

📖 01.关于项目中文件上传的两种方式-本地上传

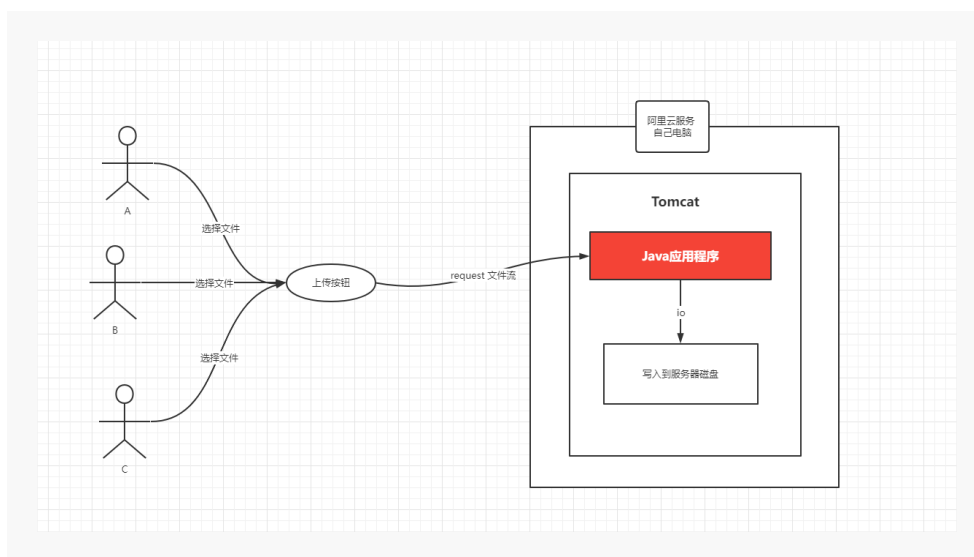
🧠 概述场景

文件上传，是程序开发中必须会使用到一个功能，比如：

- 添加商品，用户头像，文章封面等需求
- 富文本编辑（插件文件上传）

🧠 文件上传原理是什么？

为什么要实现文件上传，就要共享资源，大家都可以看你的在平台上上传的文件。就一句话：把用户的自己电脑中的文件，通过程序上传到服务器的过程！



其实就是一句话：把用户的文件通过javaio流程复制到服务器的过程，称之为文件上传。

03、使用springboot如何实现文件上传呢？

- 使用springboot完成的本地文件上传

01、实现步骤

01、搭建一个springboot工程

02、准备一个页面文件上传的页面

在resources/templates/upload.html

```
1  server:
2    port: 8777
3
4
5  spring:
6    freemarker:
7      suffix: .html
8      cache: false
9
```

03、实现后台的文件上传

定义文件上传的service

```
1 package com.kuangstudy.service;
2
3 import
  org.springframework.stereotype.Service;
4 import
  org.springframework.web.multipart.MultipartFile;
5
6 import java.io.File;
7 import java.io.IOException;
8
9 @Service
10 public class UploadService {
11
12
13     /**
14      * MultipartFile 这个对象是springMvc提供的
      文件上传的接受的类，
15      * 它的底层自动会去和HttpServletRequest
      request中的request.getInputStream()融合
16      * 从而达到文件上传的效果。也就是告诉你一个道
      理：
17      * 文件上传底层原理是：
      request.getInputStream()
18      *
19      * @param multipartFile
20      * @param dir
21      * @return
22      */
23     public String uploadImg(MultipartFile
      multipartFile, String dir) {
24         // 1: 指定文件上传的目录
25         File targetFile = new
      File("F://tmp/" + dir);
```

```

26         try {
27
28             if(!targetFile.exists())targetFile.mkdirs()
29             ;
30             // 2: 指定文件上传以后的目录
31             File targetFileName = new
32             File(targetFile,"aaa.png");
33             // 2: 文件上传到指定的目录
34
35             multipartFile.transferTo(targetFileName);
36             return "ok";
37         } catch (IOException e) {
38             e.printStackTrace();
39             return "fail";
40         }
41     }
42 }

```

定义一个文件上传的controller

```

1 package com.kuangstudy.controller;
2
3 import com.kuangstudy.service.UploadService;
4 import
5     org.springframework.beans.factory.annotation
6     .Autowired;
7 import
8     org.springframework.stereotype.Controller;
9 import
10    org.springframework.web.bind.annotation.Post
11    Mapping;
12 import
13    org.springframework.web.bind.annotation.Requ
14    estParam;

