

快盘：开放接口协议 1.0

本文档修订历史：

版本号	日期	修订说明
1.0	2011-12-20	新建
1.2	2012-02-10	增加各接口。
1.4	2012-03-01	完善流程图，勘误
1.6	2012-03-08	请求示例显示更加直观；勘误
1.7	2012-03-12	上传下载目前不支持 https；
1.8	2012-03-20	搜索；accessToken 返回 user_id；405 Not Allowed 的 FAQ
1.9	2012-03-26	shares 不支持文件夹；FAQ 关于 cookies 处理
1.10	2012-04-09	metadata 返回值 path 的说明有误
1.11	2012-04-11	create_folder 的请求示例和返回值不恰当
1.12	2012-04-12	FAQ 增加 access token 过期时间的描述
1.13	2012-05-11	移除 search；增加 history；metadata 能返回文件夹下总文件数
1.14	2012-05-18	增加接口 copy_ref；copy 接口参数作调整。
1.15	2012-07-02	规范 OAuth 的名字大小写 Oauth -> OAuth
1.16	2012-07-04	正式移除 search
1.17	2012-07-13	增加时间同步接口
1.18	2012-07-16	FAQ 第一点做稍微调整，说明 app_folder 和快盘的区别。
1.19	2012-07-19	metadata sort_type date -> time
1.20	2012-08-27	download_file 可以指定版本；增加接口 upload_file_by_id，download_file_by_id
1.21	2012-08-28	download_file_by_id 参数 fileId -> file_id
1.22	2012-08-28	upload_locate 不用 oauth 参数
1.23	2012-09-01	upload_file_by_id 不支持 overwrite 参数，默认是 overwrite=True
1.24	2012-09-06	metadata 返回 sha1 和 share_id

