path : "static/lib/editormd/lib/" // 路径一定要写对, 这个插件要自动寻找 static
    文件夹的
  });

  /*
    // or
    testEditor = editormd({
      id      : "test-editormd",
      width   : "90%",
      height  : 640,
      path     : "../Lib/"
    });
  */
});

function changepic() {
  $("#prompt3").css("display", "none");
  var reads = new FileReader();
  f = document.getElementById('file').files[0];
  reads.readAsDataURL(f);
  reads.onload = function(e) {
    document.getElementById('img3').src = this.result;
    $("#img3").css("display", "block");
  };
}

```