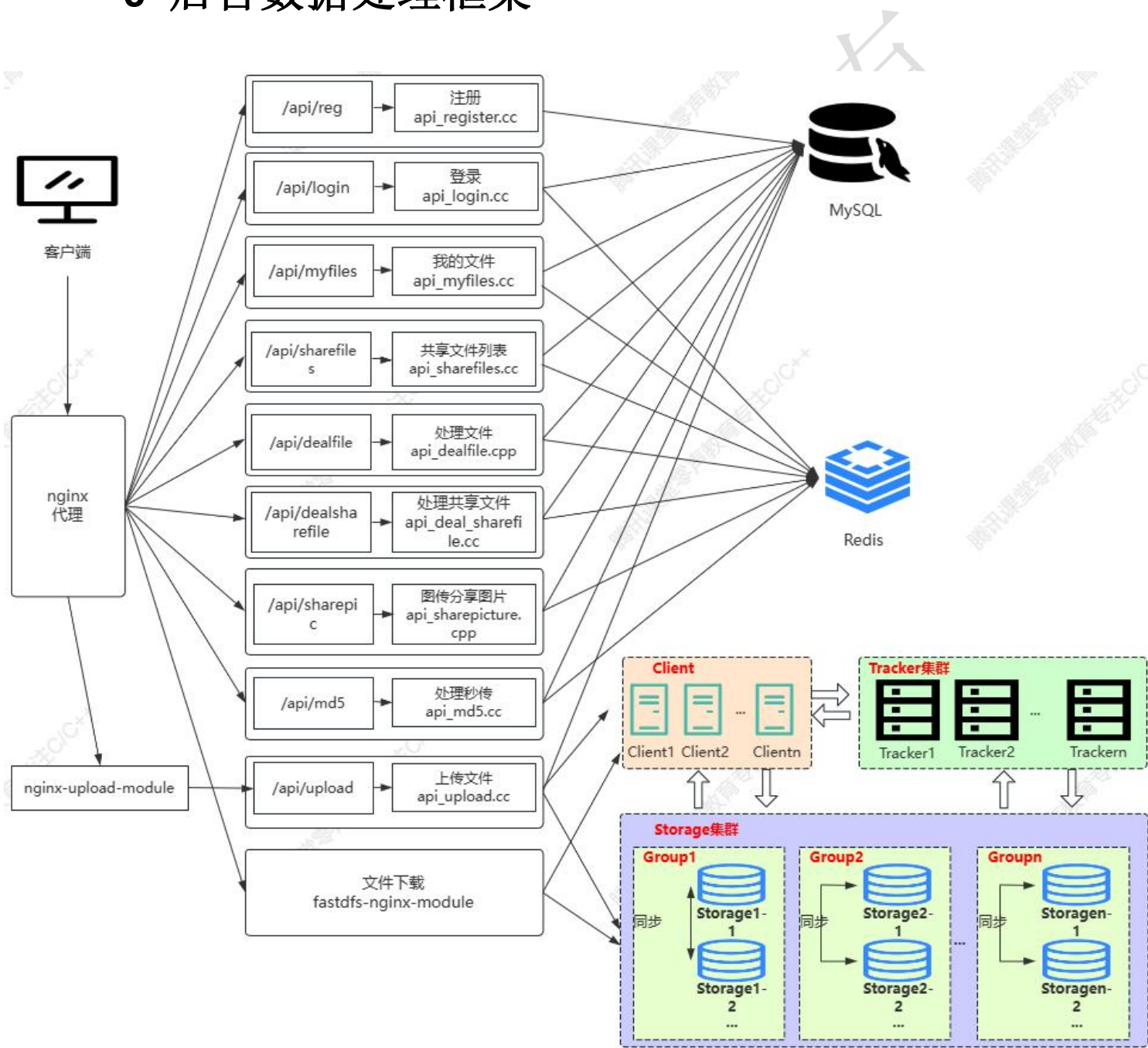


图床后台框架和 HTTP API

初稿	milo	日期	2020/07/03
审核	Darren	日期	2023/02/28
版本	V4.1	备注	无

0 后台数据处理框架



该框架需要结合数据库设计来看：我这里先把数据库的一些重点字段列举处理：

file_info 表描述一个文件的存储：

- **md5**: 文件的 md5 值，文件上传先可以先匹配数据库是否存在相同的 md5 值，如果存在意味着是同一文件。
- **file_id**: 文件 id，对应 fastdfs 的文件路径
- **url**: 文件的完整存储路径，比如 192.168.52.139:80/group1/M00/00/00/xxx.png
- **count**: 文件引用计数，每增加一个用户拥有此文件，此计数器+1

share_file_list 分享文件列表

- **user**: 文件所属用户
- **md5**: 对应的 md5 值
- **file_name**: 文件名
- **pv**: 文件下载量

user_file_count 用户文件数量，记录每个用户拥有的文件数量

- **user**: 用户名
- **count**: 文件数量

user_file_list 用户个人文件列表，记录每个人拥有的文件

- **user**: 用户名
- **md5 值**: 根据该值去 **file_info** 进一步查找具体的文件
- **file_name**: 文件名
- **shared_status**: 分享状态
- **pv**: 下载量

user_info 用户信息

- **user_name**: 用户名，唯一
- **password**: 密码，md5 加密

code 为 0 的时候是正常的，非 0 就是异常。

测试: <https://getman.cn/>

Md5 在线生成: <https://www.cmd5.com/>

Json 在线校验: <https://www.bejson.com/json/format/>

以下测试接口中涉及的 MD5 值都是 123456 的 MD5 值。

1 注册功能 api_register.cc

1.1 /api/reg 注册

1.1.1 请求和应答

注册是一个简单的 HTTP 接口，根据用户输入的注册信息，创建一个新的用户。

请求 URL

URL	http://114.215.169.66/api/reg
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
email	邮箱	必须符合 email 规范	可选	无
firstPwd	密码	md5 加密后的值	必填	无
nickName	用户昵称	不能超过 32 个字符	必填	无
phone	手机号码	不能超过 16 个字符	可选	无
userName	用户名称	不能超过 32 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败 2: 用户存在

服务示例

调用接口

http://114.215.169.66/api/reg

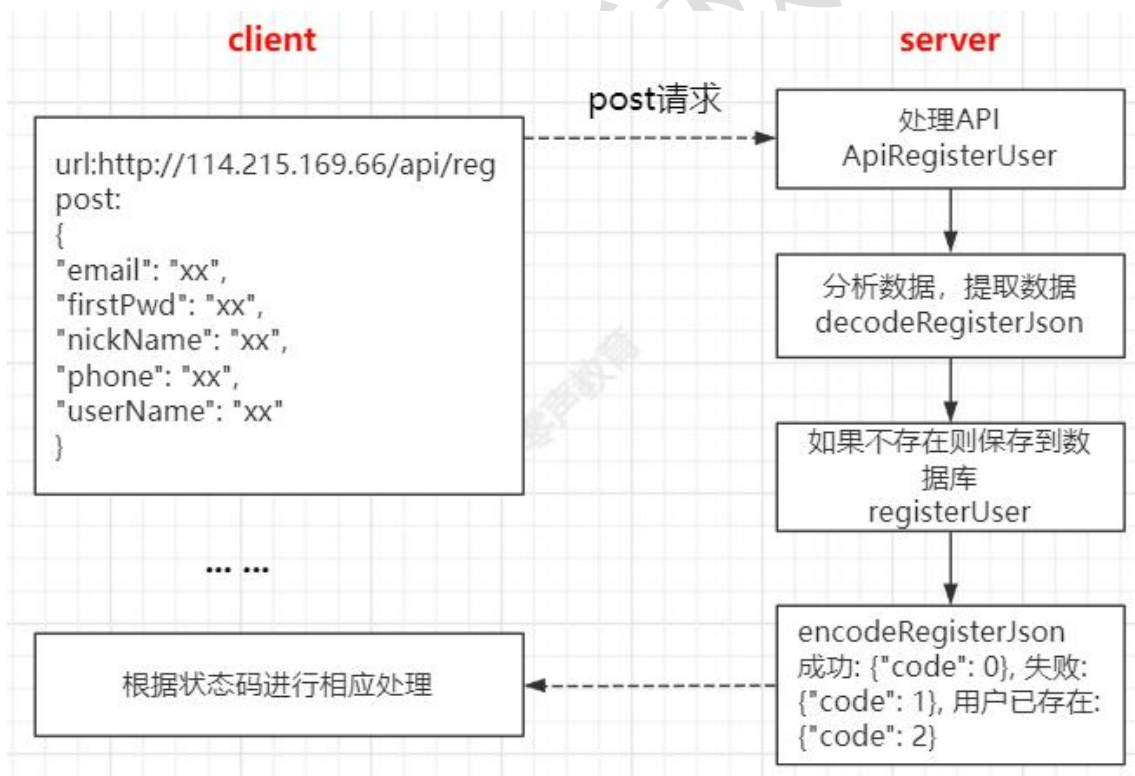
参数

```
{
  "email": "472251823@qq.com",
  "firstPwd": "e10adc3949ba59abbe56e057f20f883e",
  "nickName": "lucky",
  "phone": "18612345678",
  "userName": "qingfu"
}
```

返回结果

```
{
  "code": 0
}
```

1.1.2 处理逻辑



客户端提交的密码需要使用 md5 加密，不能提供明文的密码。

1.13 知识扩展

MD5

MD5 即 Message-Digest Algorithm 5（信息-摘要算法 5），用于确保信息传输完整一致。是计算机广泛使用的杂凑算法之一（又译摘要算法、哈希算法），主流编程语言普遍已有 MD5 实现。

理论上 MD5 是不可逆的，而且 MD5 本来也不是作加密使用，而是用来校验数据的完整性，**只是因其不可逆且稳定、快速的特点，被广泛用于对明文密码的加密。**

但是简单密码来说，破解者完全可以将一定范围内的密码字典全部计算出来之后存为数据库，之后直接查询进行破解。

用户重要信息(如密码)不应该明文保存到数据库，可以通过 MD5 加密后再保存：

id	user_name	nick_name	password
9	mil0	mil0	e10adc3949ba59abbe56e057f20f883e

盐值加密方式拓展

<https://www.yuque.com/docs/share/9276a004-7bc0-4d15-a477-c88068fb37ba?#> 《给密码加盐》

2 登陆功能 api_login.cc

2.1 /api/login 登录

2.1.1 请求和应答

登录，根据用户输入的登录信息，登录进入到后台系统。

请求 URL

URL	http://114.215.169.66/api/login
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
-----	----	------	------	-----

pwd	密码	md5 加密后的值	必填	无
user	用户名称	不能超过 32 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败
token	令牌	每次登录后, 生成的 token 不一样, 后续其他接口请求时, 需要带上 token。

服务示例

调用接口

http://114.215.169.66/api/login

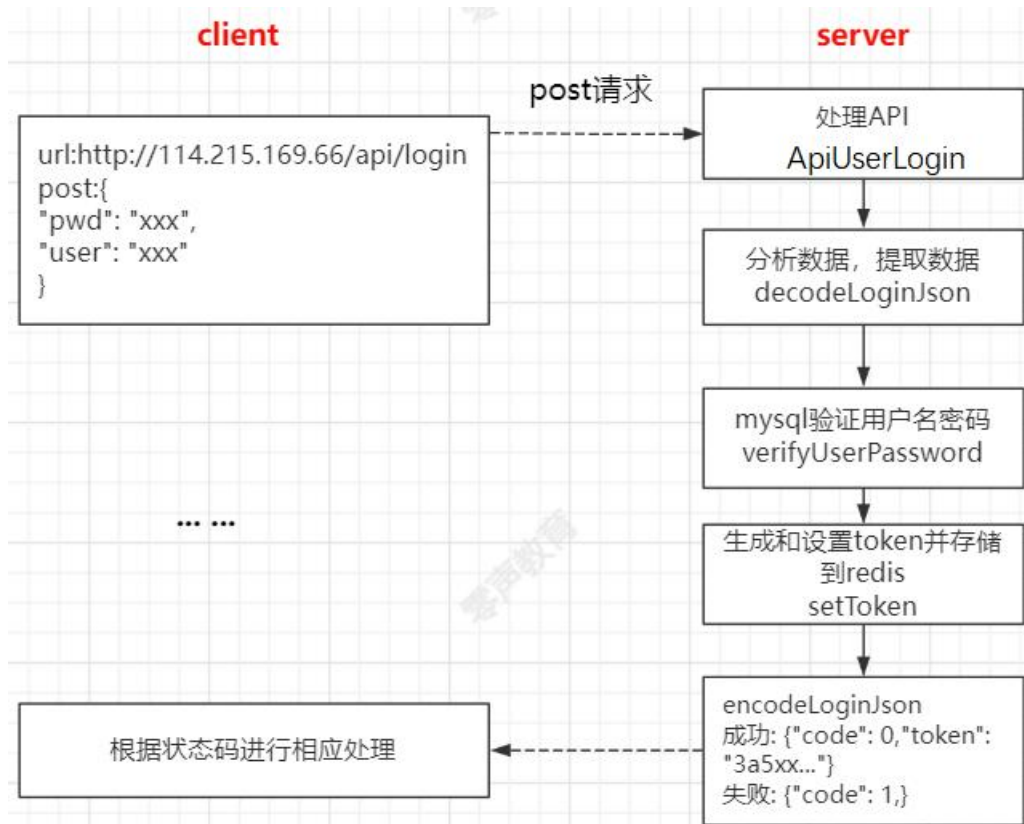
参数

```
{  
  "pwd": "e10adc3949ba59abbe56e057f20f883e",  
  "user": "qingfu"  
}
```

