引用配置文件写好得文件保存路径

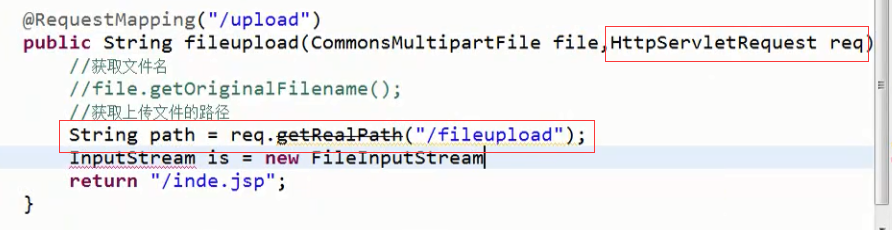
@Value("file.uploadFolder")

public String str ;

public可以换成pricate

# 获取部署项目下的路径

通过HttpSerslvetRequest获取值.



# 公司文件上传案例

index.html显示页面

index.js控制html的方法

通过class="dropdown-toggle"

控制

调用/sys/user/sessionUser

接口实现数据的上传等功能.

UserController.java

控制后台代码

uploadUserPhoto()

批量上传的数据表在t\_file中. 是一个主从表.

# 批量文件上传

参考文献

<https://www.rainss.cn/essay/1206.html>

在接受文件的参数修改为数组。

例如：

file[]

遍历数组获取文件信息

for(int i = 0; file.length; i++){

....

}

前端页面input标签的name值要相同.

IE10+以上版本可以使用

<input type="file" multiple name="upload">

上传多个文件.

JavaScript原生可以使用获取选中的文件的标题等属性

document.getElementById("file").files[1].name

其中document.getElementById("file").files 会获取一个数组，数组中的元素拥有的属性有lastModifiedDate，name，size，type，webkitRelativePath。

JQuery版修改为

$("#file")[0].files[1].name

上传文件的方式

在选择文件的时候按住ctrl就可以选择多个文件了.

file multiple多选无法删除里面已经选中的文件，会一起回发服务器. 但是可以通过js记录删除的图片下标，一起回发服务器，然后服务器遍历Files容器保存的时候对比被删除下标，对比成功则不保存这张图片.

# 公司案例

controller

|  |
| --- |
| @Value("${file.uploadFolder}") private String fileRootPath; @ResponseBody @PostMapping("/upload") public MyResult fileUpload(@RequestParam("files") MultipartFile[] files,  @RequestParam("demoDesc") String demoDesc){  String filePath = "";  int a = files.length;  String x = demoDesc;  // 多文件上传  for (MultipartFile file : files){  // 上传简单文件名  String originalFilename = file.getOriginalFilename();  // 存储路径  filePath = new StringBuilder(fileRootPath)  .append(System.*currentTimeMillis*())  .append(originalFilename)  .toString();  try {  // 保存文件  file.transferTo(new File(filePath));  } catch (IOException e) {  e.printStackTrace();  }  }  return MyResult.*ok*(); } |

html

|  |
| --- |
| <form class="form-horizontal" id="form-admin-add11" οnsubmit="return false;" enctype="multipart/form-data" method="post"> <div class="box-body">   <div class="form-group">  <label class="col-sm-2 control-label">描述</label>  <div class="col-sm-8">  <input type="text" id="demoDesc" name="demoDesc" value="" class="form-control"/>  </div>  </div>  <div class="form-group">  <label class="col-sm-2 control-label">上传</label>  <div class="col-sm-8">  <input type="file" name="files" id="files" multiple/>  </div>  </div>  <div class="form-group">  <label class="col-sm-2 control-label"></label>  <div class="col-sm-10">  <button class="btn btn-primary" id="submit" type="submit">&nbsp;保存  </button>  </div>  </div> </div> </form> |

js

|  |
| --- |
| $(function () {  $("#form-admin-add11").bootstrapValidator().on('success.form.bv', function (e) {//点击提交之后  e.preventDefault();  var $form = $(e.target);  var bv = $form.data('bootstrapValidator');  //附件数组  var crowd\_file = $('#files')[0].files;  var formData = new ***FormData***();  //附件遍历  for( var i = 0;i < crowd\_file.length; i++){  formData.append("files",$('#files')[0].files[i]);  }  //描述信息  formData.append("demoDesc", $('#demoDesc').val());   ***$***.ajax({  url: "/demo11/upload",  data: formData,  dataType: "json",  processData : false, // 使数据不做处理  contentType : false, // 不要设置Content-Type请求头  type: "POST",  success: function(data) {  if(data.code==0){  //编辑成功  layer.msg('上传成功！', {  skin: 'layer-skin-blue' //样式类名  ,closeBtn: 0  ,time:3000  ,icon:1  ,shade:0.3  ,end:function(){  layer.closeAll();  ***parent***.layer.closeAll();  }  });  }else{  //编辑失败  layer.alert("上传失败");  }  },  error:function(){  }  });  });  $(".select2").select2({  placeholder: "请选择",  width: "100%" //设置下拉框的宽度  }); }); |

