每个文件上传之前都会走 /zone/upload/md5Check 这个接口校验文件是否已经存在数据库记录中，此时checkType 参数值为1

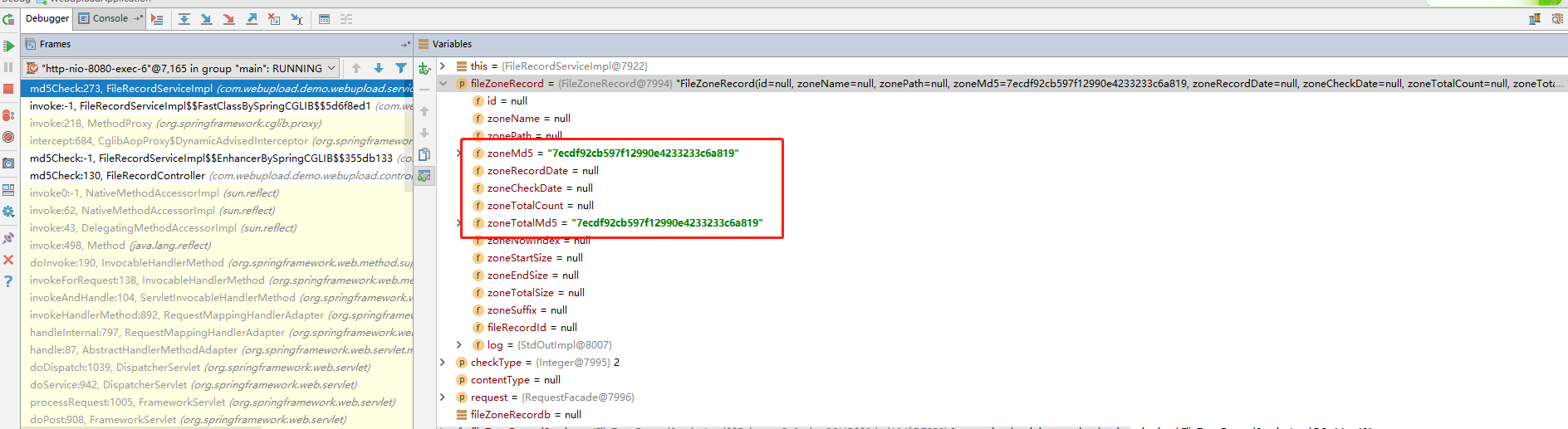
是根据文件的MD5值（md5\_value字段）和upload\_type两个字段去数据库的sys\_file\_record表里面查

第一种情况：如果没有查到数据记录，说明这个文件之前没上传过，则返回结果集 {code:'9999',success:false,msg:'请选择文件上传'}

前台代码收到这个返回值后会执行layWebupload.js 的第321行代码逻辑 继续上传文件 ，然后还是走的 /zone/upload/md5Check 这个接口，此时