返回结果

```
{  
  "code": 0,  
  "token": "3a58ca22317e637797f8bcad5c047446"  
}
```

2.1.2 处理逻辑



2.1.3 知识扩展

Token 验证

Token 的中文有人翻译成“令牌”，意思就是，你拿着这个令牌，才能过一些关卡。

Token 是一个用户自定义的任意字符串。在成功提交了开发者自定义的这个字符串之后，Token 的值会保存到服务器后台。**只有服务器和客户端前端知道这个字符串**，于是 Token 就成了这两者之间的密钥，它可以让服务器确认请求是来自客户端还是恶意的第三方。

这里所说的 Token，本质上就是 http session: [《http session 介绍》](#)

使用基于 Token 的身份验证方法，在服务端不需要存储用户的登录记录。大概的流程是这样的：

- 1) 客户端使用用户名跟密码请求登录
- 2) 服务端收到请求，去验证用户名与密码
- 3) 验证成功后，服务端生成一个 Token，这个 Token 可以存储在内存、磁盘、或者数据库里，再把这个 Token 发送给客户端
- 4) 客户端收到 Token 以后可以把它存储起来，比如放在 Cookie 里或者 Local Storage

- 5) 客户端每次向服务端请求资源的时候需要带着服务端签发的 Token
- 6) 服务端收到请求，然后去验证客户端请求里面带着的 Token，如果验证成功，就向客户端返回请求的数据

简单而言，本服务器程序是采用随机数+base64+md5 生成的 token，对于过期时间是通过 redis 的 key 超时机制实现。

更进一步的阅读: <https://www.yuque.com/docs/share/18838587-4e59-4f2a-abd2-8a35471ead9d?#> 《还分不清 Cookie、Session、Token、JWT》

Base64

用记事本打开 exe、jpg、pdf 这些文件时，我们都会看到一大堆乱码，因为二进制文件包含很多无法显示和打印的字符。

当不可见字符在网络上传输时，比如说从 A 计算机传到 B 计算机，往往要经过多个路由设备，由于不同的设备对字符的处理方式有一些不同，这样那些不可见字符就有可能被处理错误，这是不利于传输的。

为了解决这个问题，我们可以先对数据进行编码，比如 编码，变成可见字符，也就是 ASCII 码可表示的可见字符，从而确保数据可靠传输。的内容是有 0 ~ 9, a ~ z, A ~ Z, +, / 组成，正好 64 个字符，这些字符是在 ASCII 可表示的范围内，属于 95 个可见字符的一部分。

所以，如果要想让记事本这样的文本处理软件能处理二进制数据，如使用 json 保存二进制信息，需要先把数据先做一个 Base64 编码，统统变成可见字符，再保存。

在 Base64 中的可打印字符包括大写英文字母 A-Z、小写英文字母 a-z、阿拉伯数字 0-9，这样共有 62 个字符，此外两个可打印符号在不同的系统中而不同，通常用加号 (+) 和正斜杠 (/)。外加“补全符号”，通常用等号 (=)。

Base64 是一种用 64 个字符来表示任意二进制数据的方法，常用于在 URL、Cookie、网页中传输少量二进制数据。

Base64 要求把每三个 8Bit 的字节转换为四个 6Bit 的字节 ($3 \times 8 = 4 \times 6 = 24$)，然后把 6Bit 再添两位高位 0，组成四个 8Bit 的字节，也就是说，转换后的字符串理论上将要比原来的长 1/3。

(原文)转换前 11111111, 11111111, 11111111 (二进制)

(Base64 编码)转换后 00111111, 00111111, 00111111, 00111111 (二进制)

3 用户文件列表 api_myfiles.cc

3.1 /api/myfiles&cmd=count 文件数量

3.1.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/myfiles?cmd=count
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	token	同上	必填	无
user	用户名称	不能超过 32 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 验证成功 1: 验证失败
total	文件数量	

服务示例

调用接口

http://114.215.169.66/api/myfiles?cmd=count

参数

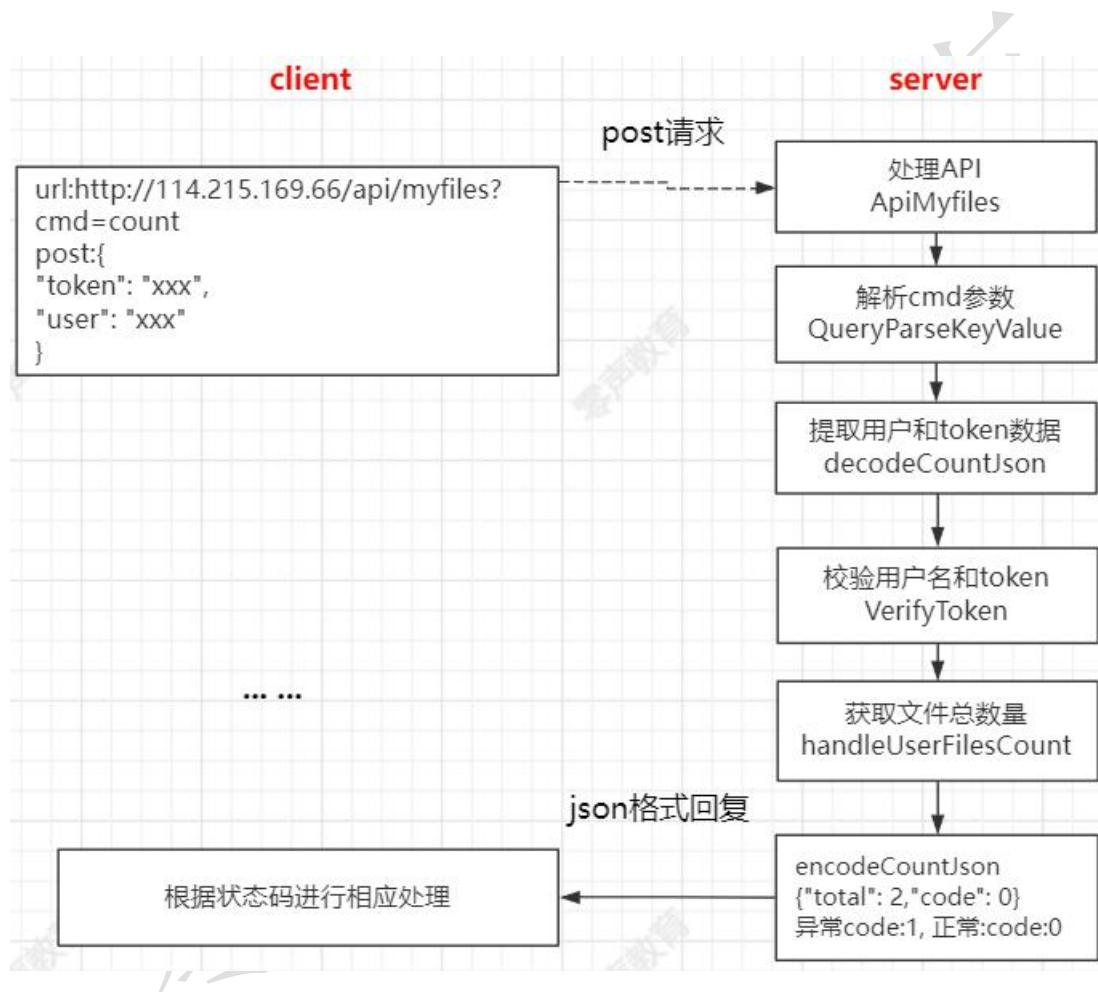
```
{
  "token": "3a58ca22317e637797f8bcad5c047446",
  "user": "qingfu"
}
```

返回结果

```
{  
  "total": 1,  
  "code": 0  
}
```

如果没有文件则返回{"code": 0, "total": 0}。

3.1.2 处理逻辑



3.2 /api/myfiles&cmd=normal 文件列表

3.2.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/myfiles?cmd=normal
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	令牌	同上	必填	无
user	用户名称	不能超过 32 个字符	必填	无
count	文件个数	文件个数需大于 0	必填	无
start	开始位置		必填	无

返回结果参数说明

名称	含义	规则说明
files	文件结果集	"code": 0 正常, 1 失败 "count": 返回的当前文件数量, 比如 2 "total": 个人文件总共的数量 "user": 用户名称, "md5": md5 值, "create_time": 创建时间, "file_name": 文件名, "share_status": 共享状态, 0 为没有共享, 1 为共享 "pv": 文件下载量, 下载一次加 1 "url": URL, "size": 文件大小, "type": 文件类型

服务示例

调用接口

<http://114.215.169.66/api/myfiles?cmd=normal>

参数

```
{
  "count": 2,
  "start": 0,
  "token": "3a58ca22317e637797f8bcad5c047446",
  "user": "qingfu"
}
```