1. 概览

术语：

OpenAPI：快盘团队提供的开发者访问快盘资源的接口。

开发者：使用 OpenAPI 的人，可以拥有多个应用。

（第三方）应用：由 consumer_key 唯一标识，开发者使用 OpenAPI 的应用程序。

（快盘）用户：使用快盘进行文件同步、备份的人。

关系：

开发者-应用-用户 三者关系的 ER 图可以表示为：

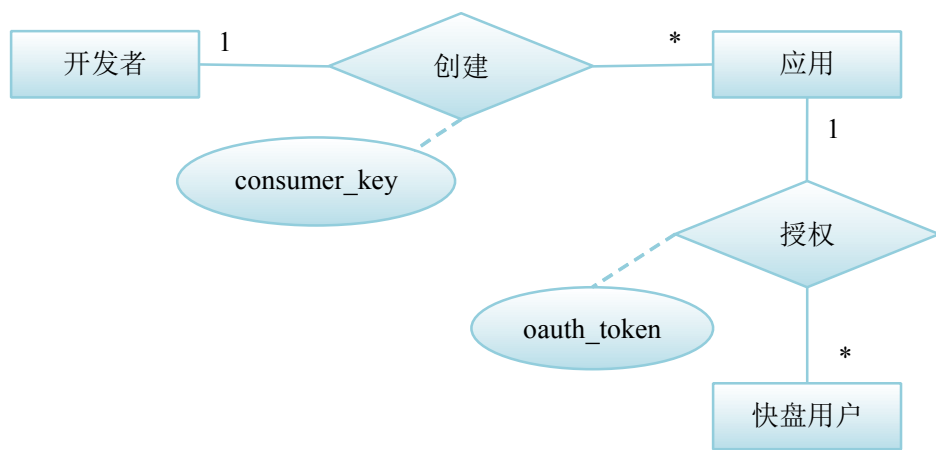
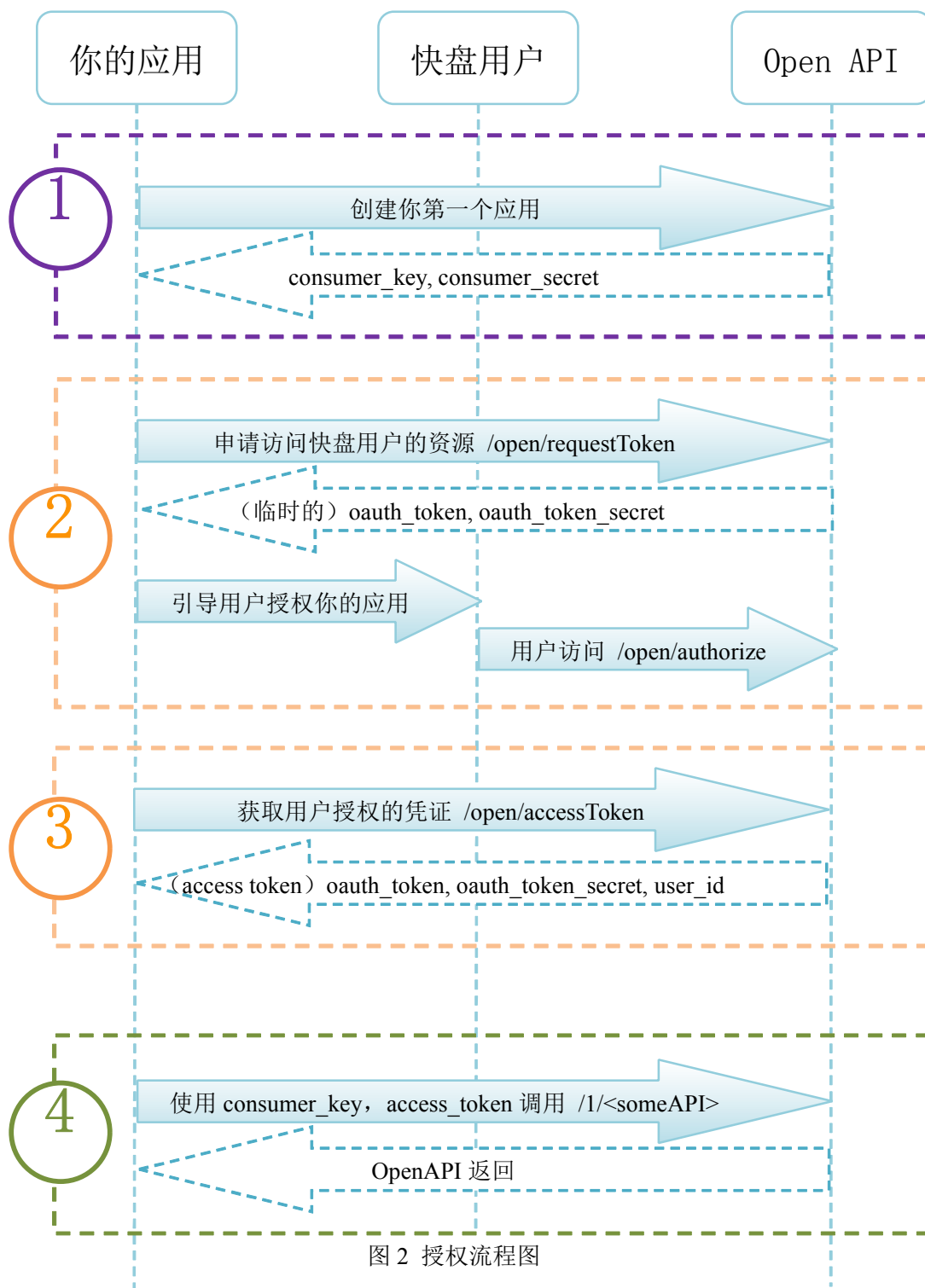


图 1 开发者-应用-用户 ER 图

2. 流程图



第 1 步创建应用请到这里: <http://www.kuaipan.cn/developers/create.htm>。

第 2-3 步属于请求授权流程, 可以参考 [OAuth 协议授权流程](#)。授权结束后, 应用可以把 access token 保存下来 (有必要的話 user_id 和 user_name 也可以保存), 直到过期才重新向用户请求授权。

第 4 步属于 OpenAPI 调用流程, 目前 v=1, 具体接口和定义可以参考 [快盘开放接口定义](#)。

3. 获取访问快盘 API 资格 (consumer_key)

访问:

<http://www.kuaipan.cn/developers/create.htm>

填写资料，创建你的应用。

4. 获取访问用户资源的资格 (**access_token**)