checkType 参数值为2

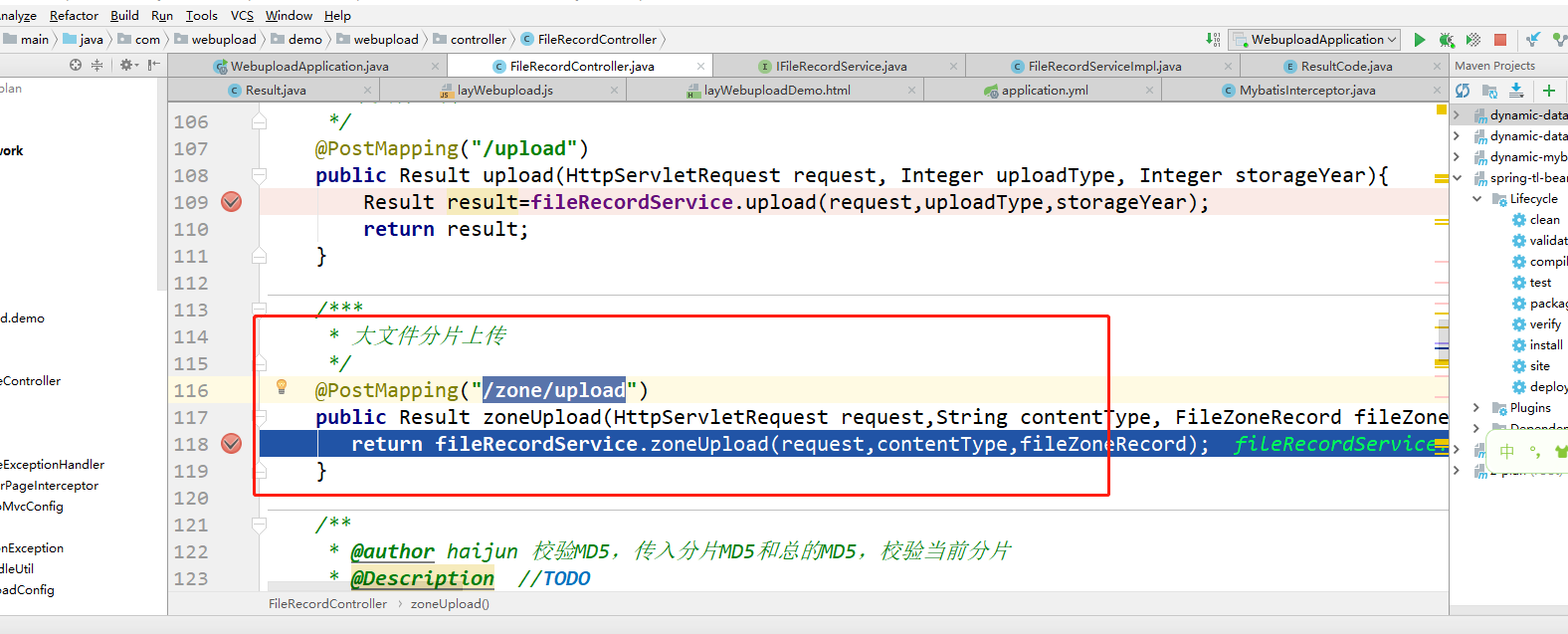




根据zone\_total\_md5（整个文件的MD5值） zone\_md5（文件分片的MD5值）去数据库的文件分片表查询，如果上传的文件小于10M ,则文件不分片，zone\_total\_md5值和zone\_md5值是一样的

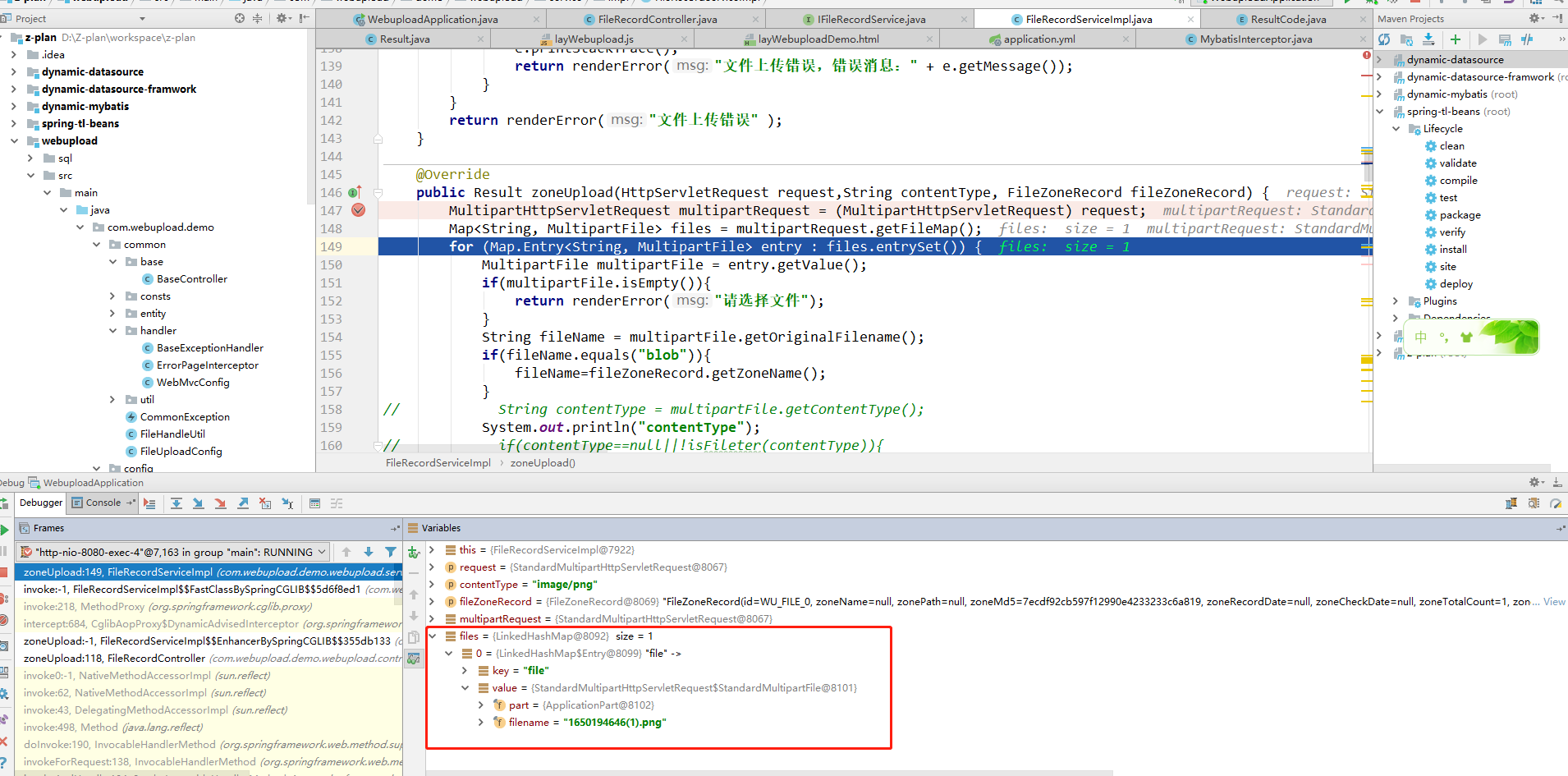
SELECT id,zone\_name,zone\_path,zone\_md5,zone\_record\_date,zone\_check\_date,zone\_total\_count,zone\_total\_md5,zone\_now\_index,zone\_start\_size,zone\_end\_size,zone\_total\_size,zone\_suffix,file\_record\_id FROM sys\_file\_zone\_record WHERE zone\_md5 = ? AND zone\_total\_md5 = ?