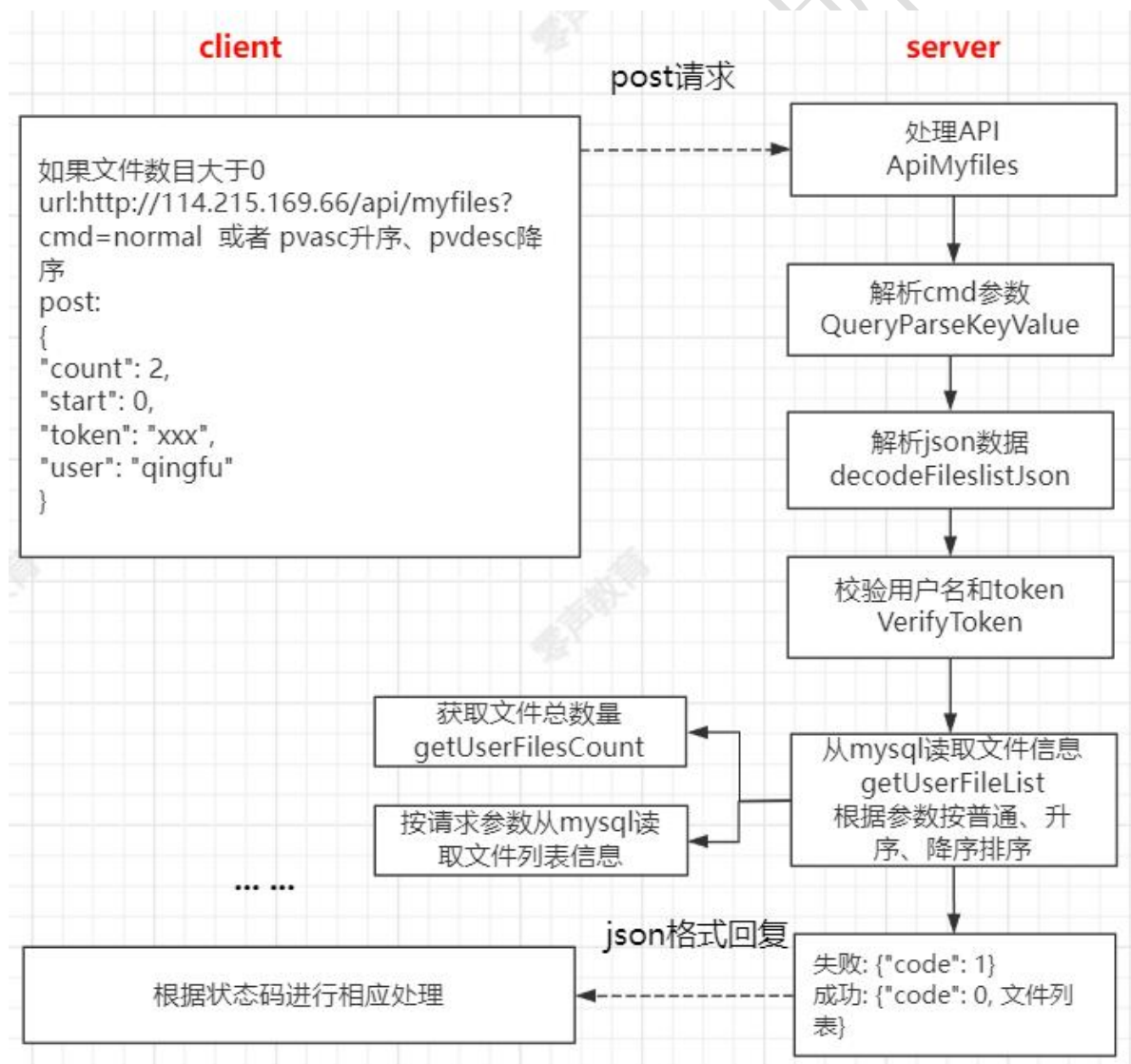
返回结果

```
{
  "code": 0,
  "count": 2, // 如果为 0 则不需要解析 files
  "total": 100, 个人文件总共的数量 100
  "files": [
    {
      "user": "qingfu",
      "md5": "6c5fa2864bb264c91167258b3e478fa0",
      "create_time": "2021-05-15 11:31:00",
      "file_name": "Qt5Svg.dll",
      "share_status": 0,
      "pv": 1,
      "url":
        "http://114.215.169.66:80/group1/M00/00/00/eBuDxWCfQHSATopyAAV8AJV_1mw866.dll",
      "size": 359424,
      "type": "dll"
    },
    {
      "user": "qingfu",
      "md5": "c32b5d82be3d2fcec96de06e81f87814",
      "create_time": "2021-05-15 11:33:04",
      "file_name": "1.png",
      "share_status": 0,
      "pv": 0,
      "url":
        "http://114.215.169.66:80/group1/M00/00/00/eBuDxWCfQPCAapRFAACtu7ei-L0000.png",
      "size": 44475,
      "type": "png"
    }
  ]
}
```

文件信息字段:

- ✧ code: 0 正常, 1 错误
- ✧ user: 文件上传者
- ✧ md5: 文件 md5 码
- ✧ time: 文件上传的时间
- ✧ filename : 用户名
- ✧ share_status : 文件是否已共享, 已经共享值为 1, 否则值为 0
- ✧ pv : 文件当前的下载量, 下载一次, pv++
- ✧ url : 文件存放在 fastdfs 分布式存储服务器的 url 地址
- ✧ size : 文件大小
- ✧ type : 文件类型

3.2.1 处理逻辑



3.3 /api/myfiles&cmd=pvasc 文件列表(按下载量升序)

和 cmd=normal 逻辑一致，只是稍微改了 sql 语句的查询方式。

升序：order by pv asc

3.4 /api/myfiles&cmd=pvdesc 文件列表(按下载量降序)

和 cmd=normal 逻辑一致，只是稍微改了 sql 语句的查询方式。

降序：order by pv desc

4 上传文件 api_md5.cc（秒传）

用于秒传文件的请求。

上传文件的时候：

- 先调用 md5 接口判断服务器是否有该文件，如果 md5 调用成功，则说明服务器已经存在该文件，客户端不需要再去调用 upload 接口上传文件。
- 如果不成功则客户端继续调用 upload 接口上传文件。

4.1 /api/md5 请求对比数据库文件信息的 md5

4.1.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/md5
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	令牌	同上	必填	无
md5	md5 值	不能超过 32 个字符	必填	无
filename	文件名称	不能超过 128 个字符	必填	无
user	用户名称	不能超过 32 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 秒传成功 1: 秒传失败 4: token 校验失败 5: 文件已存在

服务示例

调用接口

http://114.215.169.66/api/md5

参数

```
{
  "fileName": "ui_buttongroup.h",
  "md5": "a89390d867d5da18c8b1a95908d7c653",
  "token": "3a58ca22317e637797f8bcad5c047446",
  "user": "qingfu"
}
```

返回结果

```
{
  "code": 0
}
```

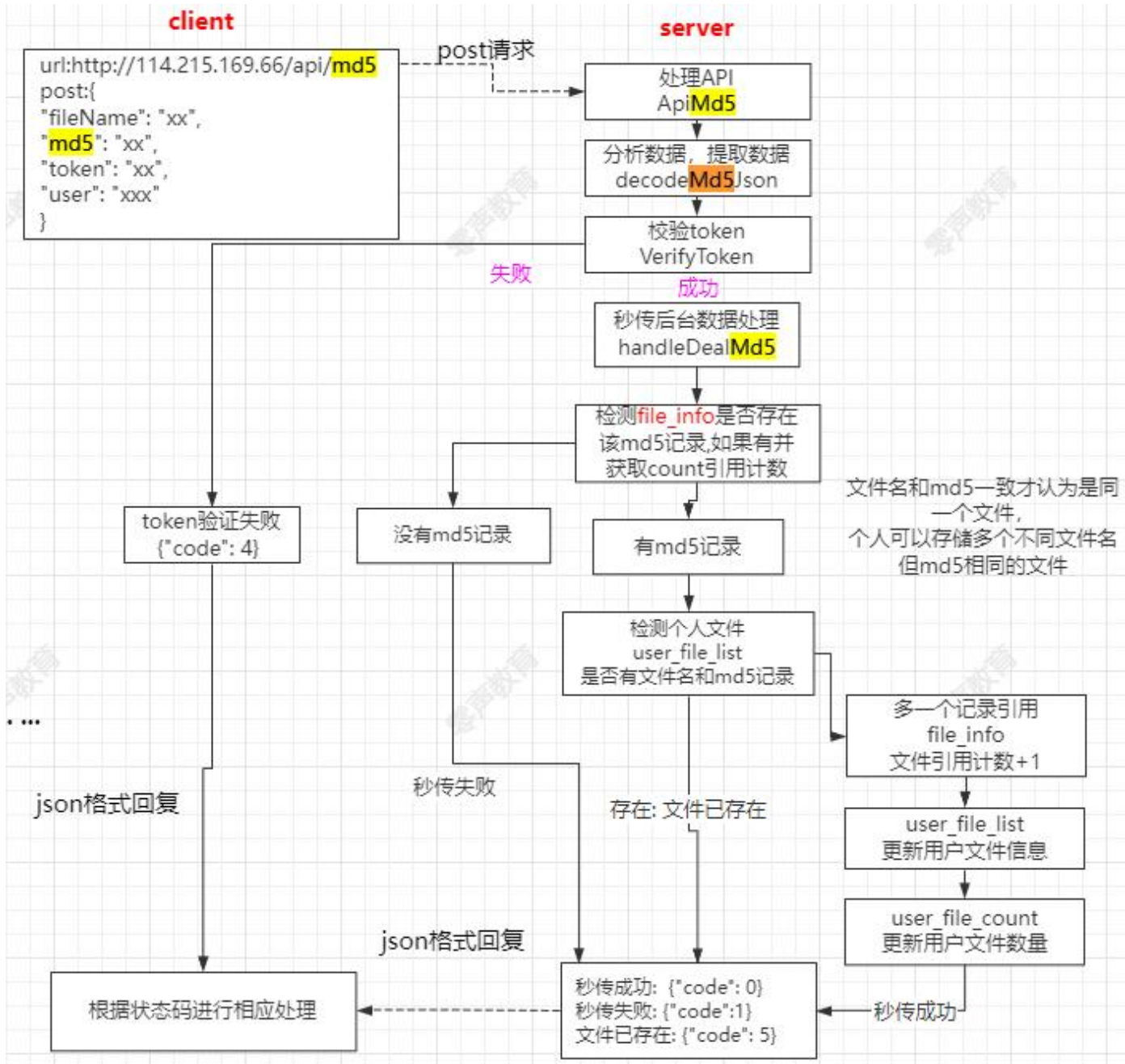
4.1.2 处理逻辑

每个文件都有一个唯一的 MD5 值（比如 2bf8170b42cc7124b04a8886c83a9c6f），就好比每个人的指纹都是唯一的一样，校验 MD5 就是用来确保文件在传输过程中未被修改过。

- 客户端在上传文件之前将文件的 MD5 码上传到服务器
- 服务器端判断是否已存在此 MD5 码，如果存在，说明该文件已存在，则此文件无需再上传，在此文件的计数器加 1，说明此文件多了一个用户共用。

file_info count
 插入用户文件列表
 用户文件计数+1

- 如果服务器没有此 MD5 码，说明上传的文件是新文件，则真正上传此文件



5 上传文件 api_upload.cc

- 客户端在上传文件之前将文件的 MD5 码上传到服务器
- 服务器端判断是否已存在此 MD5 码，如果存在，说明该文件已存在，则此文件无需再上传，在此文件的计数器加 1，说明此文件多了一个用户共用。

file_info count
 插入用户文件列表
 用户文件计数+1

- 如果服务器没有此 MD5 码，说明上传的文件是新文件，则真正上传此文件。本流程就是处理文件上传的逻辑。

上传文件的总体逻辑：

1. 先通过 nginx-upload-module 模块上传文件到临时目录
2. nginx-upload-module 模块上传文件后通知/api/upload 后端处理程序：
3. 后端处理程序 ApiUpload 函数解析文件信息，然后将临时文件上传到 fastdfs

5.1 /api/upload 上传文件

5.1.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/upload
请求方式	POST
HTTP 版本	1.1
Content-Type	application/octet-stream