目前支持 OAuth1.0a 协议获取 access_token 的资格。

完成协议授权后，你将会获得 access_token、access_token_secret，和资源对应用户的 user_id、user_name。

OAuth 协议授权流程

OAuth 协议旨在为用户资源提供一个安全开放的标准。目前支持 OAuth1.0a¹。

三步走：

1. <https://openapi.kuaipan.cn/open/requestToken>
2. <https://www.kuaipan.cn/api.php>
3. <https://openapi.kuaipan.cn/open/accessToken>

分别是：

1. 获取未授权的临时 token；
2. 用户登录并授权你的应用；
3. 获取 access_token。

1. requestToken

接口说明：获取未授权的临时 token。

接口协议：**HTTPS** GET

URL 结构：<https://openapi.kuaipan.cn/open/requestToken>

接口参数：

Name	Required	Type and limit	Description
oauth_consumer_key	Y	string	第 1 步的 consumer_key
oauth_signature_method	N	string	请用“HMAC-SHA1”
oauth_signature	Y	string	本次请求的签名，生成方法参考 RFC5849 3.4 HMAC- SHA1 签名部分，或参考 附录-签名生成算法
oauth_timestamp	Y	int	时间戳，正整数， 和标准时间不超过 5 分钟
oauth_nonce	Y	string	[0-9A-Za-z_] 随机字符串，长度小于 32 字节。每次请求请使用不同的 nonce
oauth_version	N	string	请填写“1.0”
oauth_callback	N	string	回调函数

请求示例（换行和缩进只是为了更好的展示）：

¹关于 OAuth1.0a 的协议详细说明可以参考 <http://hueniverse.com/oauth/guide/>

https://openapi.kuaipan.cn/open/requestToken?

oauth_nonce=33233984&

oauth_timestamp=1331181769&

oauth_consumer_key=662d78dbbc7944a4ba7a5d491b158a0c&

oauth_signature_method=HMAC-SHA1&

oauth_version=1.0&

oauth_signature=4zGJGW3NpedYuUHULGnp8Bae98k%3D

接口返回：

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，消息体返回参数（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
oauth_token	Y	string	未授权的 token
oauth_token_secret	Y	string	对应 secret
oauth_callback_confirmed	Y	JSON boolean	True/False, callback 是否接收

JSON 示例：

```
{
  "oauth_token_secret": "b3697c3165ad473a864ae4d295003d0d",
  "oauth_token": "8519c219d45545ce8d4fb423140c9061",
  "oauth_callback_confirmed": False
}
```

2. authorize

接口说明：用户验证 token。这个接口返回是一个 web 页面，要求快盘用户登录和授权。

接口协议：**HTTPS** GET

URL 结构：

https://www.kuaipan.cn/api.php?ac=open&op=authorise&oauth_token=<YourTemporaryToken>

此页面中会要求用户登陆，然后选择同意或者拒绝对应用授权。

授权成功后：

A: web 应用会重定向到 oauth_callback 所指定的 URL（含返回参数）。

B: 客户端应用(oauth_callback 不传)会在网页中给出授权码, 用户需要手工将验证码输入到应用中才能完成授权流程。

授权地址 (引导用户授权) :

https://www.kuaipan.cn/api.php?ac=open&op=authorise&oauth_token=8519c219d45545ce8d4fb423140c9061

返回参数, 如果指定 callback, 则会把下面 2 个值 post 到指定的 callback, 否则可以跳到第三步。

Name	Required	Type and limit	Description
oauth_token	Y	string	用户授权的 token
oauth_verifier	Y	string	验证码, 用于 accessToken

3. accessToken

接口说明: 用临时 token 换取 access token。

接口协议: **HTTPS** GET

URL 结构: <https://openapi.kuaipan.cn/open/accessToken>

接口参数:

Name	Required	Type and limit	Description
oauth_consumer_key	Y	string	第 1 步的 consumer_key
oauth_signature_method	N	string	请用“HMAC-SHA1”
oauth_signature	Y	string	本次请求的签名, 生成方法参考 RFC5849 3.4 HMAC- SHA1 签名部分, 或参考 附录-签名生成算法
oauth_timestamp	Y	int	时间戳, 正整数, 和标准时间不超过 5 分钟
oauth_nonce	Y	string	[0-9A-Za-z_] 随机字符串, 长度小于 32 字节。
oauth_version	N	string	请填“1.0”
oauth_token	Y	string	临时 token
oauth_verifier	N	string	上一步返回的验证码, 为了简化用户操作, 可以不填 verifier, 但是前提是必须有用户通过了授权