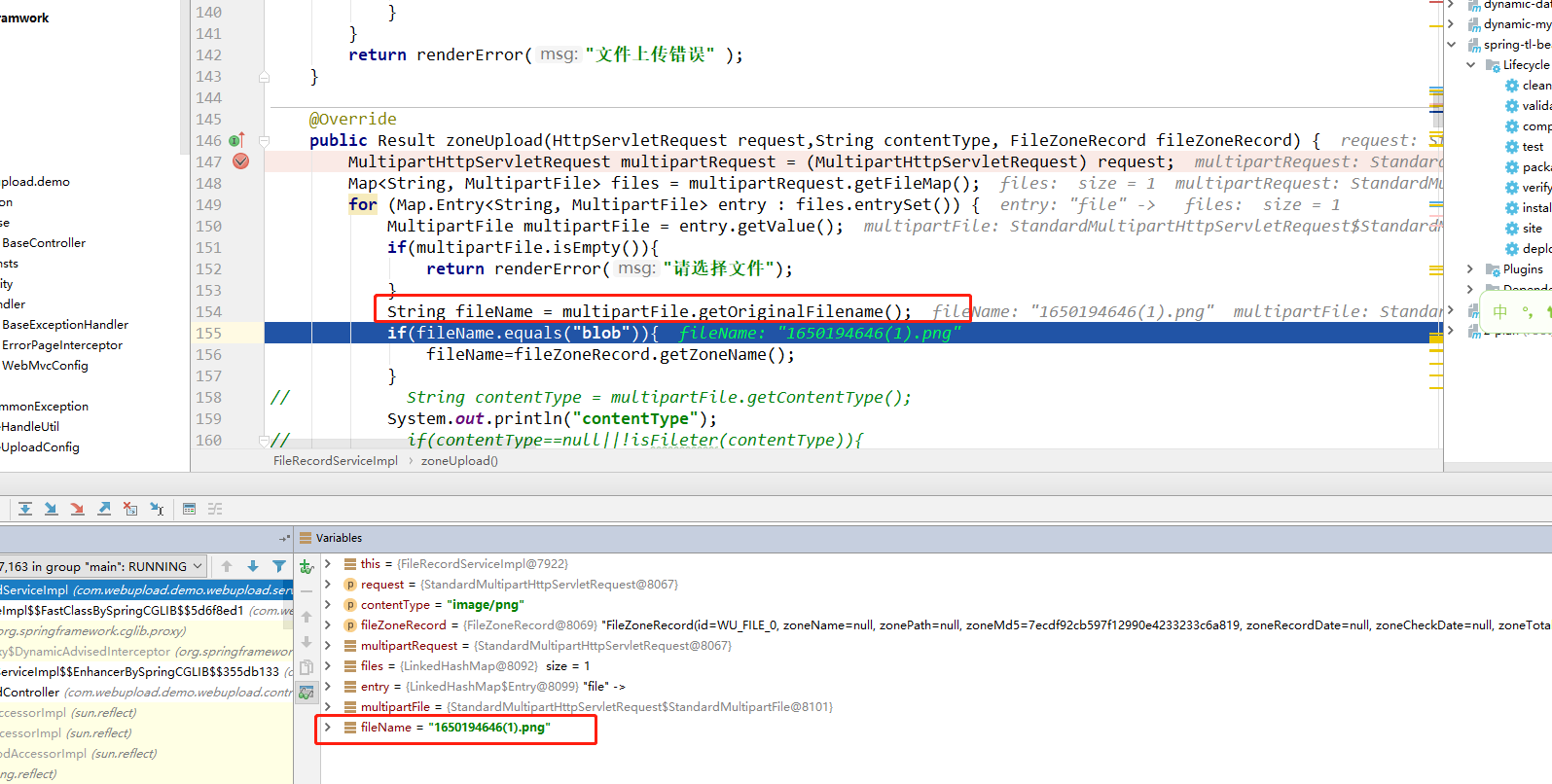
如果没有查到：则开始走/zone/upload 接口上传文件



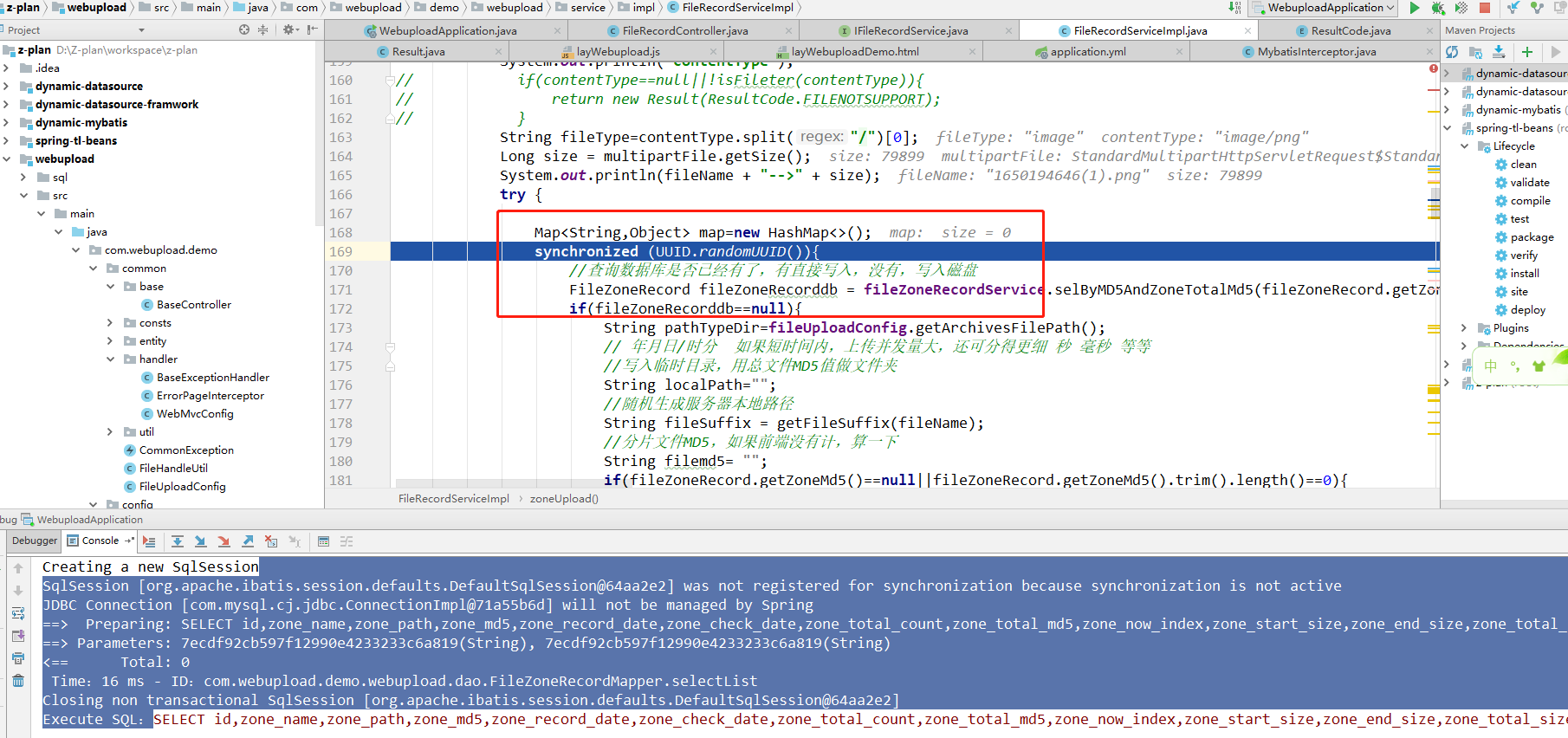