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 上传成功 1: 上传失败

5.1.2 处理逻辑

客户端上传文件信息具有一定的格式，先把文件上传到 nginx

```
-----WebKitFormBoundary88asdgtgewx\r\n
```

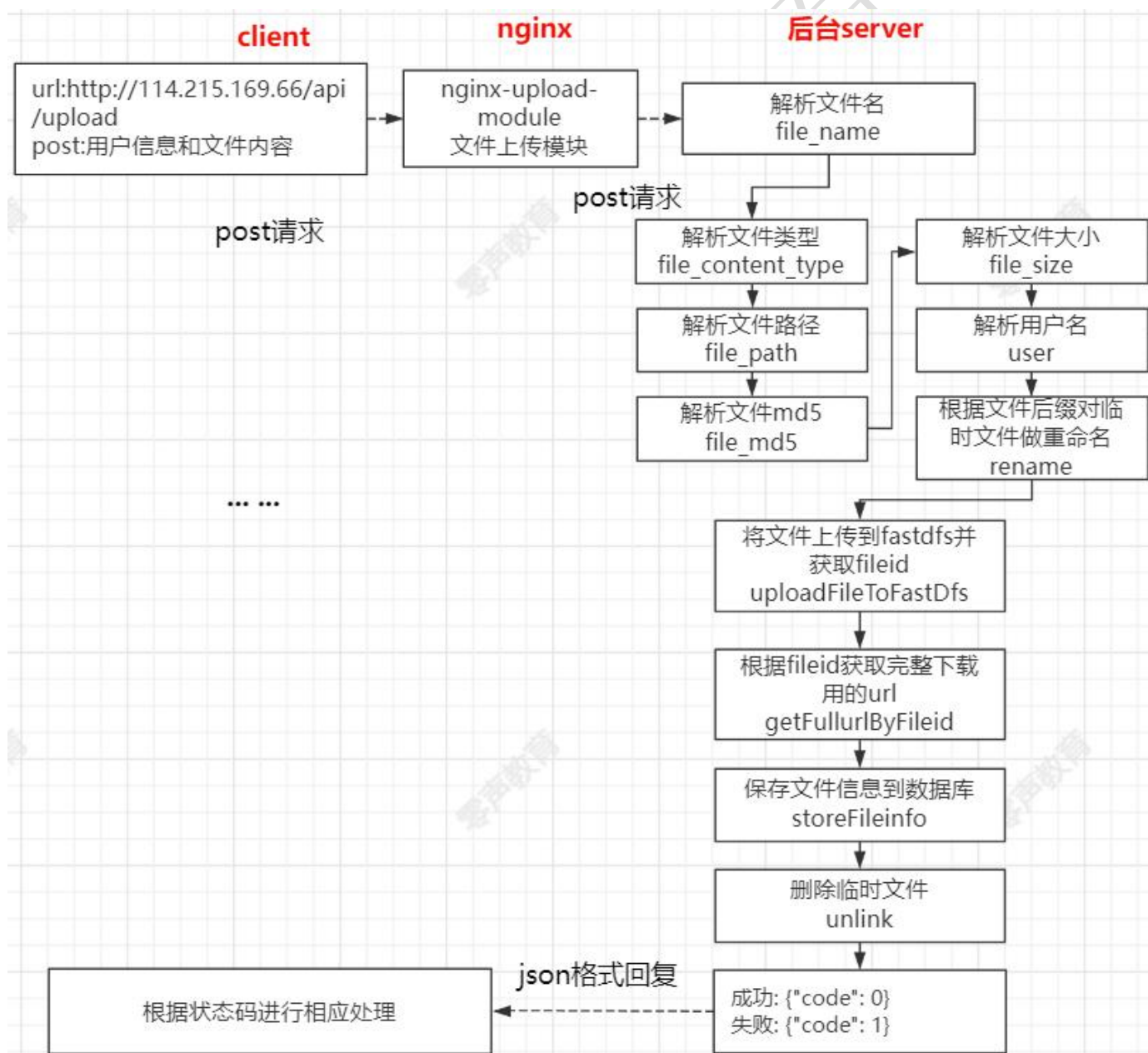
```
Content-Disposition: form-data; user = "milo"; filename = "xxx.jpg"; md5 = "xxxx"; size = 1024\r\n
```

```
Content-Type: application/octet-stream\r\n
```

```
\r\n
```

```
真正的文件内容\r\n
```

```
-----WebKitFormBoundary88asdgtgewx
```



6 获取共享文件或下载榜 api_sharefiles.cc

分 3 个接口：

- 获取共享文件个数 <http://114.215.169.66/api/sharefiles?cmd=count>
- 获取共享文件列表 <http://114.215.169.66/api/sharefiles?cmd=normal>
- 获取共享文件下载排行榜 <http://114.215.169.66/api/sharefiles?cmd=pvdesc>

6.1/api/sharefiles?cmd=count 获取共享文件数量

6.1.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/sharefiles?cmd=count
请求方式	GET
HTTP 版本	1.1
Content-Type	application/x-www-form-urlencoded

请求参数

无 公共的请求，这里暂时不做 token 验证

返回结果参数说明

名称	含义	规则说明
code	结果码	0：正常；1：失败
total	总数量	

服务示例

调用接口

<http://114.215.169.66/api/sharefiles?cmd=count>

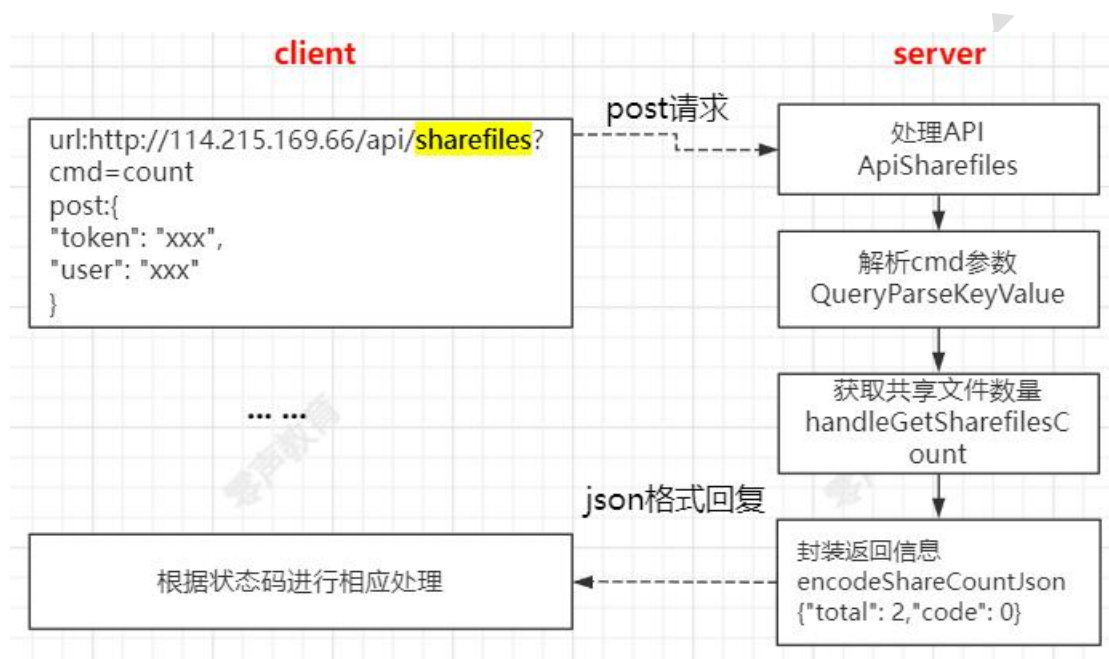
参数

返回结果

```
{  
  "code": 0,  
  "total": 1  
}
```

如果没有文件则返回{"code": 0, "total": 0}。

6.1.2 处理逻辑



6.2 /api/sharefiles?cmd=normal 获取共享文件列表

6.2.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/sharefiles?cmd=normal
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
count	数量		必填	无
start	开始位置		必填	无

返回结果参数说明

名称	含义	规则说明
files	文件结果集	"code": 0: 正常; 1: 失败 "count": 2, // 分页返回数量, 如果为 0 则不需要解析 files "total": 2, 总的文件数量 "user": 用户名称, "md5": md5 值, "create_time": 创建时间, "file_name": 文件名, "share_status": 共享状态, 0 为没有共享, 1 为共享 "pv": 文件下载量, 下载一次加 1 "url": URL, "size": 文件大小, "type": 文件类型

服务示例

调用接口

http://114.215.169.66/api/sharefiles?cmd=normal

参数

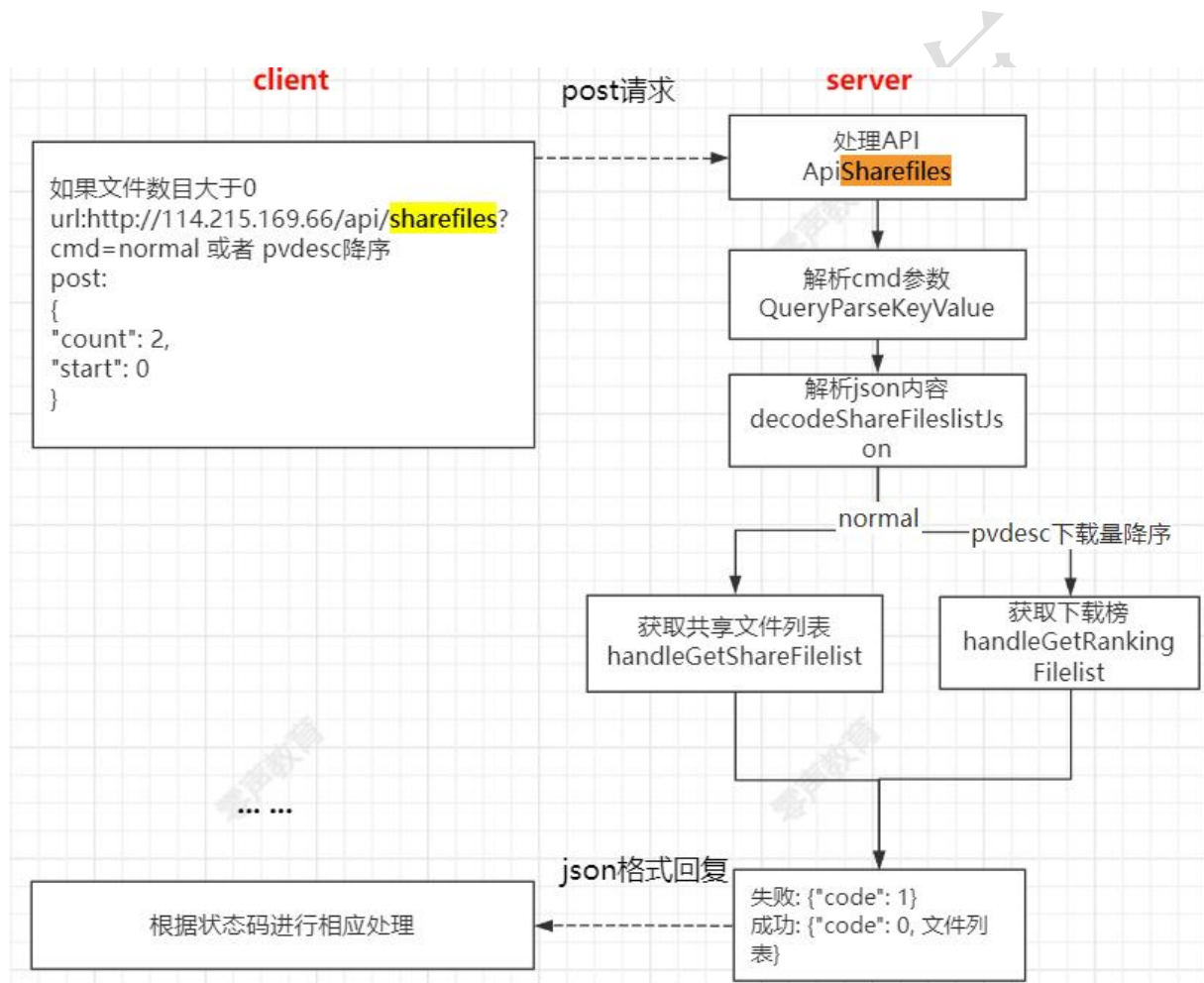
```
{  
  "count": 1,  
  "start": 0  
}
```

返回结果

```
{  
  "code": 0,  
  "count": 1, // 如果为 0 则不需要解析 files  
  "total": 2, 总的文件数量  
  "files": [{  
    "user": "qingfu",  
    "md5": "602fdf30db2aacf517badf4565124f51",  
    "file_name": "Makefile",  
    "share_status": 1,  
  }]
```

```
"pv": 0,  
"create_time": "2020-07-03 23:41:15",  
"url": "http://172.16.0.15:80/group1/M00/00/00/rBAAD17_Jn6AM-iuAACBBC5AIsc2532509",  
"size": 33028,  
"type": "null"  
}]  
}
```

6.2.2 处理逻辑



6.3 /api/sharefiles?cmd=pvdesc 获取共享文件列表

6.3.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/sharefiles?cmd=pvdesc
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
count	数量		必填	无
start	开始位置		必填	无

返回结果参数说明

名称	含义	规则说明
files	文件结果集	"filename": 文件名, "pv": 文件下载量, 下载一次加 1
code	返回码	0: 正常; 1: 失败
cout	分页文件数量	比如 1, 如果为 0 则不需要解析 files
total	总文件数量	比如 2