请求示例 (换行和缩进只是为了更好的展示) :

https://openapi.kuaipan.cn/open/accessToken?

oauth_nonce=95000720&

oauth_timestamp=1331181774&

oauth_consumer_key=662d78dbbc7944a4ba7a5d491b158a0c&

oauth_signature_method=HMAC-SHA1&

oauth_version=1.0&

oauth_token=4e5cac43baf441cc88fba684855e248d&

oauth_signature=0wziZZcdm4IrR2v5WlksFbEiAgI%3D

接口返回：

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，消息体返回参数（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
oauth_token	Y	string	已经授权的 token
oauth_token_secret	Y	string	对应 secret
charged_dir	N	string	可见的根目录 ID
user_id	Y	string	用户 id

JSON 示例：

```
{  
  "oauth_token_secret": "e76ccab72f9f44f1a445d15b798c4fd2",  
  "oauth_token": "fe6265afa5c74866a62573b34ca411fb",  
  "user_id": 101,  
  "charged_dir": "0"  
}
```

5. 调用快盘开放接口

获得 access_token 和 access_token_secret 后，以它们作为特定用户凭证，访问特定用户资源。

生成参数列表，并生成签名，“接口相关参数”请参见具体开放接口的定义。

接口协议：**HTTP** GET

URL 结构：<http://openapi.kuaipan.cn/<v>/someapi>

接口参数

Name	Required	Type and limit	Description
oauth_consumer_key	Y	string	consumer_key
oauth_token	Y	string	access_token
oauth_signature_method	N	string	“HMAC-SHA1”
oauth_signature	Y	string	本次请求的签名
oauth_timestamp	Y	string	时间戳
oauth_nonce	Y	string	随机字符串
oauth_version	N	string	请填”1.0”
api_param_1			
api_param_...
api_param_N			

请求示例（换行和缩进只是为了更好的展示）：

https://openapi.kuaipan.cn/1/someapi?

oauth_nonce=95000720&

oauth_timestamp=1331181774&

oauth_consumer_key=662d78dbbc7944a4ba7a5d491b158a0c&

oauth_signature_method=HMAC-SHA1&

oauth_version=1.0&

oauth_token=4e5cac43baf441cc88fba684855e248d&

oauth_signature= %2BxicZMKhzINyxq5tOTdAZgHiNyk%3D&

<api_param_1>=<value1>&

<api_param_2>=<value2>&

...

<api_param_N>=<valueN>

接口返回:

返回 HTTP 的状态码含义, 请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200, 消息体返回参数 (当前只支持返回 JSON 化字符串):

Name	Required	Type and limit	Description
api_rtn_param_1			
api_rtn_param_...			
api_rtn_param_N			

示例:

```
{
  "api_rtn_param_1": "oxoxoxoxoxoxoxoooo",
  ... .. ,
  "api_rtn_param_.....": "oxoxoxoxoxoxoxoooo",
  ... .. ,
  "api_rtn_param_N": "oxoxoxoxoxoxoxoooo"
}
```