上传文件过程：

遍历fileMap集合

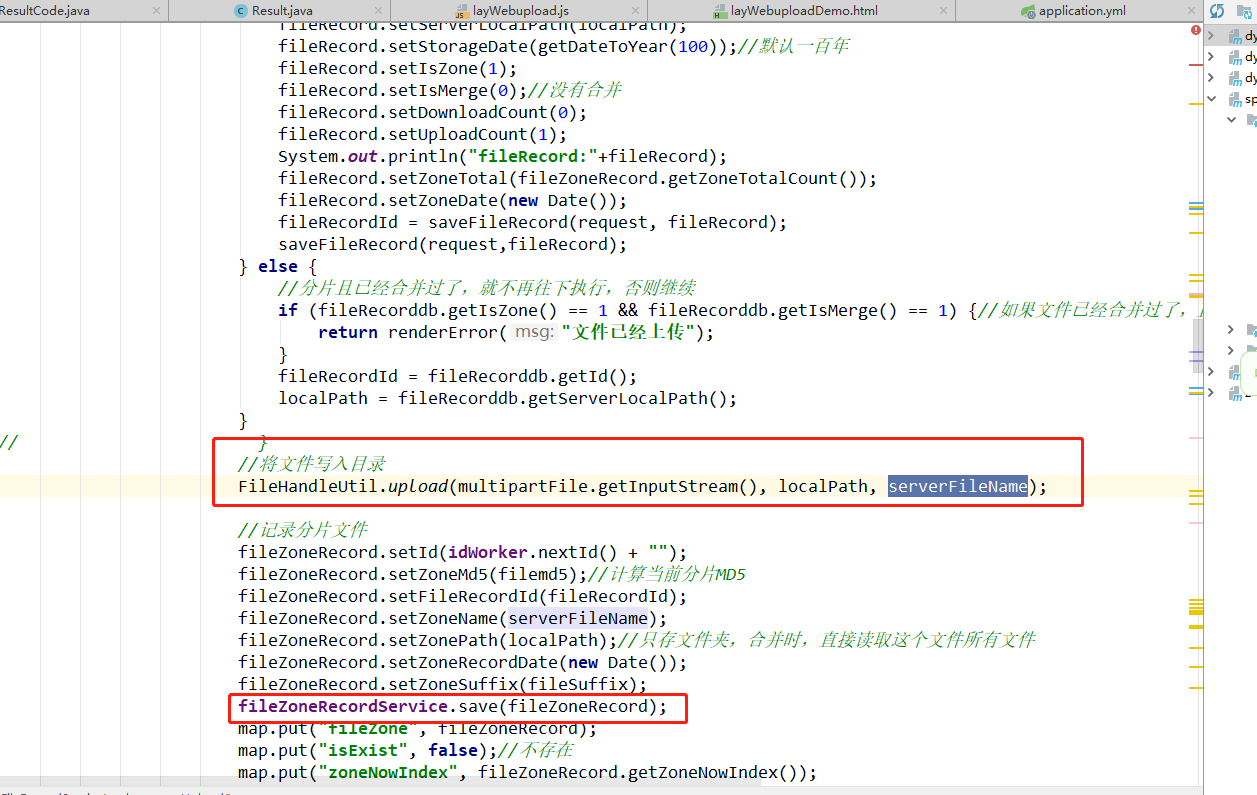




加锁：并发情况下防止文件重复上传