服务示例

调用接口

http://114.215.169.66/api/sharefiles?cmd=pvdesc

参数

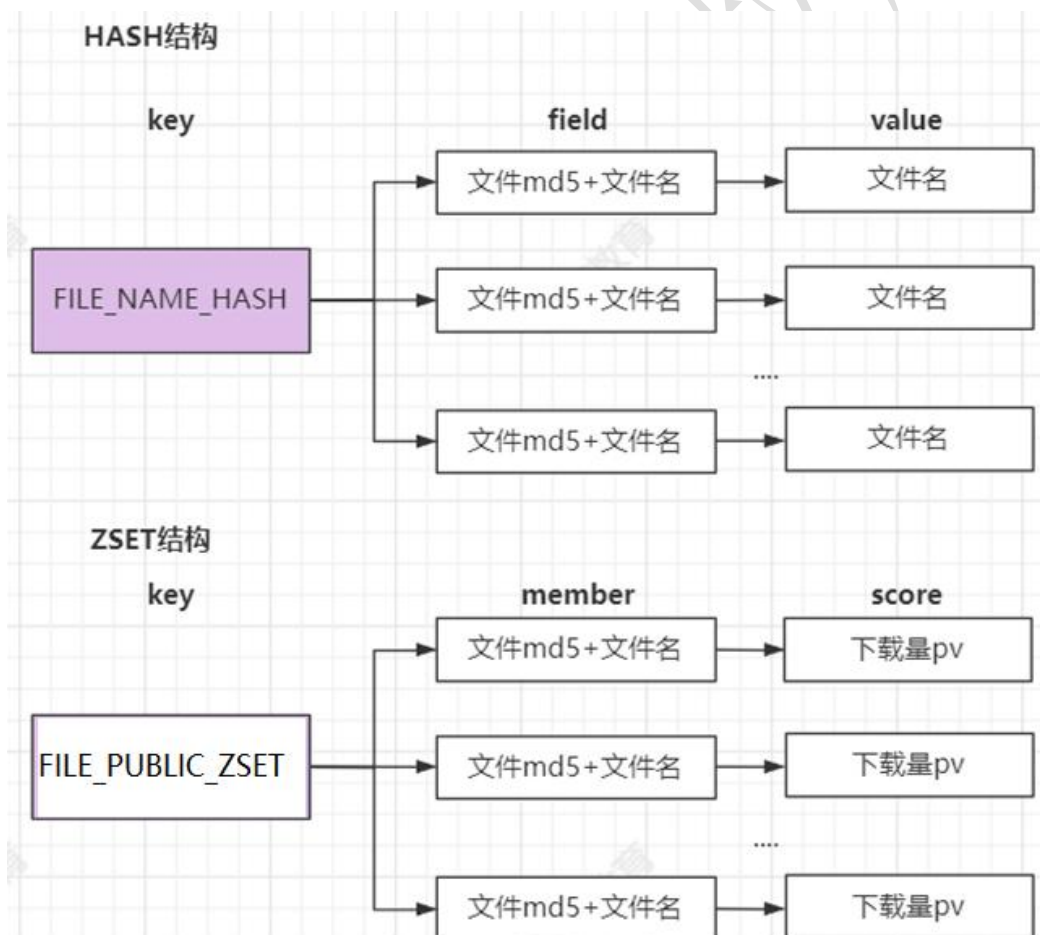
```
{  
  "count": 1,  
  "start": 0  
}
```

返回结果

```
{
  "code": 0,
  "count": 1,
  "total": 2,
  "files": [{
    "filename": "ui_buttongroup.h",
    "pv": 0
  }]
}
```

6.3.2 处理逻辑

排行榜的逻辑比较简单，就是使用 redis 的 ZSET 做排行榜。
这里涉及到 mysql 和 redis，获取返回的是文件名和下载量。



这里文件名可能重名，所以这里用了文件 md5+文件名作为唯一 ID。

1. 先从 ZSET 获取排行榜，此时的 member 是 md5+文件名，score 是下载量 pv

- 然后将 member 的 md5+文件名 通过 HASH 查找对应的文件名 filename
- 将文件名 filename 和下载量 pv 返回给前端展示。

下载文件后，需要更新排行榜。

7 分享/删除文件,更新下载计数 api_dealfile.cc

7.1 /api/dealfile?cmd=share 分享文件

7.1.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/dealfile?cmd=share
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	令牌	同上	必填	无
user	用户名称	不能超过 32 个字符	必填	无
md5	md5 值	md5 加密后的值	必填	无
filename	文件名称	不能超过 128 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败 3: 别人已经分享此文件

		4: token 验证失败
--	--	---------------

服务示例

调用接口

<http://114.215.169.66/api/dealfile?cmd=pv>

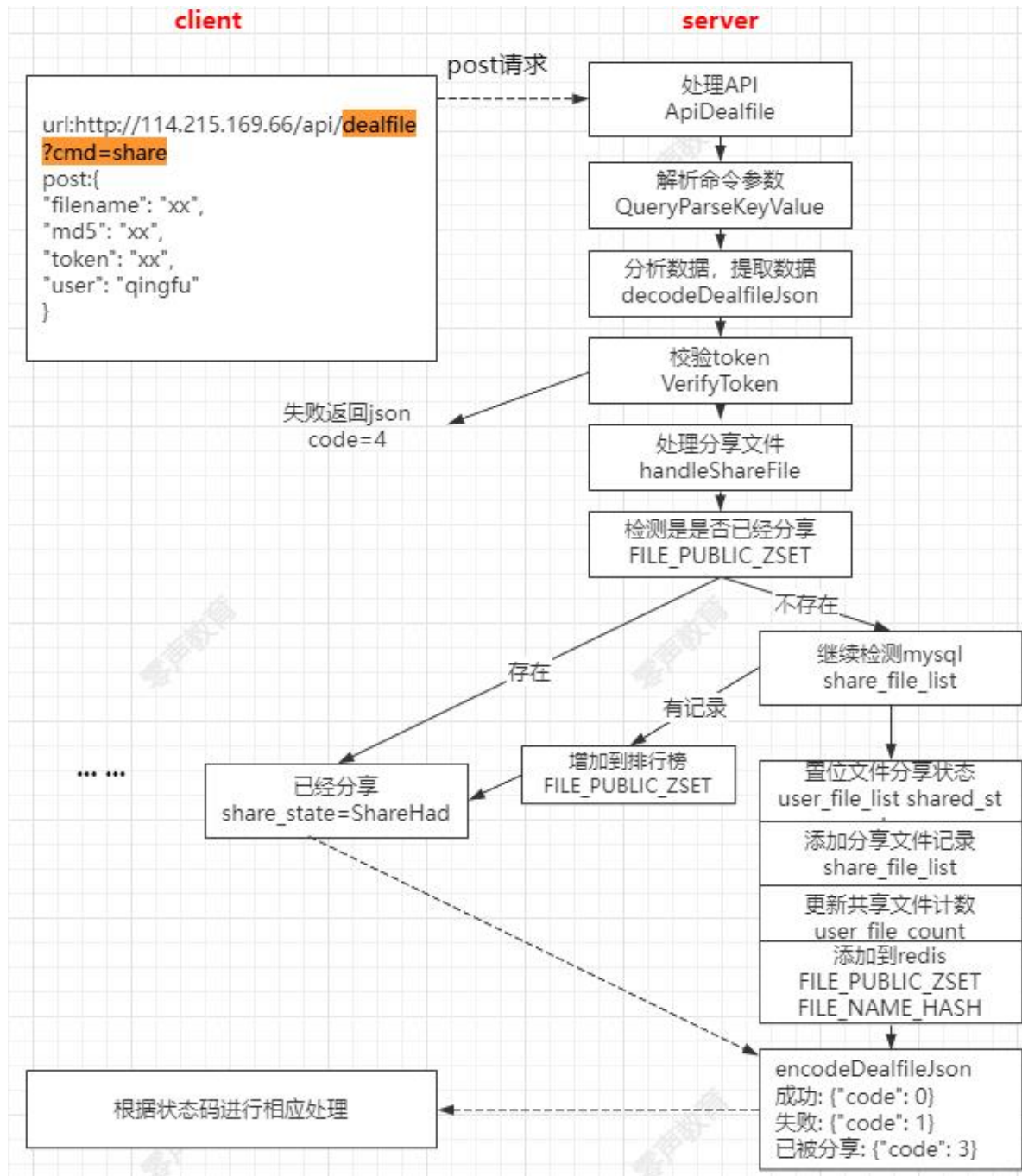
参数

```
{  
  "filename": "Makefile",  
  "md5": "602fdf30db2aacf517badf4565124f51",  
  "token": "3a58ca22317e637797f8bcad5c047446",  
  "user": "qingfu"  
}
```

返回结果

```
{  
  "code": 0  
}
```

7.1.2 处理逻辑



cmd=share

- 先在 `FILE_PUBLIC_ZSET` 判断是否已经有人分享该文件，对应的 `fieldid` 为 `md5+filename`，如果已经存在则返回结果。
- 在 `user_file_list` 将文件设置为分享状态
- 查询共享文件的数量
- 更新共享文件的数量（共享后同属于 `xxx_share_xxx_file_xxx_list_xxx_count_xxx` 用户）
- 更新 `FILE_PUBLIC_ZSET` 排行榜

- 更新 FILE_NAME_HASH 标识和对应的文件名

7.2 /api/dealfile?cmd=del 删除文件

7.2.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/dealfile?cmd=del
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	令牌	同上	必填	无
user	用户名称	不能超过 32 个字符	必填	无
md5	md5 值	md5 加密后的值	必填	无
filename	文件名称	不能超过 128 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败

服务示例

调用接口

http://114.215.169.66/api/dealfile?cmd=del

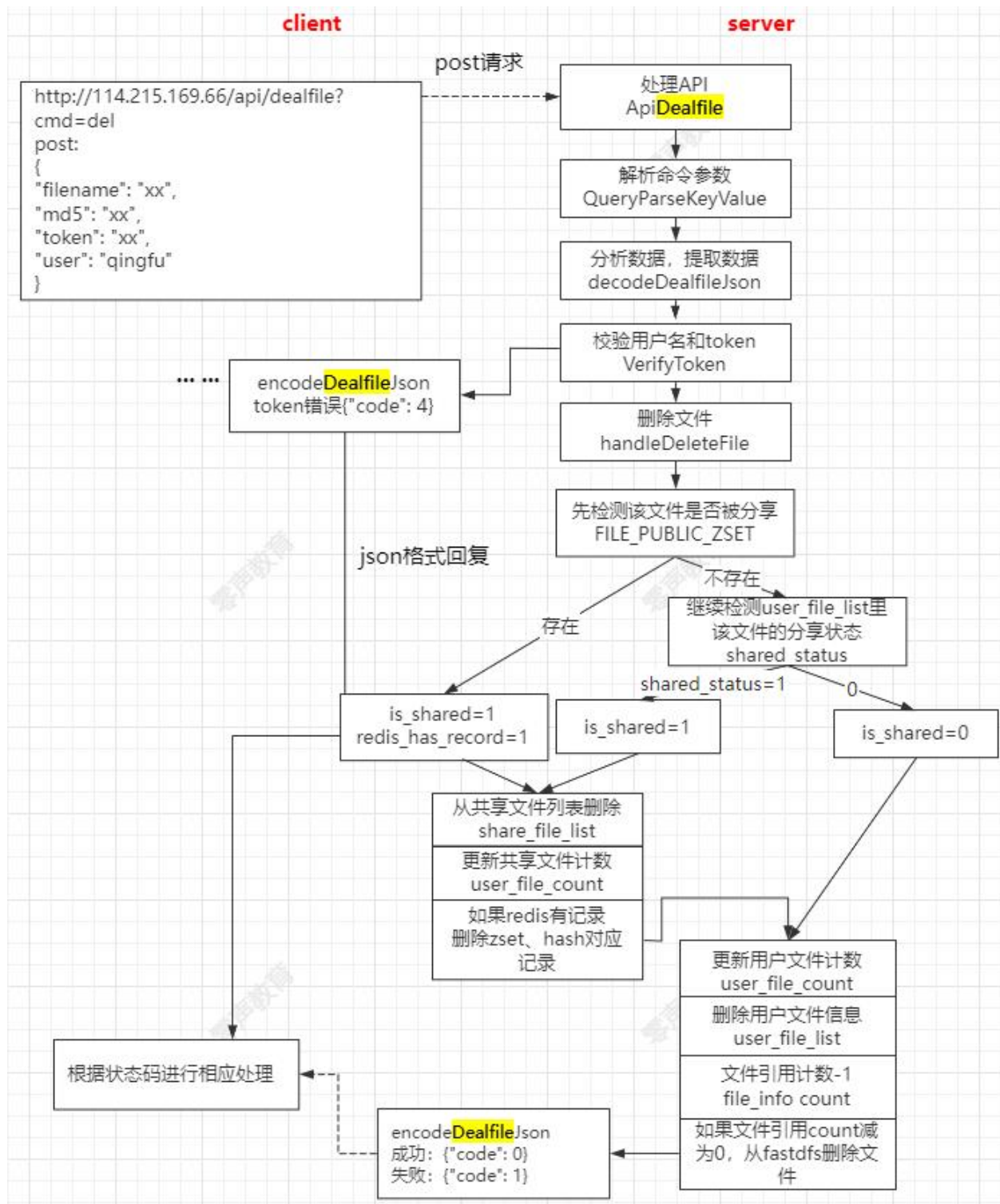
参数

```
{  
  "filename": "Makefile",  
}
```

```
"md5": "602fdf30db2aacf517badf4565124f51",  
"token": "3a58ca22317e637797f8bcad5c047446",  
"user": "qingfu"  
}  
返回结果  
{  
  "code": 0  
}
```