其 HTTP 状态码不为 200, 消息体返回参数。Msg 含义请参考[附录 MSG 定义](#)。

Name	Required	Type and limit	Description
msg	Y	String	其它描述, 视特定的接口而定

示例:

```
{
```

```
"msg": "this is why http response status' not 200"  
}
```

6. FAQ

- 创建应用时候，访问权限选择里面的 **整个快盘** 和 **应用文件夹** 有什么不同？
整个快盘 – 调用接口的时候 `root=kuaipan`。你的应用可以访问和操作用户快盘里面所有文件，一般情况下我们不建议你们申请这种方式，也**比较难通过审核**。
应用文件夹 – 调用接口的时候 `root=app_folder`。你的应用只可以访问和操作用户快盘 / 我的应用 / <你的应用名> （如/我的应用/快盘-android）的文件，类似沙盒的环境，用户的其他数据会得到保护，安全性和保密性都很好。
- 为什么不提供直接用快盘用户的账号密码直接登录的接口？
快盘 OpenAPI 可以由任何个人开发者使用，假如提供用快盘用户密码登录的授权验证方式，会带来 2 个问题：
从拥有资源的快盘用户的角度上看，把自己快盘的账号密码交给不被信任的第三方应用，有泄露的风险；
同时，假如用户某天想取消对某一第三方应用的授权，只能通过修改密码，这是一个用户体验很糟糕的方式。
OAuth 协议就是为了解决上述问题而存在的授权协议。它可以在用户不告知应用密码的情况下，让应用能访问到自己的快盘文件。
- OAuth 授权第三步（/accessToken）的时候，怎样获得那个 `oauth_verifier`？
`verifier` 现在是一个可选参数，只要用户在第二步的时候授权了，就可以不传 `verifier` 而获得 `access token`。
- 我想用 HTTPS，进一步提高安全性？
除了上传下载，其他接口都支持 HTTPS。
但是有时候为了减少延迟，部分接口可以使用 HTTP 协议。接口定义里面写的方法只是建议值。
- 总是返回 401？
检查以下情况：
`msg="request expired"` 程序本地的系统时间是否正确；
`msg="reused nonce"` `oauth_nonce` 是否每次都不一样，或者该 URL 是否被调用了 2 次或以上；
`msg="bad signature"` 可以参考 <http://www.kuaipan.cn/developers/createsign.htm> 对比工具和自己生成的有什么不一样。
- 上传文件时候返回 405 Not Allowed？
 1. 检查下 `path` 是否带有文件名，如 `/abd/test.txt`；
 2. 第一点满足的时候，检查一下文件所在的路径是否存在，如上述例子用的 `/abd` 这个文件夹是存在的；
 3. 确保 `path`（如 `/abd/test.txt`）不是指向一个文件夹；
 4. 检查上传的 URL 没有写错；

5. 如上述情况都满足但仍然有错误，可以把相关 HTTP 包的完整信息截取，到快盘开发者论坛上发帖子提问。

- 为什么 申请生产使用 的时候会失败？
对于第三方应用，必须先经过开发者自己测试后才能给用户使用。
因此应用在从来没有得到任何用户授权的时候，不能申请生产使用。
注意的是，处于开发状态（development）的应用，只能用自己的创建应用时候的快盘帐号来授权。
- 为什么不能授权 /open/authorize 或 /open/accessToken？
对于开发中的应用（development）只能授权给创建应用时候使用的快盘帐号。
- 使用 OpenAPI 上传的文件是占用谁的空间？
快盘用户的。
- 不提供云存储服务（类似 S3）吗？
目前快盘只为个人开发者提供访问用户文件的 API，没有提供基础存储服务。
企业开发如有需要，请和我们 kuaipan_bd@kingsoft.com 取得联系。
- /open/requestToken 时候的回调函数（callback）有什么用？
callback 是用户访问/open/authorize 登录并授权后，快盘服务器会返回一个 302 重定向到用户的浏览器（或者其他 User-Agent），重定向的内容是把（临时的）oauth_token，oauth_verifier 以 POST 的方式到应用指定的 callback，假如应用是 web 类型的应用（有自己的 HTTP 服务器），就能在第一时间知道授权已经成功。对本地应用（如 PC、手机）来说，callback 一般没有作用，可以不传。
- 下载文件的时候报 403 错误，但是把 URL 放到 IE 等浏览器里面能下载？
第一次打开下载 URL 的时候有时会带一个 HTTP 302 重定向，某些时候还会有 cookies，在编写 HTTP Client 的时候，需要处理好重定向，并且把 cookies 也带上；

Java 程序在处理 302 跳转地址遇到中文编码问题的时候，可以参考这篇帖子
<http://bbs.kuaipan.cn/viewthread.php?tid=20131> ；

假如使用 cURL 库，加上以下参数就能自动处理跳转了：

```
setopt(FOLLOWLOCATION, 1)  
setopt(COOKIEFILE, "cookie_file_name")
```

- Access Token 什么时候过期？
一般情况下 access token 过期时间为 1 年，但是用户可以随时取消授权，取消授权后 token 会立刻过期，这时候用这个 token 调用接口服务器会返回 authorization expired。，客户端应该把保存在本地的 access token 删除，并重新执行授权流程。