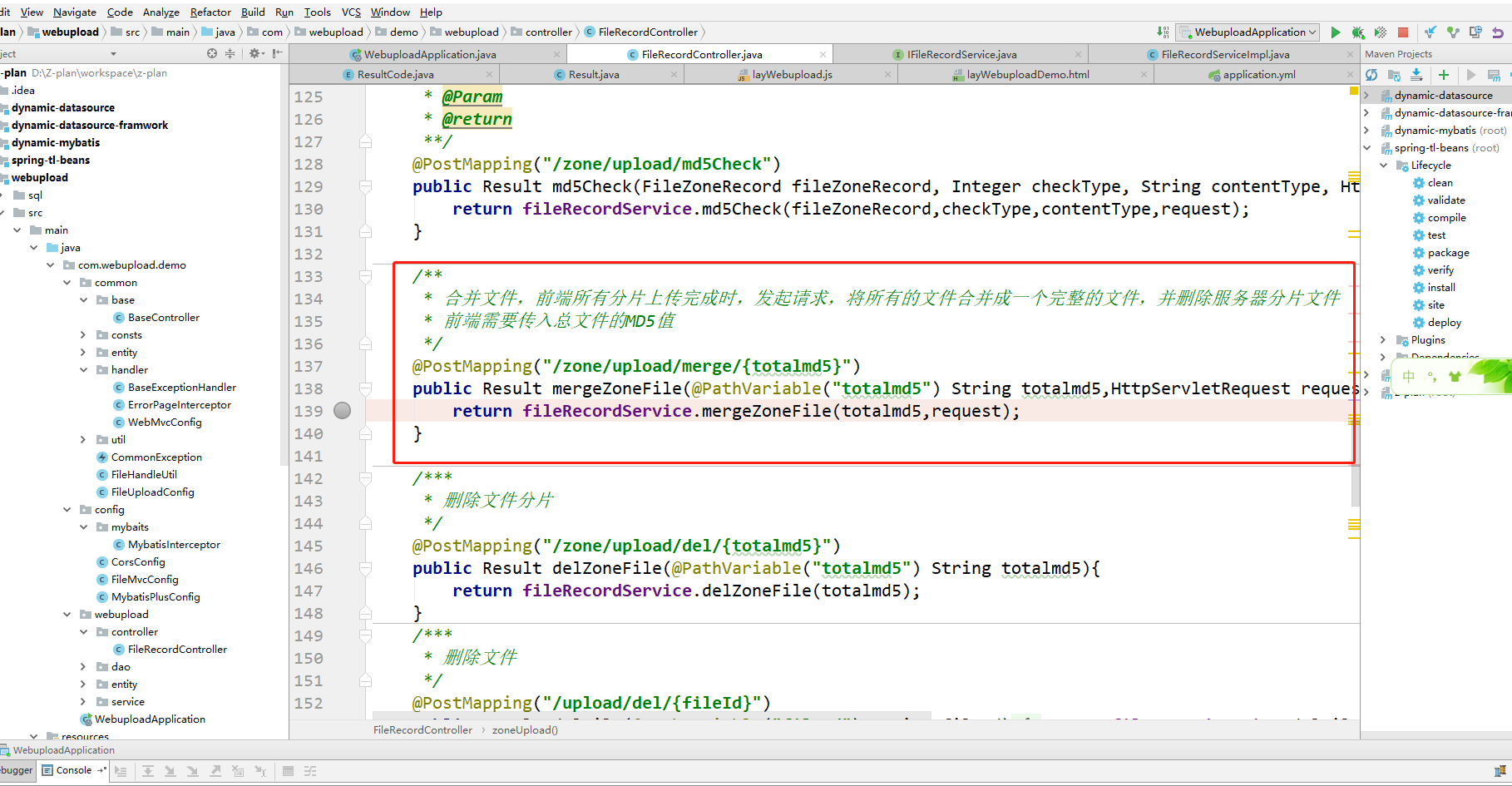


将文件的一个个分片存储到磁盘的临时目录，并将每个分片的信息（磁盘路劲，大小，MD5值，总的MD5值）存到后台数据库的sys\_file\_zone\_record 表中