7.2.2 处理逻辑

零声出品 版权所有



- 先判断此文件是否已经分享
- 判断集合有没有这个文件，如果有，说明别人已经分享此文件(redis 操作)
- 如果集合没有此元素，可能因为 redis 中没有记录，再从 mysql 中查询，如果 mysql 也没有，说明真没有(mysql 操作)
- 如果 mysql 有记录，而 redis 没有记录，那么分享文件处理只需要处理 mysql (mysql 操作)
- 如果 redis 有记录，mysql 和 redis 都需要处理，删除相关记录

如果删除文件，则也将其从共享列表删除，其他人如果想要保存共享文件则可以转存到个人文件列表。

注意：

- 共享文件和我的文件 指向的是同一个文件
- 一个人共享了一个文件，删除文件的时候也会删除共享文件；两个人同时删除文件则才会删除共享文件。
- 本质上而言，删除自己的文件时是否该删除共享文件，是业务的问题，确定了业务根据业务做代码实现即可。

7.3 /api/dealfile?cmd=pv 更新文件下载计数

7.3.1 请求和应答

是用来更新指定文件的下载量，每次成功下载一个文件成功后，调用该接口更新对应文件的 pv 值。

请求 URL

URL	http://114.215.169.66/api/dealfile?cmd=pv
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	令牌	同上	必填	无
user	用户名称	不能超过 32 个字符	必填	无
md5	md5 值	md5 加密后的值	必填	无
filename	文件名称	不能超过 128 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败

服务示例

调用接口

<http://114.215.169.66/api/dealfile?cmd=pv>

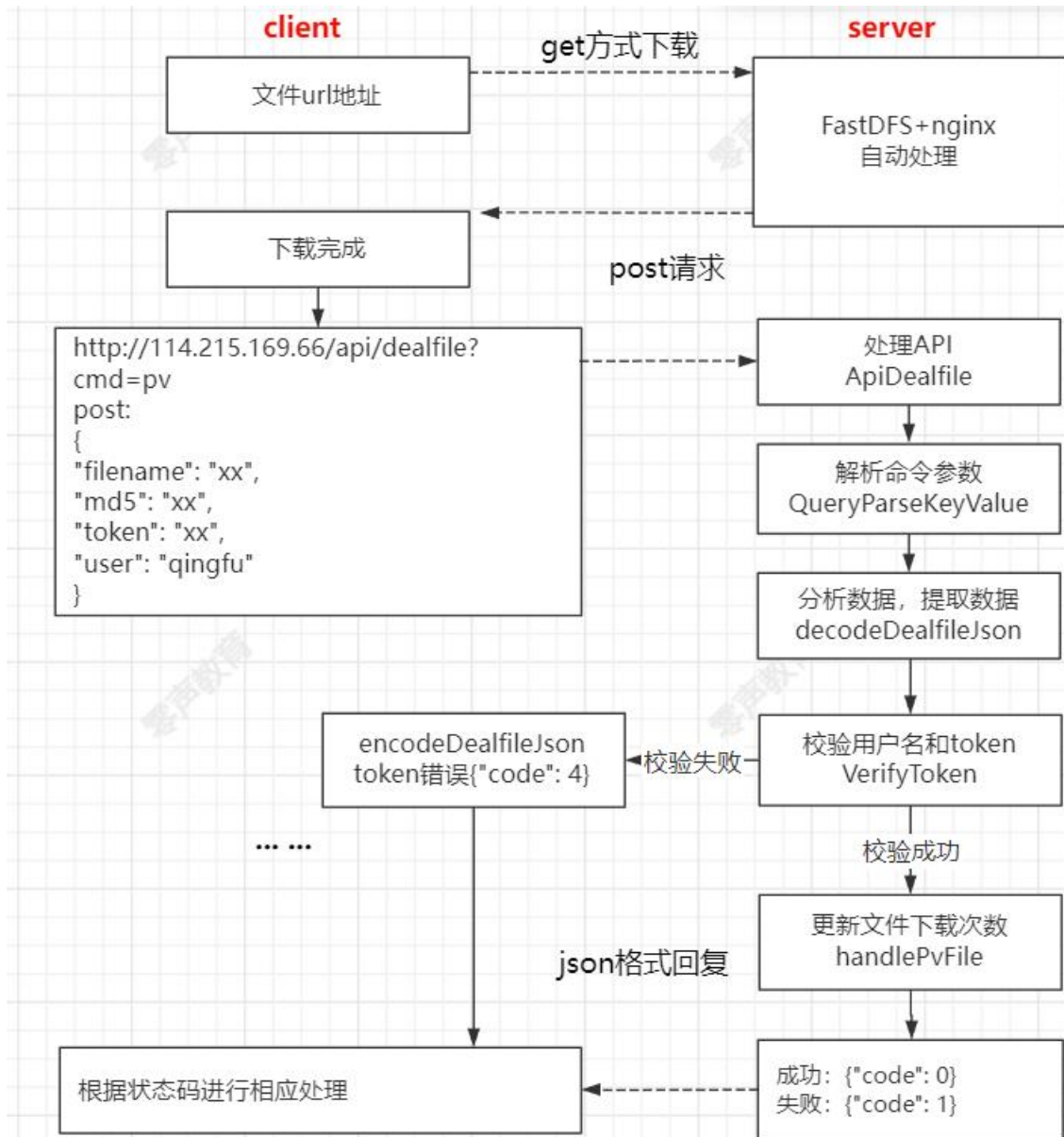
参数

```
{  
  "filename": "object_script.0voiceCloudDisk.Debug",  
  "md5": "4d41f1670e561265d08d8900a5b33550",  
  "token": "82b363791cc123e9a50d0fbf60412652",  
  "user": "qingfu"  
}
```

返回结果

```
{  
  "code": 0  
}
```


7.3.2 处理逻辑



cmd=pv
更新 user_file_list 对应的 pv 值。

8 取消分享/转存/更新下载计数 api_deal_sharefile.cc

8.1 /dealsharefile?cmd=cancel 取消分享

8.1.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/dealsharefile?cmd=cancel
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
user	用户名称	不能超过 32 个字符	必填	无
md5	md5 值	md5 加密后的值	必填	无
filename	文件名称	不能超过 128 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败

服务示例

调用接口

http://114.215.169.66/api/dealsharefile?cmd=cancel

参数

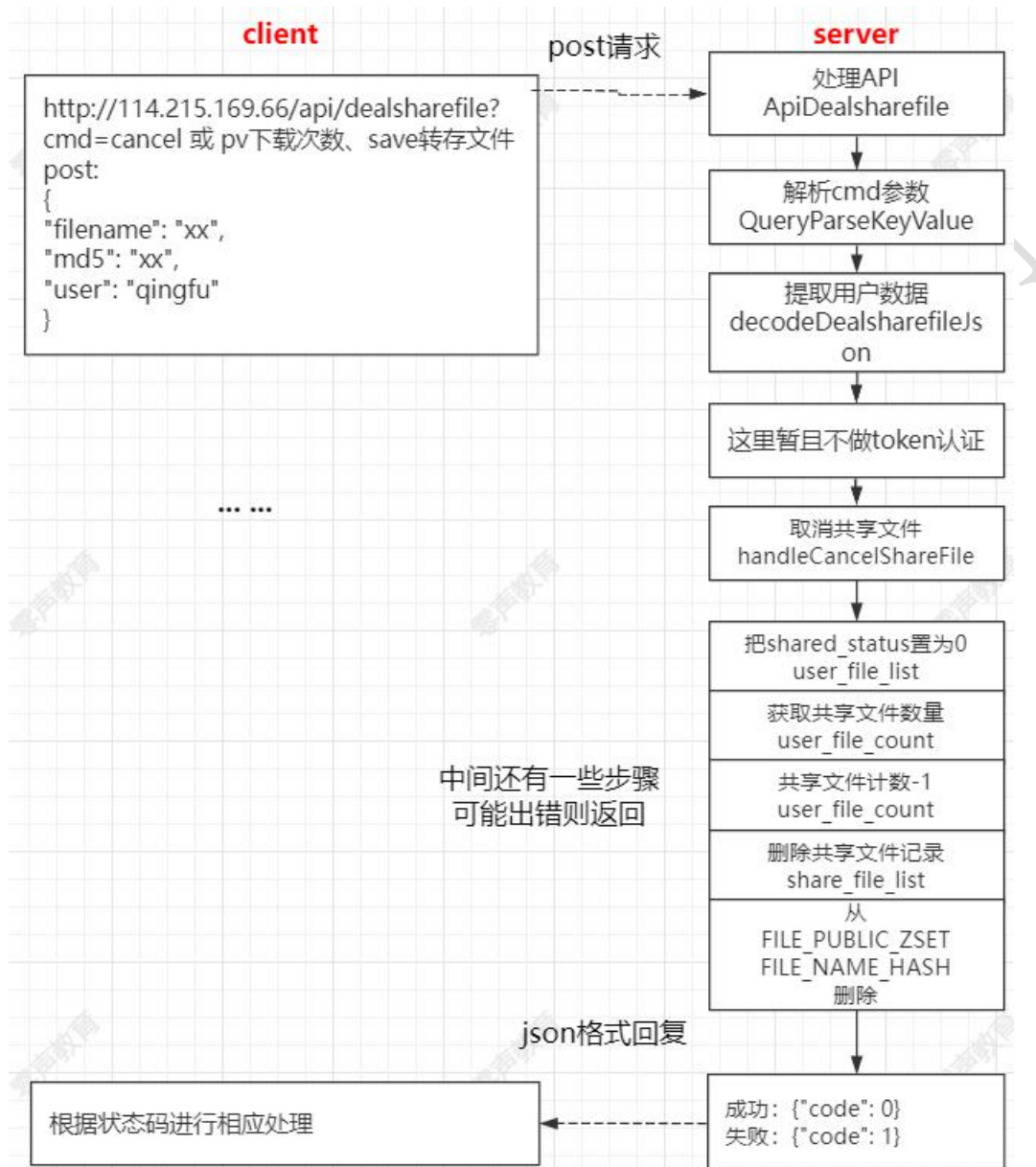
```
{  
  "filename": "ui_buttongroup.h",  
  "md5": "a89390d867d5da18c8b1a95908d7c653",  
  "user": "qingfu"  
}
```

返回结果

```
{
```

```
"code": 0
}
```

8.1.2 处理逻辑



主要逻辑:

- 查询共享文件的数量 user_file_count
- 如果共享文件数量为 1 则删除共享文件数量对应的行

```
xxx_share xxx_file xxx_list xxx_count xxx
```

- 如果共享文件数量>1，则更新共享文件数量-1

8.2 /api/dealsharefile?cmd=save 转存文件

8.2.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/dealsharefile?cmd=save
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
user	用户名称	不能超过 32 个字符	必填	无
md5	md5 值	md5 加密后的值	必填	无
filename	文件名称	不能超过 128 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败 5: 文件已存在