7. 快盘开放接口定义（v=1）

❖ 编程建议

1. 目前版本 <v> 为 1;
2. 没有特别说明下，所有接口，参数都是大小写敏感的;
3. 时间的默认格式是 %Y-%m-%d %H:%M:%S ， (UTC+08:00);
4. 所有接口都支持 HTTPS 协议,在对安全性要求不高的情况下，部分接口可以用使用 HTTP 协议;
5. 在本地做好缓存，可以减小接口网络调用次数，加快响应时间以达到更好的用户体验;
6. 原则上，对于同一应用和快盘用户，只需用调用一次 accessToken，然后在应用服务器或者本地保存 access_token，access_token_secret 和 charged_dir ，直到 access_token 授权过期（返回 authorization expired，详见[附录-MSG 定义](#)）;
7. 接口参数说明中，required 字段为 N 或者 False 的参数可以不传;
8. 接口返回说明中，required 字段为 N 或者 False 的参数不一定会返回；在未来的更新升级中，也可能会返回没有在说明中出现的字段，应用设计中要做好对应这些和说明不符的异常的兼容准备;
9. 接口返回说明中，uint64 类型的字段，部分 JSON 解析器可能由于精度问题不能获取准确值（如 JS 的解析器只能精确到 52 位），因此建议遇到此类型的也作为 string 类型来处理;
10. 接口返回中，我们会返回 JSON object 类型的对象，由于服务升级，或者维护需要，可能会出现文档中没有描述的返回值或者字段，或者由于服务器故障返回不是 JSON 格式的数据，在开发过程中应该注意，并且做好兼容准备;
11. 请不要使用文档中没提及或建议的返回值或者参数，这个可能是服务器的 bug，可能在未来升级中会失效;
12. 在 HTTP 返回的 status code 不是 200，5xx 的时候，大部分情况下会返回一个包含 msg 字段的 JSON 字符串，说明出现错误的原因，这对调试有很大帮助;
13. 申请应用时，您选择的目录访问权限会直接影响以后的接口使用。如果您选择的是 整个文件夹 访问，那么以后 root=kuaipan 或者 app_folder 是没有区别的；如果您选择的是 应用目录 ，那么在以后的接口调用中，只能使用 root=app_folder，否则会返回 403 forbidden。

❖ 时间同步接口

接口说明：查看当前服务器的时间，以时间戳的形式表示。部分客户端当前时间可能不准确，会导致 timestamp 参数不在服务器接受范围内，这时服务器会返回 401 request expired。先进行时间同步可以避免此问题。

接口协议：**HTTP GET**
URL 结构：http://openapi.kuaipan.cn/open/time
接口参数

Name	Required	Type and limit	Description

请求示例（换行和缩进只是为了更好的展示）：

```
http://openapi.kuaipan.cn/open/time
```

接口返回：
返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。
若 HTTP 状态码为 200，消息体返回参数（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
Timestamp	Y	String	当前服务器的时间戳。
Encoding	Y	String	数据编码。
Name	Y	string	服务名称
OAuth version	Y	String	OAuth 协议版本

JSON 示例：
{
 "OAuth version": "1.0a",
 "Timestamp": "1342144313",
 "Name": "Kuaipan open API",
 "Encoding": "UTF-8"
}
若 HTTP 状态码不为 200，Msg 含义请参考[附录 MSG 定义](#)。

❖ /account_info

接口说明：查看用户信息

接口协议：**HTTP** GET

URL 结构：http://openapi.kuaipan.cn/<v>/account_info

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>

请求示例（换行和缩进只是为了更好的展示）：

```
http://openapi.kuaipan.cn/1/account_info?

    oauth_nonce=20134261&

    oauth_timestamp=1326186932&

    oauth_consumer_key=79a7578ce6cf4a6fa27dbf30c6324df4&

    oauth_signature_method=HMAC-SHA1&

    oauth_version=1.0&

    oauth_token=fa361a4a1dfc4a739869020e586582f9&

    oauth_signature=n3%2B%2F%2FJ7QbIoc%2BLVAFp5TSKkpc5Y%3D
```