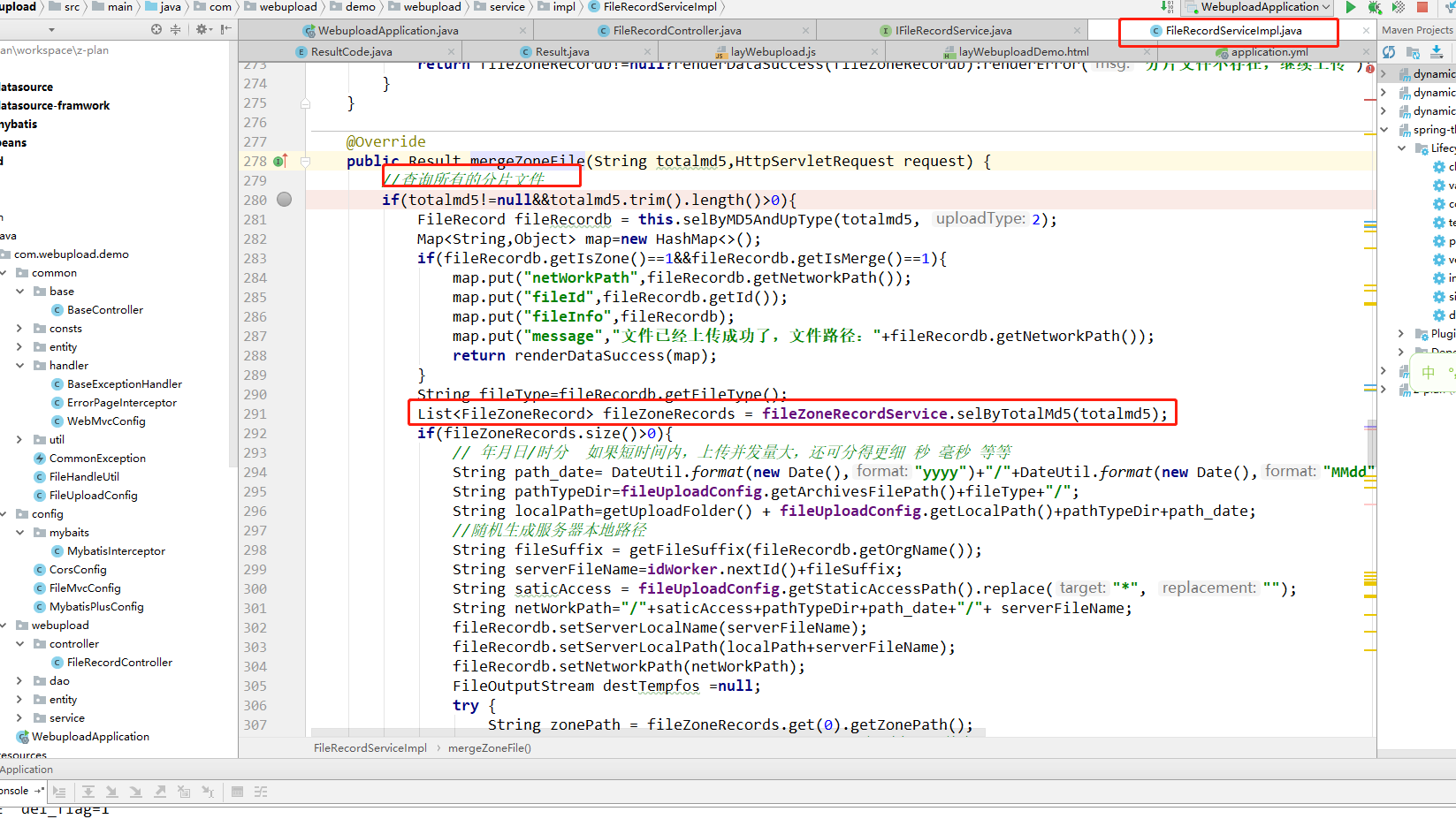


第二大步骤：合并文件

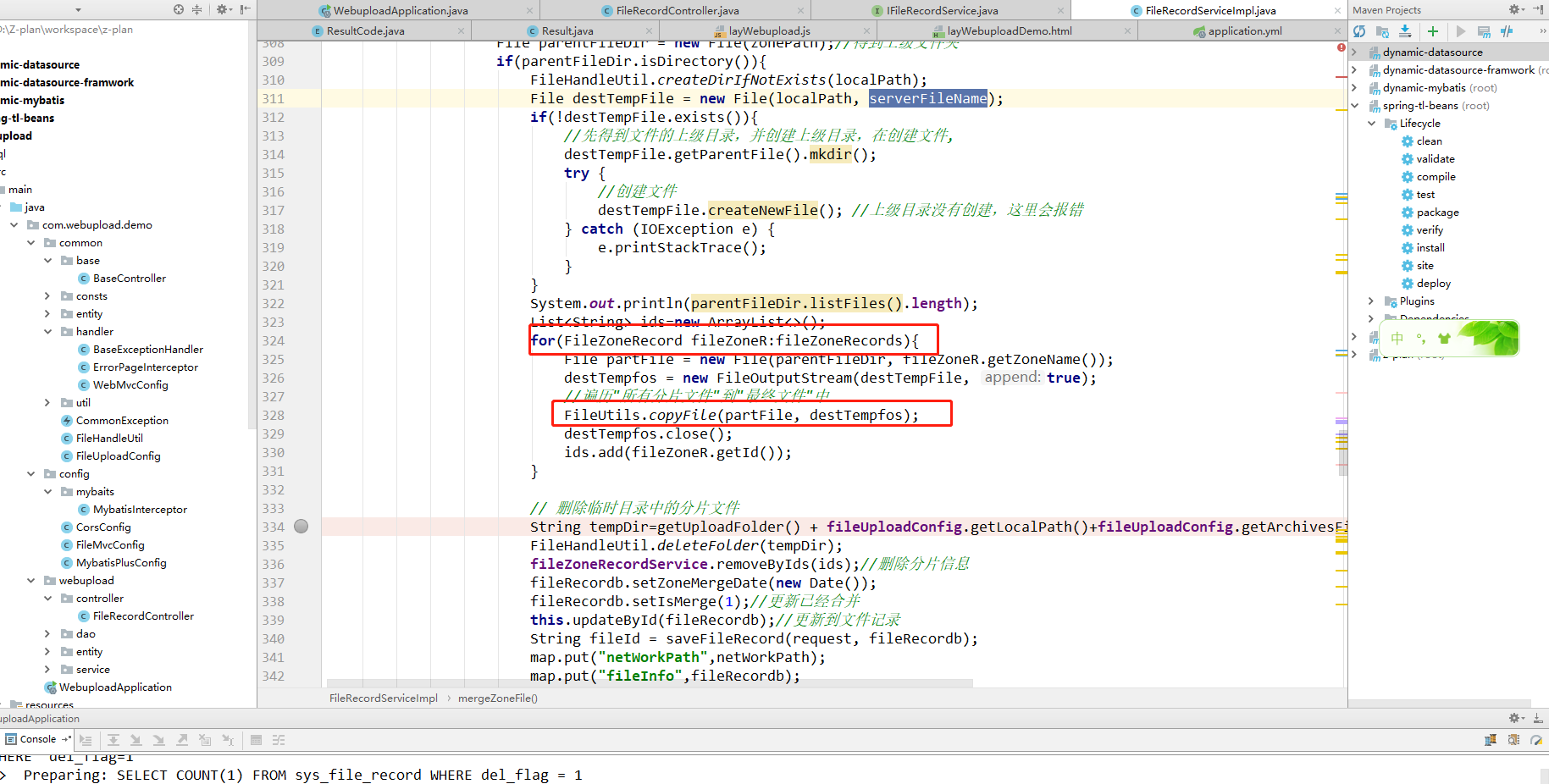
当文件的所有分片都上传之后，前台调用文件合并的接口