```

```
8  import
   org.springframework.web.bind.annotation.ResponseBody;
9  import
   org.springframework.web.multipart.MultipartFile;
10
11  import javax.servlet.http.HttpServlet;
12  import
   javax.servlet.http.HttpServletRequest;
13
14  @Controller
15  public class UploadController {
16
17
18      @Autowired
19      private UploadService uploadService;
20
21      /**
22       * 文件上传具体实现
23       * @param multipartFile
24       * @param request
25       * @return
26       */
27      @PostMapping("/upload/file")
28      @ResponseBody
29      public String
   upload(@RequestParam("file") MultipartFile
   multipartFile, HttpServletRequest request){
30          if(multipartFile.isEmpty()){
31              return "文件有误!!!";
32          }
33          long size = multipartFile.getSize();
34          String originalFilename =
   multipartFile.getOriginalFilename();
35          String contentType =
   multipartFile.getContentType();//image/png;image/jpg;image/gif
```

```

36 //
    if(!contentType.equals("png|jpg|gif")){ //伪
        代码 ， 不正确的代码
37 //          return "文件类型不正确";
38 //      }
39 // 1: 获取用户指定的文件夹。问这个文件夹为
        什么要从页面上传过来呢？
40 // 原因是：做隔离，不同业务，不同文件放在不
        同的目录中
41     String dir =
        request.getParameter("dir");
42     return
        uploadService.uploadImg(multipartFile,dir);
43 }
44
45
46 }
47

```

定义一个页面

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport"
6         content="width=device-width, user-
        scalable=no, initial-scale=1.0, maximum-
        scale=1.0, minimum-scale=1.0">
7     <meta http-equiv="X-UA-Compatible"
        content="ie=edge">
8     <title>文件上传</title>
9 </head>
10 <body>
11     <h1>文件上传</h1>
12
13     <form action="/upload/file"
        enctype="multipart/form-data" method="post">