接口返回：

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，消息体返回参数（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
user_id	Y	int32	标识用户的唯一 id
user_name	Y	string	用户名字
max_file_size	Y	int32	允许上传最大文件
quota_total	Y	uint64	用户空间总量，Byte
quota_used	Y	uint64	用户空间已使用量，Byte
quota_recycled	N	uint64	用户空间的回收站空间使用量，Byte

JSON 示例：

```
{
    "max_file_size": 314572800,
```

```
"user_name": "test_open@126.com",  
"quota_used": 7250,  
"quota_total": 5368709120,  
"user_id": 101  
}
```

若 HTTP 状态码不为 200，Msg 含义请参考[附录 MSG 定义](#)。

❖ /metadata

接口说明：获取单个文件、文件夹信息

接口协议：**HTTP** GET

接口建议：返回的字段会较多，您只需获取自己需要的，并且能兼容一些额外的返回值。

URL 结构：<http://openapi.kuaipan.cn/<v>/metadata/<root>/<path>>

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
list	N	string/JSON bool	默认 True。 当路径指向是文件夹时，会返回其包含的子文件
file_limit	N	int	默认并且最大为 10000，在查询文件夹信息时，如果文件夹下的文件数目超过 file_limit 时，不返回文件列表，而是返回 406 (too many files)
hash	N	char[32]	Hash 为服务器计算的文件列表的哈希值，如果不传本值，那么获取<path>下的最新文件列表。如果传本值，如果文件列表并未更新，则返回 304 (Not modified)。 本特性暂不支持。请不要使用
include_deleted	N	bool	是否包含删除文件。 本特性暂不支持。请不要使用
page	N	int	默认 0 - 不分页； 和 page_size 一起使用； 从 1 开始，表示显示第 N 页的文件
page_size	N	int	默认 20； page 不等于 0 才有效； 表示每页显示的文件内容。
filter_ext	N	string	过滤特定扩展名的文件，用英文半角逗号分隔，只能是 ascii 字符，每一项的长度不超过 5，最大长度不超过 64。 如： jpg,bmp,png,jpeg 会返回包含上述扩展的文档（如 1.JPG）。 文件夹不会被过滤。

			请用小写。
sort_by	N	enum(time, name, size,)	默认空，不排序。 在分页（ page!=0 ）的时候才有效。 排序依据， time - 修改日期 name - 文件名 size - 文件大小 在前面加上字母"r"的时候代表倒序，如： sort_by=rsz 表示从大到小排序。

请求示例（换行和缩进只是为了更好的展示）：

```

http://openapi.kuaipan.cn/1/metadata/app_folder/?

    oauth_nonce=54775796&

    oauth_timestamp=1326186932&

    oauth_consumer_key=79a7578ce6cf4a6fa27dbf30c6324df4&

    oauth_signature_method=HMAC-SHA1&

    oauth_version=1.0&

    oauth_token=fa361a4a1dfc4a739869020e586582f9&

    oauth_signature=I0hQbk9AweHOZutR7yGOeTVtFPQ%3D

```

接口返回：

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，消息体返回参数（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
path	Y	string	文件或文件夹相对<root>的路径
root	Y	string	kuaipan 或 app_folder
hash	N	char[32]	list=True 才返回； 当前这级文件夹的哈希值。
file_id	N	string	path=/, root=kuaipan 时不返回。 文件唯一标识 id
type	N	enum(file, folder)	path=/, root=kuaipan 时不返回。 folder 为文件夹，file 为文件，
size	N	uint32	path=/, root=kuaipan 时不返回。