服务示例

调用接口

http://114.215.169.66/api/dealsharefile?cmd=save

参数

```
{  
  "filename": "ui_buttongroup.h",  
}
```

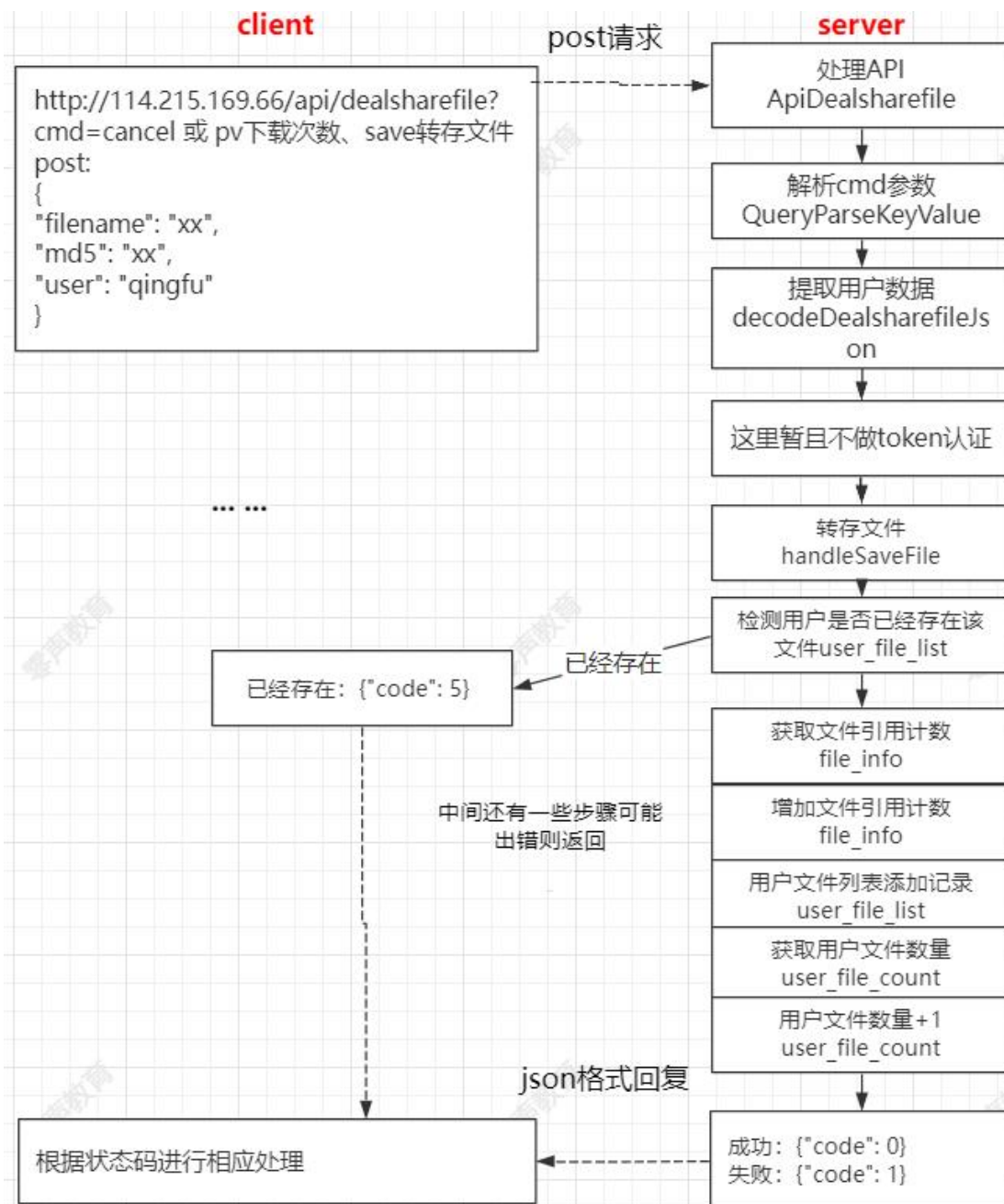
```
"md5": "a89390d867d5da18c8b1a95908d7c653",  
"user": "King"  
}  
返回结果  
{  
  "code": 0  
}
```

8.2.2 处理逻辑

<http://114.215.169.66/api/dealsharefile?cmd=save>

- 先查询是个人文件列表是否已经存在该文件。
- 增加 file_info 表的 count 计数，表示多一个人保存了该文件。
- 个人的 user_file_list 增加一条文件记录
- 更新个人的 user_file_count

当我们转存该文件后，即使分享者删除自己的文件，不会影响到我们自己转存的文件。



8.3 /api/dealsharefile?cmd=pv 更新共享文件下载次数

8.3.1 请求和应答

即是共享文件下载成功后，发该 url 的请求，该文件下次数量+1.

请求 URL

URL	http://114.215.169.66/api/dealsharefile?cmd=pv
-----	--

请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
user	用户名称	不能超过 32 个字符	必填	无
md5	md5 值	md5 加密后的值	必填	无
filename	文件名称	不能超过 128 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败

服务示例

调用接口

http://114.215.169.66/api/dealsharefile?cmd=pv

参数

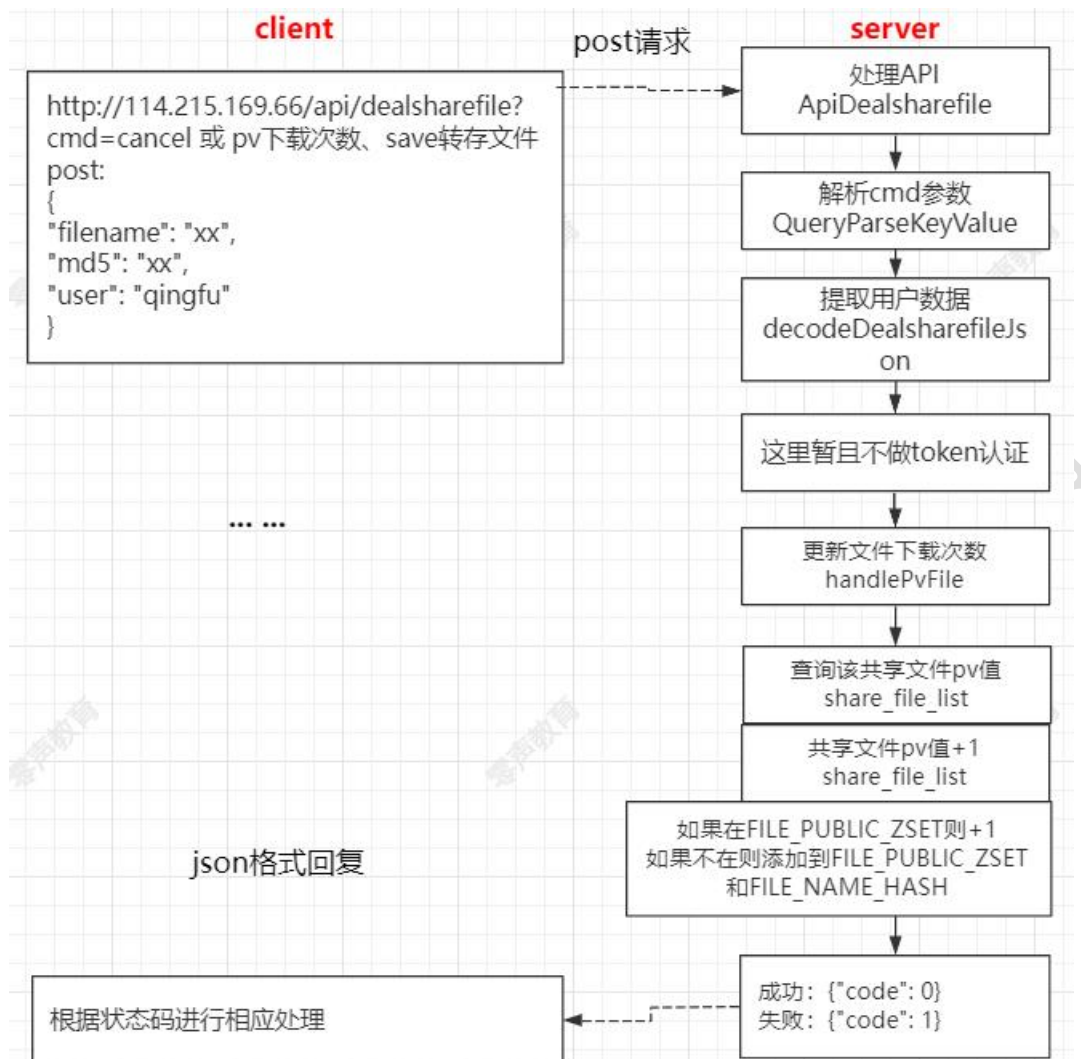
```
{  
  "filename": "ui_buttongroup.h",  
  "md5": "a89390d867d5da18c8b1a95908d7c653",  
  "user": "qingfu"  
}
```

返回结果

```
{  
  "code": 0  
}
```

8.3.2 处理逻辑

<http://114.215.169.66/api/dealsharefile?cmd=pv>



更新 `share_file_list` 的 pv 值

更新 redis 里的 `FILE_PUBLIC_ZSET`，用作排行榜

9 图床分享图片 `api_sharepicture.cc`

分享图片里的文件是个人的分享列表，文件共享则是所有人共享的列表。

`sharepicture_cgi.c` 分享后每个人都可以看到。

- 图片分享: <http://114.215.169.66/api/sharepic?cmd=share>

- 图片浏览: <http://114.215.169.66/api/sharepic?cmd=browse>
- 我的图片分享: <http://114.215.169.66/api/sharepic?cmd=normal>
- 取消图片分享: <http://114.215.169.66/api/sharepic?cmd=cancel>

9.1 /api/sharepic?cmd=share 请求图片分享

9.1.1 请求和应答

Api

URL	http://114.215.169.66/api/sharepic?cmd=share
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	令牌	同上（目前服务端暂时不处理）	必填	无
user	用户名称	不能超过 32 个字符	必填	无
md5	md5 值	md5 加密后的值	必填	无
filename	文件名称	不能超过 128 个字符	必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功 1: 失败 4: token 验证失败
urlmd5	分享图片的标识	取消分享时也要该标识

服务示例

调用接口

<http://114.215.169.66/api/sharepic?cmd=share>

参数

```
{
  "filename": "Makefile",
  "md5": "602fdf30db2aacf517badf4565124f51",
  "token": "3a58ca22317e637797f8bcad5c047446",
  "user": "qingfu"
}
```

返回结果

```
{
  "code": 0,
  "urlmd5": "602fdf30db2aacf517badf4565121234"
}
```

9.1.2 前端如何使用

(1) 访问链接: <http://114.215.169.66/api/share?urlmd5=602fdf30db2aacf517badf4565121234>, 该访问链接由 web 服务器提供。

(2) 访问链接的 web 向 后台服务器请求图片下载地址
请求接口 <http://114.215.169.66/api/sharepic?cmd=browse>

请求格式

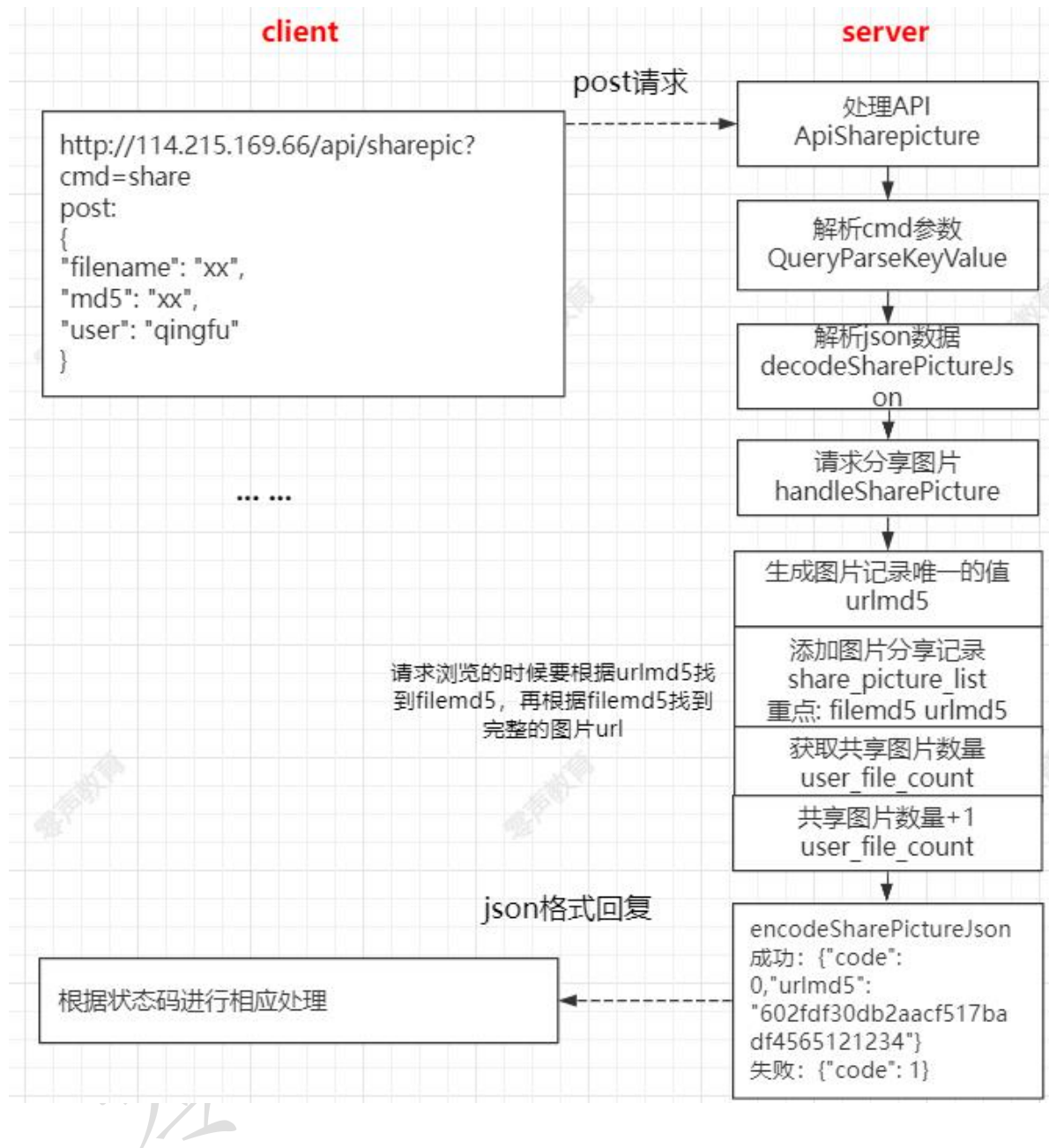
```
{
  "urlmd5": "602fdf30db2aacf517badf4565121234" // 来自请求链接
}
```

