



链滴

# SpringBoot 之文件批量上传和下载

作者: [hjljy](#)

原文链接: <https://ld246.com/article/1581242384633>

来源网站: [链滴](#)

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



## springboot 文件的批量上传和批量下载

之前学习并使用过文件的单个上传：[springboot学习日志之DAY08文件上传功能](#)

最近写程序过程当中需要使用到批量上传和批量下载，批量和单个文件的上传下载还是有很多的区别。特此记录

### 文件批量上传

HTML代码：

```
<input type="file" name="md" id="article_md" multiple="multiple" onchange="uploadFile()>
```

注意添加multiple="multiple"属性，这样弹出文件选择框之后就可以选择多个文件。

JS 后台代码：

```
function uploadFile() {  
    var files = $("#article_md")[0].files;  
    var formData = new FormData();  
    for (let file of files) {  
        //将文件属性放入formdata里面，不能直接使用 formData.append("files",files)  
        formData.append("files",file,file.name);  
    }  
    $.ajax({  
        url: '/import',  
        type: 'post',  
        async: false,  
        data: formData,  
        processData: false,// 告诉jQuery不要去处理发送的数据
```

```

contentType: false,// 告诉jQuery不要去设置Content-Type请求头
beforeSend: function (){//过程...
    console.log('正在进行, 请稍候')
},
success: function (res) {
    if (res.code == 0) {
        console.log('上传成功')
    } else {
        console.log('上传失败')
    }
},
error: function () {
    console.log('上传失败')
}
})
}

```

JAVA后台代码:

```

@PostMapping("/import")
public AjaxResult importMd(@RequestParam("files") MultipartFile[] files, HttpServletRequest request) {
    //files 和formdata里面的属性名保持同名原则
    for (MultipartFile file : files) {
        // 文件处理
    }
    return AjaxResult.SUCCESS();
}

```

## 文件的批量下载

其实批量下载和单个下载都是一样的, 区别在于, 批量下载是将要下载的所有文件打包成一个压缩文, 然后下载这个压缩文件。

前端代码:

```

<a href="#" onclick="exportMd()" ></a>

function exportMd(){
    window.location.href = "/export"
}

```

后端代码:

```

@GetMapping("/export")
public void exportedMd(HttpServletRequest response) {
    List<MlogArticlesEntity> list = mlogArticlesService.list();
    //生成zip文件存放位置
    long timeMillis = System.currentTimeMillis();
    String strZipPath = "D:/markdown/" + timeMillis + ".zip";
    File file = new File("D:/markdown/");
    //文件存放位置目录不存在就创建
    if (!file.isDirectory() && !file.exists()) {
        file.mkdirs();
    }
}

```

```

}
try {
//通过response的outputStream输出文件
ServletOutputStream outputStream = response.getOutputStream();
ZipOutputStream out = new ZipOutputStream(new FileOutputStream(strZipPath));
for (int i = 0; i < list.size(); i++) {
    out.putNextEntry(new ZipEntry("文件名称"));
    int len;
    // 读入需要下载的文件的内容，打包到zip文件
    out.write("文件内容");
    out.closeEntry();
}
out.close();
BufferedInputStream bis = new BufferedInputStream(new FileInputStream(strZipPath))

//将输入流的数据拷贝到输入流输出
FileCopyUtils.copy(bis, outputStream);

} catch (IOException e) {
    e.printStackTrace();
}
//删除文件或者文件夹下所有文件
removeDir(file);
}

private void removeDir(File dir) {
    File[] files=dir.listFiles();
    for(File file:files){
        if(file.isDirectory()){
            removeDir(file);
        }else{
            file.delete();
        }
    }
}
}
}

```

总结：很简单的一个功能，但还是记录一下，好久没有写上传下载的代码，再次接触，有些生疏了啊