



链滴

# 关于 vditor 图片的粘体上传 / 选择上传

作者: [carlfeng](#)

原文链接: <https://ld246.com/article/1603001006832>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

## 目前情况

```
outline: true,
placeholder:"此刻，正凝聚智慧的光芒~",
upload: {
  accept: 'image/*,.mp3, .wav, .rar',
  token: 'test',
  url: '/upload/editor',
  // fieldName:"filezero",
  // linkToImgUrl: '/upload/editor',
  filename (name) {
    return name.replace(/^[^a-zA-Z0-9\u4e00-\u9fa5\.]|/g,
      replace(/[\?\\\/:|<>*\[\]\(\)\$%\{\}\@~]/g, '').
      replace('/\\s/g', ''))
  },
  linkToImgCallback(){
    console.log("api 处理")
  },
  success(editor, msg){
    console.log(msg)
    let responseData = JSON.parse(msg)
    console.log(this.filename)
    console.log(responseData.data.data.succMap)
    alert("上传成功了")
    _this.contentEditor.insertValue('Vditor ');
    return true
  },
  error(msg){
    console.log("error"+msg)
    alert("上传失败了")
  },
  // format(files, responseText){
```

这个图片相关上传搞了一个多星期了，还是没弄成。

目前，是能够是使用 url: '/upload/editor'，图片粘体/选择上传的时候都会进入到这个接口中，然后这个接口中处理图片的上传保存到云存储中，最后将图片的url地址返回。

我直接模拟返回图片数据(如图):

```
31 pass
32
33 class ImageReceive(APIView):
34
35     def post(self, request):
36         print("hallo")
37         vditor_file = request.FILES.get("file[]", None)
38         file_name = 123
39
40
41         # path_file = settings.MEDIA_ROOT
42         # print(type(vditor_file))
43
44         data = {
45             "msg": "",
46             "code": 0,
47             "data": {
48                 "errFiles": [],
49                 "succMap": {
50                     "logo.png": "https://b3logfile.com/file/2020/10/logo-6c6ccc33.png"
51                 }
52             }
53         }
54
55         return Response(data)
```

成功返回后，会调用这个回调： `success(editor: HTMLElement, msg: string): void`。

我的想法是在这个成功的回调中获取到图片链接，然后使用 `insertValue('Vditor')`；这个方法将图片接插入到编辑器中。

## 遇到的问题

1.在成功的回调中我能获取到返回的图片对象，但不知道如何获取文件名，然后取出url。

我尝试使用 `file()`方法，发现可以获取到文件名，但是却不调用 `url:/upload/image/` 这个接口了

2.我使用的是vue，查阅了相关基于vditor的开源项目，发现都没实现这个 粘体/图片上传的功能。

如果可以的话，谁能提供下完整的示例，不胜感激。