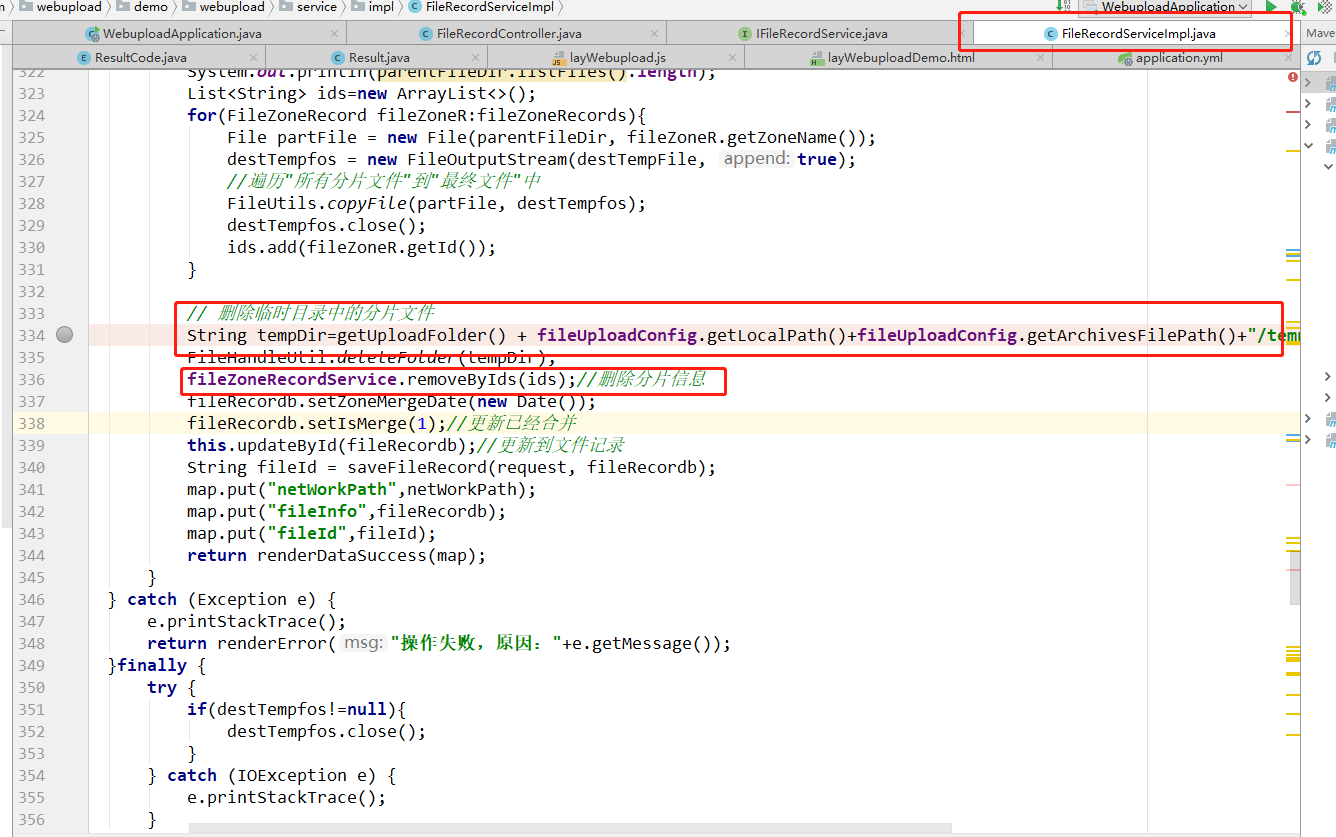
参数：传入文件的md5值，根据文件的MD5值去数据库sys\_file\_zon\_record表查询文件的所有分片的存储目录



