断点续传

## 什么是断点续传

断点续传就是文件上传到一部分，因为断网或者用户不想上传取消操作，或者页面刷新导致文件不能上传，下一次上传同一个文件时，可以从上一次用户已经上传的部分接着上传，而不会重新上传

## 断点续传的作用

1. 防止已经上传的文件再一次重新上传
2. 可以传输较大的文件
3. 节省流量，节省时间

## 三．实现的原理

前端h5 blob（包含只读文件原始数据的类文件对象，可以看中是存放二进制数据的容器，可以用js直接操作的二进制数据）通过slice切片，后端通过concat-files进行切片合并

主要代码摘要：

H5



获取文件



文件切片



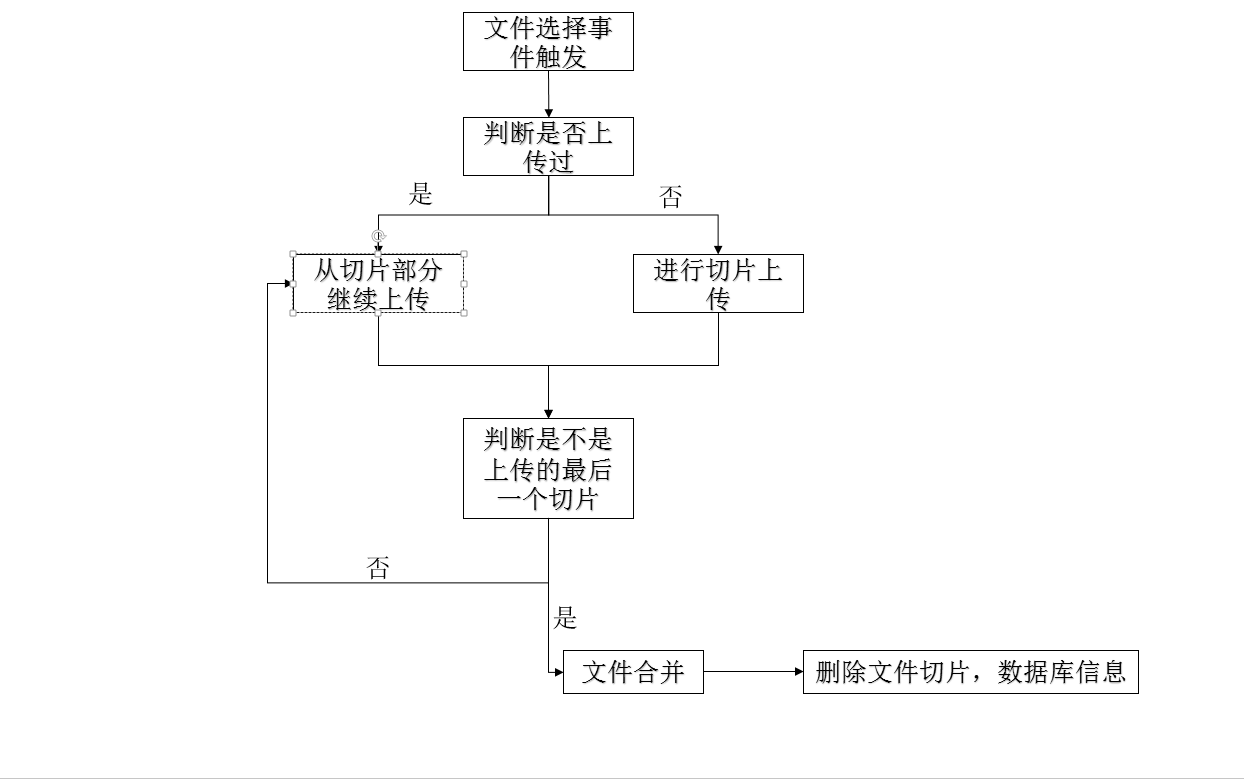
文件合并

**var** concat = *require*(**'concat-files'**);

concat([“http://localhost:8988/1.jpg”],” [http://localhost:8988/comm.jpg”,function(err){})](http://localhost:8988/comm.jpg”,function(err)%7b%7d))

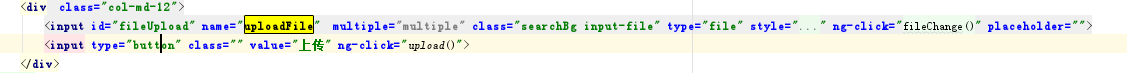
(concat参数 1 要合并的文件，2 合并后生成指定的文件 3. 合并完成 )

## 四．断点续传思路实现图



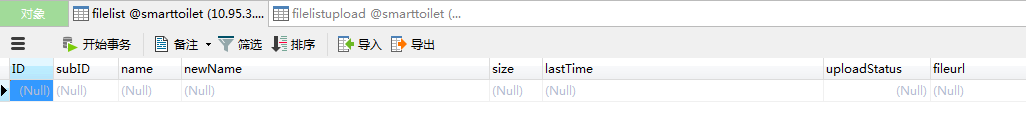
## 五．详细代码图

### 1.HTML5

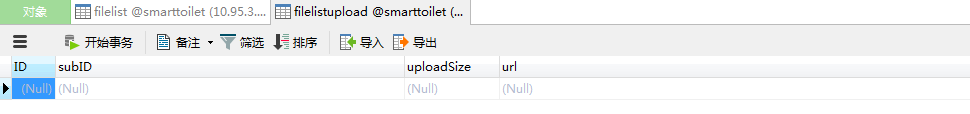


### 2.库表结构

Filelist 存放文件信息 subID(文件与切片联系ID) name(文件原名称) newName(文件新名称) size(文件总大小) lastTime(文件最后修改时间) uploadStatus (文件是否上传完成1 完成 0 未完成 fileurl 文件外网访问地址)



Filelistupload存放文件切片信息 subID(文件与切片联系ID) uploadSize(切片的结束位置) url(服务器本地文件绝对路径)



### 3.js



接口

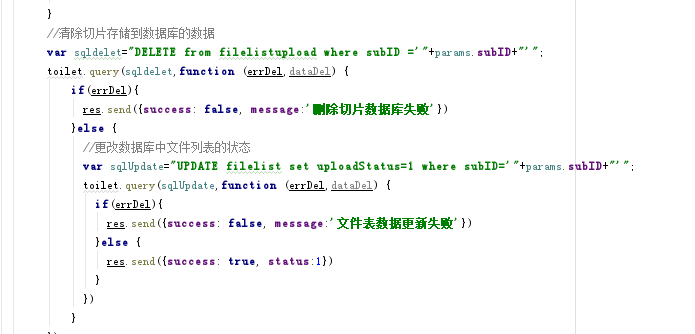


主要做查询功能,如果上传过了，直接alert,没有，文件表里面插一条数据



接口





负责切片的插入，插入完成后切片数据文件删除，文件状态修改