# 错误

MultipartException: Current request is not a multipart request

## BootStrap fileinput.js插件

## 效果图



<**link rel="stylesheet" th:href="@{/js/fileinput/css/bootstrap.min.css}"**>  
<**link rel="stylesheet" th:href="@{/js/fileinput/css/fileinput.css}"** />

*<!--引用fileinput插件-->*<**script th:src="@{/js/fileinput/js/fileinput.js}"**></**script**>  
<**script th:src="@{/js/fileinput/js/fileinput\_locale\_zh.js}"**></**script**>  
<**script th:src="@{/js/demo11/demo22Add.js}"**></**script**>

<**form enctype="multipart/form-data" method="post"**>  
 测试:

<**input type="text" name="cs" id="cs"**>  
 <**div class="form-group"**>  
 <**input id="file-5" name="files" class="file" type="file" multiple data-preview-file-type="any" data-upload-url="#" data-preview-file-icon=""**>  
 </**div**>  
</**form**>

*$*(**"#file-5"**).fileinput({*//初始化uploadfile控件* **language**:**'zh'**,  
 **uploadUrl**:**'/demo11/upload22'**,*//初始化url参数能否重新赋值  
 // allowedFileExtensions:['txt'],//上传文件后缀格式* **overwriteInitial**: **false**,*//是否覆盖初始化* **uploadAsync**:**true**,*//默认异步上传* **showUpload**:**true**,*//是否显示上传按钮* **showRemove**:**true**,*//显示移除按钮* **showPreview**:**true**,*//是否显示预览  
 // showCaption:false,//是否显示被选文件的简介  
 // maxFileSize: 1000, //文件大小限制 单位:KB 例如:1000, 1000KB  
 // allowedFileTypes: ['image', 'video', 'flash'],//允许上传文件类型  
 // previewFileType:['image', 'html', 'text', 'video', 'audio', 'flash', 'object','other'],//预览文件类型* **browseClass**:**"btn btn-primary"**,*//按钮样式* **dropZoneEnable**:**true**,*//是否显示拖拽区域* **maxFileCount**:10,*//允许同时上传的最大文件数* **enctype**:**'multipart/form-data'**,*//上传类型* **validateInitialCount**:**true**,*//覆盖初始化初始化计数* slugCallback: **function**(filename) {

*// 显示中文*  
 **return** filename.replace(**'('**, **'\_'**).replace(**']'**, **'\_'**);  
 },  
 uploadExtraData:**function**(){*//向后台传递参数* **var** data={  
 **cs**:*$*(**"#cs"**).val()  
 };  
 **return** data;  
 },  
});

*// 隐藏fileinput插件自带的上传文件按钮  
$*(**"a[title='上传选中文件']"**).hide();

这个是隐藏那个单个上传的小按钮的

$("button[title='Upload file']").hide();

注意, 此js功能是一个独立的方法, 不用将其放到onload里.

## controller

|  |
| --- |
| @ResponseBody @PostMapping(**"/upload22"**) **public** MyResult file22Upload(@RequestParam(**"files"**) MultipartFile[] files,@RequestParam(**"cs"**) String cs){  String filePath = **""**;  **int** a = files.**length**;  String x = cs;  *// 多文件上传* **for** (MultipartFile file : files){  *// 上传简单文件名* String originalFilename = file.getOriginalFilename();  *// 存储路径* filePath = **new** StringBuilder(**fileRootPath**)  .append(System.*currentTimeMillis*())  .append(originalFilename)  .toString();  **try** {  *// 保存文件* file.transferTo(**new** File(filePath));  } **catch** (IOException e) {  e.printStackTrace();  }  }  **return** MyResult.*ok*(); } |

## 上传回调函数

同步的回调走 filebatchuploadsuccess  
异步的回调走 fileuploaded

配置项  
uploadAsync: false,//同步上传

官方案例

$('#input-id').on('filebatchuploadcomplete', function(event, files, extra) {

console.log('File batch upload complete');

});

我使用时时这样的, 所有应该添加在这个位置.

|  |
| --- |
| *$*(**"#file-5"**).fileinput({  *//初始化uploadfile控件  //配置参数* }).on(**'filebatchuploadsuccess'**, **function**(event,data,previewId,index) {*//同步上传回调  // alert("异步上传失败结果处理")* **layer**.msg(**'新增成功！'**, {  **skin**: **'layer-skin-blue'** *//样式类名* , **closeBtn**: 0  , **time**: 3000  , **icon**: 1  , **shade**: 0.3  , end: **function** () {  **layer**.closeAll();  ***parent***.**layer**.closeAll();  }  }); }); |

## 错误

### fileinput.js

Cannot read property 'fn' of undefined

你引入的文件是在jQuery以后引入的, 将JQuery.js放到fileinput.js之前.

# 参考文献

1. <https://www.iteye.com/blog/412887952-qq-com-2293385>
2. <https://blog.csdn.net/godha/article/details/12949139>