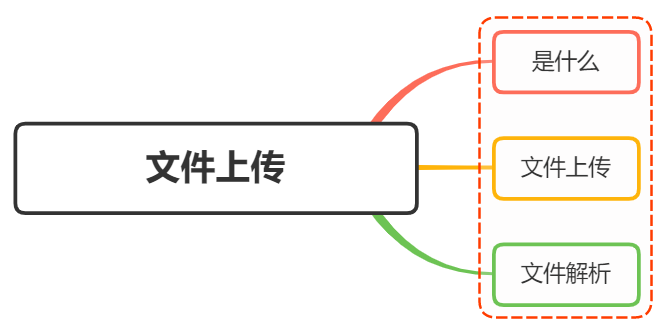
# 面试官：如何实现文件上传？说说你的思路



## 一、是什么

文件上传在日常开发中应用很广泛，我们发微博、发微信朋友圈都会用到了图片上传功能

因为浏览器限制，浏览器不能直接操作文件系统的，需要通过浏览器所暴露出来的统一接口，由用户主动授权发起来访问文件动作，然后读取文件内容进指定内存里，最后执行提交请求操作，将内存里的文件内容数据上传到服务端，服务端解析前端传来的数据信息后存入文件里

对于文件上传，我们需要设置请求头为content-type:multipart/form-data

multipart互联网上的混合资源，就是资源由多种元素组成，form-data表示可以使用HTML Forms 和 POST 方法上传文件

结构如下：

POST /t2/upload.do HTTP/1.1  
User-Agent: SOHUWapRebot  
Accept-Language: zh-cn,zh;q=0.5  
Accept-Charset: GBK,utf-8;q=0.7,\*;q=0.7  
Connection: keep-alive  
Content-Length: 60408  
Content-Type:multipart/form-data; boundary=ZnGpDtePMx0KrHh\_G0X99Yef9r8JZsRJSXC  
Host: w.sohu.com  
  
--ZnGpDtePMx0KrHh\_G0X99Yef9r8JZsRJSXC  
Content-Disposition: form-data; name="city"  
  
Santa colo  
--ZnGpDtePMx0KrHh\_G0X99Yef9r8JZsRJSXC  
Content-Disposition: form-data;name="desc"  
Content-Type: text/plain; charset=UTF-8  
Content-Transfer-Encoding: 8bit  
   
...  
--ZnGpDtePMx0KrHh\_G0X99Yef9r8JZsRJSXC  
Content-Disposition: form-data;name="pic"; filename="photo.jpg"  
Content-Type: application/octet-stream  
Content-Transfer-Encoding: binary  
   
... binary data of the jpg ...  
--ZnGpDtePMx0KrHh\_G0X99Yef9r8JZsRJSXC--

boundary表示分隔符，如果要上传多个表单项，就要使用boundary分割，每个表单项由———XXX开始，以———XXX结尾

而xxx是即时生成的字符串，用以确保整个分隔符不会在文件或表单项的内容中出现

每个表单项必须包含一个 Content-Disposition 头，其他的头信息则为可选项， 比如 Content-Type

Content-Disposition 包含了 type和 一个名字为name的 parameter，type 是 form-data，name参数的值则为表单控件（也即 field）的名字，如果是文件，那么还有一个 filename参数，值就是文件名

Content-Disposition: form-data; name="user"; filename="logo.png"

至于使用multipart/form-data，是因为文件是以二进制的形式存在，其作用是专门用于传输大型二进制数据，效率高

### 二、如何实现

关于文件的上传的上传，我们可以分成两步骤：

* 文件的上传
* 文件的解析

### 文件上传

传统前端文件上传的表单结构如下：

<form action="http://localhost:8080/api/upload" method="post" enctype="multipart/form-data">  
 <input type="file" name="file" id="file" value="" multiple="multiple" />  
 <input type="submit" value="提交"/>  
</form>

action 就是我们的提交到的接口，enctype="multipart/form-data" 就是指定上传文件格式，input 的 name 属性一定要等于file

### 文件解析

在服务器中，这里采用koa2中间件的形式解析上传的文件数据，分别有下面两种形式：

* koa-body
* koa-multer

#### koa-body

安装依赖

npm install koa-body

引入koa-body中间件

const koaBody = require('koa-body');  
app.use(koaBody({  
 multipart: true,  
 formidable: {  
 maxFileSize: 200\*1024\*1024 // 设置上传文件大小最大限制，默认2M  
 }  
}));

获取上传的文件

const file = ctx.request.files.file; // 获取上传文件

获取文件数据后，可以通过fs模块将文件保存到指定目录

router.post('/uploadfile', async (ctx, next) => {  
 // 上传单个文件  
 const file = ctx.request.files.file; // 获取上传文件  
 // 创建可读流  
 const reader = fs.createReadStream(file.path);  
 let filePath = path.join(\_\_dirname, 'public/upload/') + `/${file.name}`;  
 // 创建可写流  
 const upStream = fs.createWriteStream(filePath);  
 // 可读流通过管道写入可写流  
 reader.pipe(upStream);  
 return ctx.body = "上传成功！";  
});

#### koa-multer

安装依赖：

npm install koa-multer

使用 multer 中间件实现文件上传

const storage = multer.diskStorage({  
 destination: (req, file, cb) => {  
 cb(null, "./upload/")  
 },  
 filename: (req, file, cb) => {  
 cb(null, Date.now() + path.extname(file.originalname))  
 }  
})  
  
const upload = multer({  
 storage  
});  
  
const fileRouter = new Router();  
  
fileRouter.post("/upload", upload.single('file'), (ctx, next) => {  
 console.log(ctx.req.file); // 获取文件  
})  
  
app.use(fileRouter.routes());

## 参考文献

* https://segmentfault.com/a/1190000037411957
* https://www.jianshu.com/p/29e38bcc8a1d