遍历所有文件分片，创建文件流，将所有的分片文件流写入到一个文件中，完成所有文件分片的合并



---合并完成之后，删除文件的分片目录，删除数据库里的所有文件分片记录



当文件上传过程中，程序意外挂掉，再次上传同一个文件程序是如何做到断点续传的？

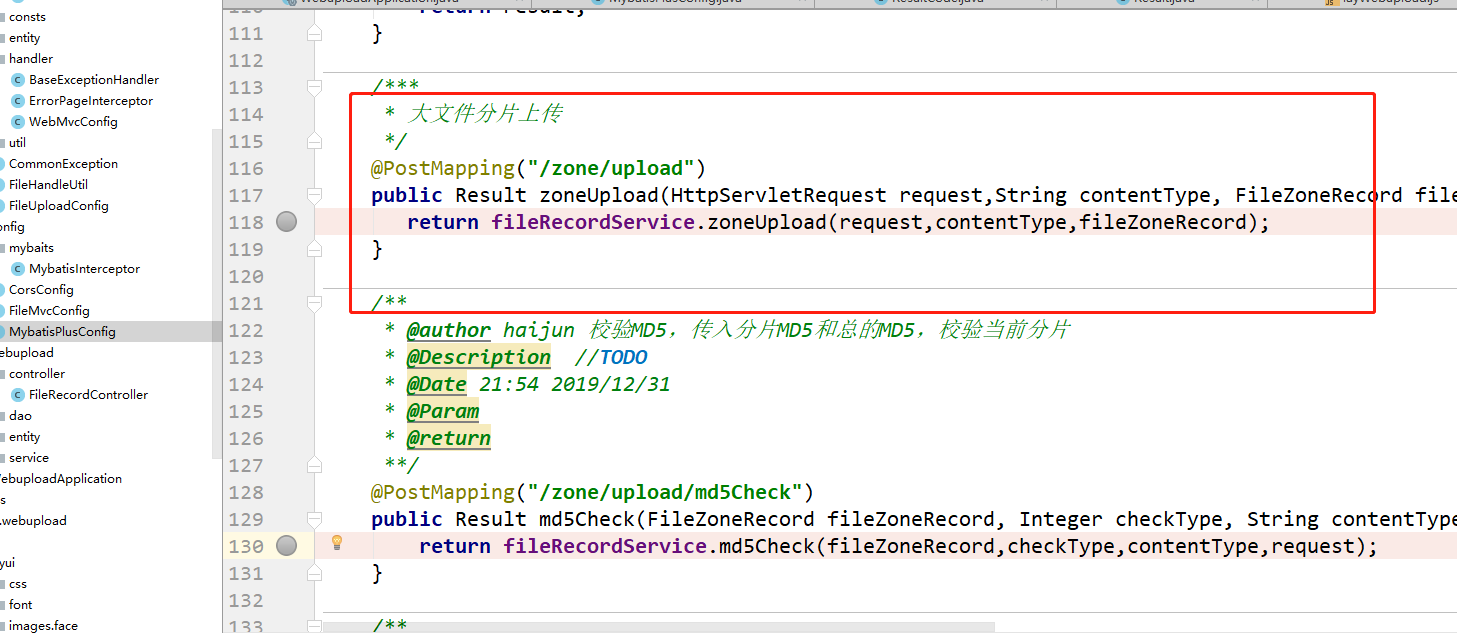
答：当文件再次上传时，后台会做校验

第一次校验checkType=1,是校验这个文件是否已经上传过，根据文件的MD5值去数据库的sys\_file\_record表查询，如果查到了说明该文件已经上传，则告诉前台文件已经上传，这就是所谓的大文件秒传。

如果数据库中没有查到这个文件的记录，说明这个文件之前没有上传或者上传过程中出现问题，没有上传完全，此时前台会根据后台返回提示准备开始上传这个文件，上传文件之前前台的webupload插件会将要上传的文件分片（默认5m一片），然后开始上传分片，分片上传之前也要走校验接口，此时checkType=2,表明是校验该该切片是否已经上传。检验的过程则是前台传入该切片的md5值和整个文件的MD5值去数据库的sys\_file-zon\_record表查

如果查到了，则返回success，前台收到success则，继续校验并上传下一个切片

如果没有查到，则通知前台走上传逻辑，调用上传分片接口，完成切片上传。这套逻辑能保证已经上传的切片不重复上传，文件没有上传的切片继续上传，从而实现文件续传的功能。



前台的一次选择多个文件上传，其实到后台还是一个个上传，没有同时上传两个或多个，只是排队一个个上传