返回格式

```
{
  "code": 0,
  "url": "http://xxx.xxx.xxx.xxx/602fdf30db2aacf517badf4565121234", // 图片的下载地址
  "user": "qingfu", // 分享者用户名
  "pv": 1, // 浏览次数
  "time": "2021-12-12 11:23:00" // 分享时间
}
```

web 页面获取到 url 后下载图片显示, 并显示分享者用户名和分享时间。

9.13 处理逻辑



9.2 /api/sharepic?cmd=browse 请求浏览图片

9.2.1 请求和应答

请求接口 `http://114.215.169.66/api/sharepic?cmd=browse`，主要用来返回具体的图片下载地址。

URL	http://114.215.169.66/api/sharepic?cmd=browse
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
urimd5	分享图片的标识		必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 请求到 url 下载的 url 1: 提取码错误 2: 文件已经被删除
url	图片下载地址	
user	分享者	分享者
time	分享时间	
pv	浏览次数	

服务示例

请求格式

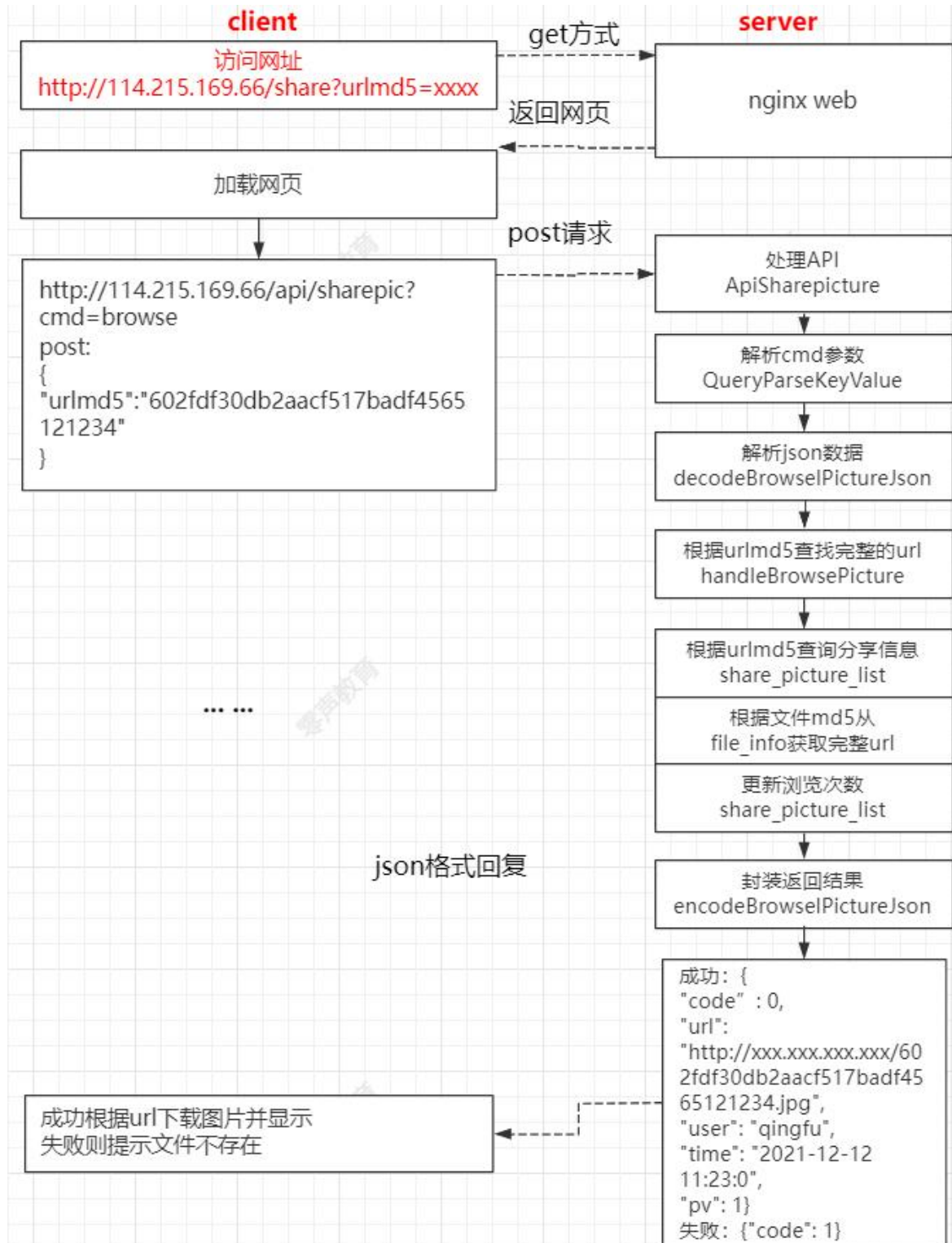
```
{  
  "urimd5": "602fdf30db2aacf517badf4565121234"  
}
```

返回格式

```
{  
  "code": 0,  
  "url": "http://xxx.xxx.xxx.xxx/602fdf30db2aacf517badf4565121234.jpg", // 图片的  
  下载地址  
  "user": "qingfu", // 分享者用户名
```

```
"time": "2021-12-12 11:23:0", // 分享时间
"pv": 1 // 浏览次数
}
```

9.2.2 处理逻辑



9.3 /api/sharepic?cmd=normal 我的图片分享

9.3.1 请求和应答

请求 URL

URL	http://114.215.169.66/api/sharepic?cmd=normal
请求方式	POST
HTTP 版本	1.1
Content-Type	application/json

请求参数

参数名	含义	规则说明	是否必须	缺省值
token	令牌		必填	无
user	用户名称	不能超过 32 个字符	必填	无
count	数量		必填	无
start	开始位置		必填	无

返回结果参数说明

名称	含义	规则说明
files	文件结果集	"code": 0: 正常; 1: 失败 "count": 2, // 分页返回数量, 如果为 0 则不需要解析 files "total": 2, 总的文件数量 "user": 用户名称, "filemd5": 文件 md5 值, "urlmd5": 图传共享时的 URL, "create_time": 创建时间, "file_name": 文件名, "pv": 文件浏览量, 浏览一次加 1 "size": 文件大小,

服务示例

调用接口

<http://114.215.169.66/api/sharepic?cmd=normal>

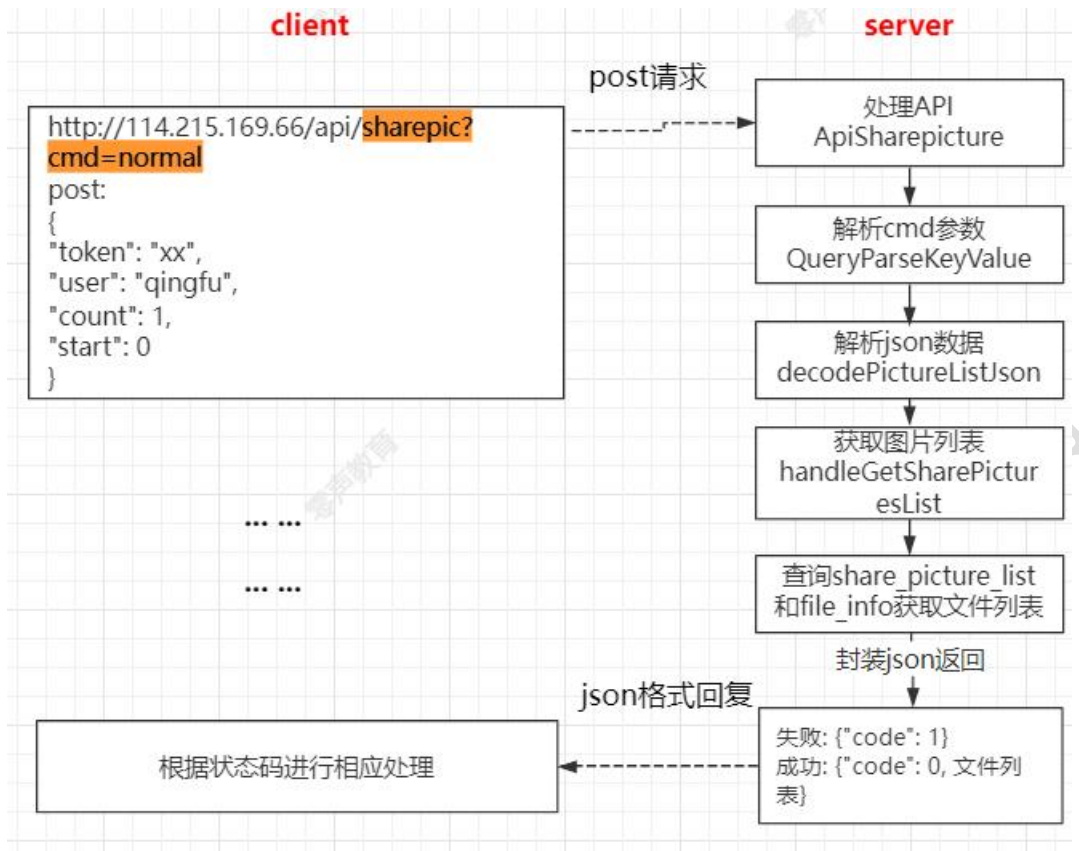
参数

```
{  
  "token": "3a58ca22317e637797f8bcad5c047446",  
  "user": "qingfu",  
  "count": 1,  
  "start": 0  
}
```

返回结果

```
{  
  "code": 0,  
  "count": 1, // 如果为 0 则不需要解析 files  
  "total": 2, 总的文件数量  
  "files": [{  
    "user": "qingfu",  
    "filemd5": "602fdf30db2aacf517badf4565124f51", // 不显示  
    "file_name": "Makefile",  
    "urlmd5": "rBAAD17_Jn6AM-iuAACBBC5AIsc2532509"  
  }],  
  "pv": 0, // 浏览量  
  "create_time": "2020-07-03 23:41:15", // 分享时间  
  "size": 33028  
}
```

9.3.2 处理逻辑



9.4 取消图片分享

9.4.1 /api/sharepic?cmd=cance 请求和应答

请求接口 `http://114.215.169.66/api/sharepic?cmd=cancel`

参数名	含义	规则说明	是否必须	缺省值
token	令牌		必填	无
urlmd5	urlmd5		必填	无

返回结果参数说明

名称	含义	规则说明
code	结果值	0: 成功

	1: 失败
--	-------

服务示例

请求格式

```
{  
  "user": "qingfu",  
  "urlmd5": "602fdf30db2aacf517badf4565121234"  
}
```

返回格式

```
{  
  "code": 0,  
}
```

9.4.2 处理逻辑

