

使用 netcore 实现统一文件上传服务

作者: [loogn](#)

原文链接: <https://ld246.com/article/1561880901579>

来源网站: [链滴](#)

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

一般来讲，每个系统都有上传图片或者文件的功能，多个项目的上传文件这个功能基本上也都是差不多的，无非是限制上传的文件类型不一样，所有我们有必要开发一个统一的服务来满足公司不同项目的传文件的需求。

要满足不同项目的上传功能，大概有两种方法。第一种，各自项目前端还是要上传到对应的后端，后统一上传到文件服务，这样的好处是可以利用各个项目的验证功能，缺点是项目后端还是有少许逻辑而且从项目前端到对应后端，再从项目后端到上传服务，数据的传输多了一步。第二种，从各个项目前端直接上传的上传服务，这种方式效率高，对应的后端也没什么逻辑，但是需要解决权限验证的问题，不然别人知道了你的上传服务地址，偷偷上传就不好了。

刚开始的时候，我使用第一种方式，效果不怎么好，就着手第二中方式的实现，第二中方式首先要解的是验证的问题，还好现在有api验证的方式可以直接借鉴，所以我决定使用jwt来做权限验证。具体程是这样的：项目前端调用本项目api获取jwt，项目前端携带jwt直接把文件上传到上传服务，上传服返回上传成功的url地址。上传服务和项目后端都属于项目后端部分，这样就确保了jwt的加密是外人可见的，思路我们想清楚了，现在就要开始细节的设计了。

首先是上传服务这边的功能配置，结合appsettings.json，我们逐个说一下：

```
{
  "AllowedHosts": "*",
  "urls": "http://localhost:6001",
  "uploadServer": {
    "rootUrl": "http://localhost:6001",
    "entryPoint1": "/upload",
    "entryPoint2": "/chunkUpload",
    "virtualPath": "",
    "physicalPath": "/Users/loogn/Desktop/uploader",
    "appendMimes": ".htm3;text/html;",
    "responseCache": 604800,
    "jwtSecret": "1234561234",
    "limitSize": "20mb",
    "allowExts": ".txt;.jpg;.jpeg;.png;.doc;.docx;.xls;.xlsx;.ppt;.pptx;.pdf",
    "apps": {
      "default": {
        "allowOrigins": "",
        "enableThumbnail": true,
        "limitExts": ".exe;",
        "thumbnailExts": ".jpg;.jpeg;.png;"
      },
      "app1": {
        "allowOrigins": "*"
      }
    }
  }
}
```

- rootUrl是上传成功后文件url的基地址，如果作为静态文件服务器来提供对上传成功的文件的访问那么这这的地址应该和urls一样；如果上传成功的文件的访问由其他服务器提供（如nginx）,那么里就是对应nginx的域名；
- entryPoint1是一般上传的服务地址，完整地址是urls+entryPoint1；
- entryPoint2是分块上传的服务地址，完整地址是urls+entryPoint2，分块上传需要html5的支持，应的客户端类库是开源项目 [simple-uploader](#)；
- virtualPath:作为静态文件服务器的时候，对应的项目目录；

- physicalPath:文件上传的本地目录;
- appendMimes: 作为静态文件服务器时, 添加的mime类型映射;
- responseCache: 作为静态文件服务器时, 静态文件的缓存时间(秒);
- jwtSecret:jwt的密钥, 各个项目后端要使用相同的密钥加密获得jwt;
- limitSize:默认限制大小, 只能以mb或kb为单位结尾;
- allowExts : 默认允许上传的文件类型后缀
- apps: 该节点中是一个字典结构, key是各个项目名称, value是特定项目的特殊配置, 其中key为default的所有项目的默认配置。

我们稍后再说value中的配置项, 现在我们说一下jwt。jwt需要通过header或者form传给上传服务。对jwt不清楚的同学可以看这里jwt.io,我们约定payload中有以下四项配置:

1. exp表示过期时间, utc时间戳, 必填
2. app表示项目名称, 必填
3. size表示项目上传大小限制, 支持mb和kb为单位,选填, 可以覆盖uploadServer.limitSize
4. exts表示该项目额外支持的文件类型, 选填

由于jwt是项目后端发布的, 所以项目后端有权限和义务做好相应的配置, 比如exp过期时间, 由于jwt发布出去之后无法收回, exp就显得额外重要了, 如果上传小文件, 估计1分钟的时间就够了(前提是目后端的时间和上传服务的时间一致), 比如exts配置, 如果该项目需要上传mp4文件, 那么并不需再上传服务器上修改配置, 只需要再返回jwt的时候加入exts:".mp4"就行了。

现在再说说上传服务的配置中apps子节点的配置:

1. allowOrigins 允许上传的前台域名
2. enableThumbnail作为静态文件服务器时, 是否启用生成缩率图功能
3. limitExts **禁止**的文件后缀, 比如".exe", 如果默认配置或jwt配置允许, 这里是最后防线, 所以目可上传的后缀为 uploadServer.allowExts+jwt中exts-apps中limitExts
4. thumbnailExts 启用缩率图时, 可以应用的文件后缀, 缩率图地址使用返回的url加?w=100&h=20这样的地址访问。

项目地址: [uploadServer](#)