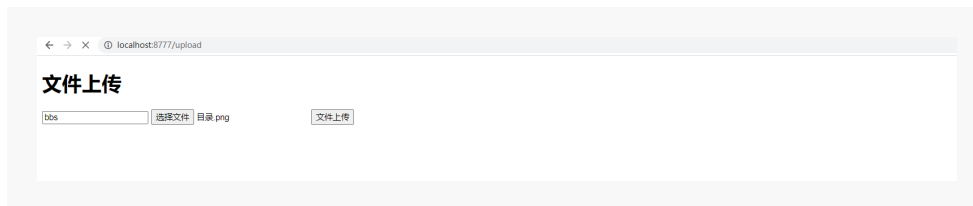
```

```

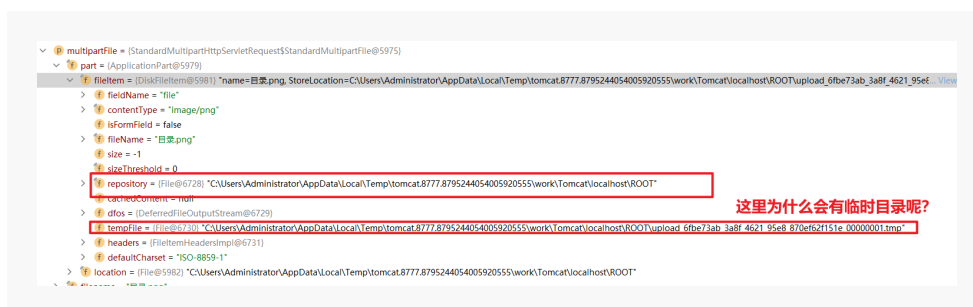
14         <input name="dir" value="bbs">
15         <input name="file" type="file">
16         <input type="submit" value="文件上
    传">
17     </form>
18
19 </body>
20 </html>

```

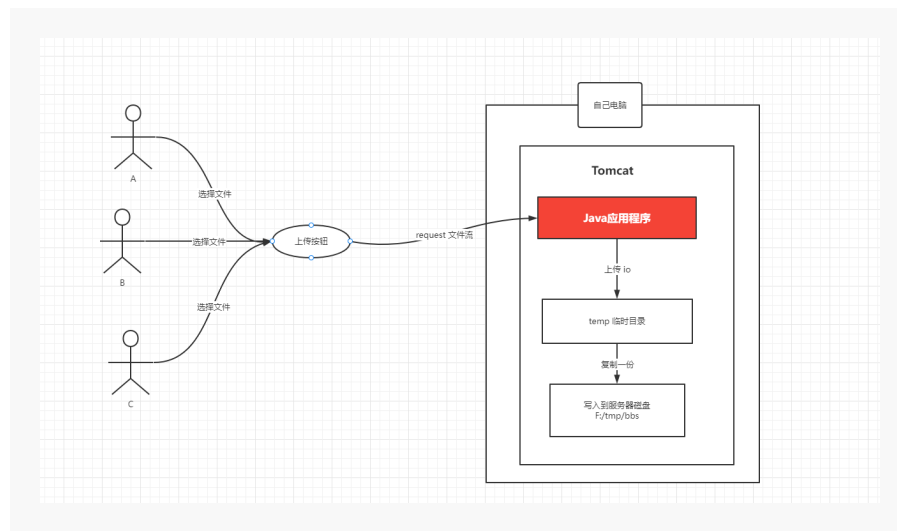
点击文件上传，然后在后台看到如下代码：



springmvc中文件上传的提供了包装对象：MultipartFile 原理如下：



- 通过MultipartFile 我们很清楚的看到，文件已经被服务器接收。
- 但是会产生一个临时目录，这个目录是可以去配置
-



文件上传不会直接上传真是的目录，它一定要经过一个临时目录的中转以后，才会上传到真是目录。作用：

- 防止上传出现网络断开，或者用户上传直接刷新或者取消。因为如果用户的网络断开或者取消，就造成大量的垃圾文件
- 保证真实目录上传的文件一定是有效的。
- 文件上传的大小：也可以进行配置的
- 文件上传的类型：也是进行配置的

```
1 server:
2     port: 8777
3
4
5 spring:
6     freemarker:
7         suffix: .html
8         cache: false
9     servlet:
10        multipart:
11            enabled: true
12            # 是单个文件大小 默认1M 10KB
13            max-file-size: 2MB
14            # 是设置总上传的数据大小
15            max-request-size: 10MB
16            #当文件达到多少时进行磁盘写入
17            file-size-threshold: 20MB
18            # 设置临时目录
19            location: F://data//tempxxxxxxxxxxxxx
```


20
21
22

04、指定文件上传的目录

配置springboot静态资源存储服务，将上传的文件放入指定的目录中

```
1 package com.kuangstudy.service;
2
3 import
  org.springframework.beans.factory.annotation
    .Value;
4 import
  org.springframework.stereotype.Service;
5 import
  org.springframework.web.multipart.MultipartF
    ile;
6
7 import
  javax.servlet.http.HttpServletRequest;
8 import java.io.File;
9 import java.io.IOException;
10 import java.text.SimpleDateFormat;
11 import java.util.Date;
12 import java.util.UUID;
13
14 @Service
15 public class UploadService {
16
17
18     @Value("${file.uploadFolder}")
19     private String uploadFolder;
20     @Value("${file.staticPath}")
```

```

21     private String staticPath;
22
23     /**
24      * MultipartFile 这个对象是springMvc提供的
    文件上传的接受的类，
25      * 它的底层会自动去和HttpServletRequest
    request中的request.getInputStream()融合
26      * 从而达到文件上传的效果。也就是告诉你一个道
    理：
27      * 文件上传底层原理是：
    request.getInputStream()
28      *
29      * @param multipartFile
30      * @param dir
31      * @return
32      */
33     public String uploadImg(MultipartFile
    multipartFile, String dir) {
34         try {
35             String realfilename =
    multipartFile.getOriginalFilename(); // 上传
    的文件：aaa.jpg
36             // 2:截图文件名的后缀
37             String imgSuffix =
    realfilename.substring(realfilename.lastInde
    xof(".")); // 拿到：.jpg
38             // 3:生成的唯一的文件名：能不能用中文
    名：不能因为统一用英文命名。
39             String newFileName =
    UUID.randomUUID().toString()+imgSuffix; // 将
    aaa.jpg改写成：SD23423k324-23423ms.jpg
40             // 4: 日期目录
41             SimpleDateFormat dateFormat =
    new SimpleDateFormat("yyyy/MM/dd");
42             String datePath =
    dateFormat.format(new Date()); // 日期目录：
    2021/10/27
43             // 5: 指定文件上传以后的目录

```

```
44         String servrepath =
uploadFolder;// 这不是tomcat服务目录，别人不认识
45         File targetPath = new
File(servrepath+dir,datePath);// 生成一个最终
目录:F://tmp/avatar/2021/10/27
46
        if(!targetPath.exists())targetPath.mkdirs()
; // 如果目录不存在: F://tmp/avatar/2021/10/27
递归创建
47         // 6: 指定文件上传以后的服务器的完整的
文件名
48         File targetFileName = new
File(targetPath,newFileName);// 文件上传以后在
服务器上最终文件名和目录是:
F://tmp/avatar/2021/10/27/SD23423k324-
23423ms.jpg
49         // 7: 文件上传到指定的目录
50
        multipartFile.transferTo(targetFileName);//
将用户选择的aaa.jpg上传到
F://tmp/avatar/2021/10/27/SD23423k324-
23423ms.jpg
51         // 可访问的路径要返回页面
52         //
http://localhost:8777/bbs/2021/10/27/5f61dea
2-4b77-4068-8d0b-fdf415eac6df.png
53         String filename =
dir+"/"+datePath+"/"+newFileName;
54         return staticPath+"/"+filename;
55     } catch (IOException e) {
56         e.printStackTrace();
57         return "fail";
58     }
59 }
60
61 }
62
```

05、通过http请求服务资源

springboot如果去指定任意目录作为的资源访问目录？

springboot有一个目录：**static**这个目录其实就是静态资源目录，这个目录下面的文件是可以通过**http**直接访问的。但是程序一般打成**jar**包，我们没办法去文件写入到这个**static**下，所以springboot提供静态资源目录的额外的映射机制，就是静态资源服务映射。它就类似于：**nginx**的静态资源映射。

01、配置静态资源服务的映射

```
1 package com.kuangstudy.config;
2
3 import
  org.springframework.beans.factory.annotation
    .Value;
4 import
  org.springframework.context.annotation.Confi
    guration;
5 import
  org.springframework.web.servlet.config.annot
    ation.ResourceHandlerRegistry;
6 import
  org.springframework.web.servlet.config.annot
    ation.WebMvcConfigurer;
7
8 @Configuration
9 public class WebMvcConfiguration implements
  WebMvcConfigurer {
10
```

```

11
12     @Value("${file.staticPatternPath}")
13     private String staticPatternPath;
14     @Value("${file.uploadFolder}")
15     private String uploadFolder;
16
17     // 这个就是springboot中springMvc让程序开发者
    去配置文件上传的额外的静态资源服务的配置
18     @Override
19     public void
    addResourceHandlers(ResourceHandlerRegistry
    registry) {
20
    registry.addResourceHandler(staticPatternPa
    th).addResourceLocations("file:"+uploadFolde
    r);
21     }
22 }
23

```

核心代码分析

```

1 registry.addResourceHandler("访问的路
    径").addResourceLocations("上传资源的路径");
2 registry.addResourceHandler("/uploadimg").add
    ResourceLocations("file:F://tmp");

```

这个时候你把：aaa.png文件上传到了F://tmp 下，那么：你可以通过：<http://localhost:8777/uploadimg/aaa.png>

环境隔离配置

application-dev.yml

```
1 # 本机配置
2 file:
3   staticPath: http://localhost:8777
4   staticPatternPath: /upimages/**
5   uploadFolder: F:/tmp/
```

application-prod.yml

```
1 # 服务器配置
2 file:
3   staticPath: https://www.kuangstudy.com
4   staticPatternPath: /upimages/**
5   uploadFolder: /www/upload/
```

application.yml激活环境

```
1 server:
2   port: 8777
3
4
5 spring:
6   freemarker:
7     suffix: .html
8     cache: false
9   profiles:
10    active: dev
11   servlet:
12     multipart:
13       enabled: true
14       # 是单个文件大小 默认1M 10KB
15       max-file-size: 2MB
16       # 是设置总上传的数据大小
17       max-request-size: 10MB
18       #当文件达到多少时进行磁盘写入
19       file-size-threshold: 20MB
20       # 设置临时目录
21       location: F://data//tempxxxxxxxxxxxxx
```

22

23

24

在浏览器访问：

<http://localhost:8777/upimages/bbs/2021/10/27/20560d68-c785-465c-bf2e-02e0de37e9ab.png>



06、对文件上传的思考和优化和控制

- 比如文件的格式
- 比如文件的大小
- 比如文件的重命名
- 比如文件的目录分类