			文件大小
create_time	N	string	path=/, root=kuaipan 时不返回。 YYYY-MM-DD hh:mm:ss
modify_time	N	string	path=/, root=kuaipan 时不返回。 YYYY-MM-DD hh:mm:ss
name	N	string	path=/, root=kuaipan 时不返回。 文件名
rev	N	string	path=/, root=kuaipan 时不返回。
is_deleted	N	string/JSON boolean	path=/, root=kuaipan 时不返回。 是否被删除的文件
sha1	N	string	path=/, root=kuaipan 时不返回。 文件数据的 sha160 值。
share_id	N	string	path=/, root=kuaipan 时不返回。 假如非 0, 则表示它是一个共享的文件。
files	N	file	list=true 才返回。
files/file_id	Y	string	文件唯一标识 id
files/type	Y	enum(file, folder)	folder 为文件夹, file 为文件,
files/size	Y	uint32	文件大小
files/create_time	Y	string	YYYY-MM-DD hh:mm:ss
files/modify_time	Y	string	YYYY-MM-DD hh:mm:ss
files/name	Y	string	文件名
files/rev	N	string	
files/is_deleted	Y	string/JSON boolean	是否被删除的文件
files/sha1	Y	string	文件数据 sha160 值
files/share_id	N	string	假如非 0, 则表示它是一个共享的文件。
files_total	N	int	list=true 才返回。该文件夹下的文件总数。

示例:

其 HTTP 状态码不为 200, 消息体返回参数 msg, msg 含义请参考[附录 MSG 定义](#)。

```
{
  "files":
  [
    {
      "is_deleted": False,
      "name": "\u6211\u7684\u5e94\u7528",
      "rev": "1",
      "file_id": "433791696897",
      "create_time": "2012-01-05 14:18:15",
```

```
        "modify_time": "2012-01-05 14:18:15",
        "type": "folder",
        "size": 0
    },
    {
        "is_deleted": False,
        "name": "1",
        "rev": "1",
        "file_id": "433791697161",
        "create_time": "2012-01-10 14:52:28",
        "modify_time": "2012-01-10 14:52:28",
        "type": "folder",
        "size": 0
    }
],
"files_total": 2,
"path": "/",
"root": "kuaipan",
"hash": "ac00000065"
}
```

❖ /shares

接口说明：创建并获取一个文件的分享链接。**发布外链的文件会接受审核。**

接口协议：**HTTP** GET

URL 结构：<http://openapi.kuaipan.cn/<v>/shares/<root>/<path>>

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
name	N	string	默认本文件名。 在下载页面显示的名字。
access_code	N	String	提取码，若需要则输入此值， [a-zA-Z]，6-10 位。

接口返回：

Name	Required	Type and limit	Description
url	Y	string	外链下载页面
access_code	N	string	提取码

示例：

```
{"url": "http://www.kuaipan.cn/file/id_433791697161.html"}
```

❖ /history

接口说明：文件的历史版本

接口协议：**HTTP** GET

URL 结构：<http://openapi.kuaipan.cn/<v>/history/<root>/<path>>

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
-	-	-	-

接口返回：

Name	Required	Type and limit	Description
files	Y	string	历史版本记录，list 对象
files/file_id	Y	string	文件 id
files/rev	Y	string	文件版本
files/create_time	Y	string	YYYY-MM-DD hh:mm:ss 该版本被覆盖的时间。

示例：

MSG:

Message	Description
file not exist	文件不存在，或没有历史版本

❖ /copy_ref

接口说明：产生一个复制引用（ref），别的帐号可以使用这个引用作为 copy 接口的参数来复制文件。复制的版本是文件的最新版本，与引用产生的时间无关。

接口协议：**HTTP** GET

URL 结构：http://openapi.kuaipan.cn/<v>/copy_ref/<root>/<path>

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>

示例：

接口返回：

Name	Required	Type and limit	Description
copy_ref	Y	string	复制引用，可以作为 copy 接口的参数。
expires	Y	string	YYYY-MM-DD hh:mm:ss 引用过期时间。

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，说明移动成功。

若 HTTP 状态码不为 200，消息体返回 msg，msg 含义请参考[附录 MSG 定义](#)。

❖ /fileops/create_folder

接口说明：新建文件夹

接口协议：**HTTP** GET

URL 结构：http://openapi.kuaipan.cn/<v>/fileops/create_folder

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
root	Y	string	app_folder 或 kuaipan
path	Y	string，utf8 编码字符，长度不超过 255，其所在绝对路径长度也不能超过 255	相对于 root 的路径，包含文件名

请求示例（换行和缩进只是为了更好的展示）：

```
http://openapi.kuaipan.cn/1/fileops/create_folder?  
  
  oauth_version=1.0&  
  
  oauth_signature=dds2TuW6i433uhBxxLrE6HG%2FtRY%3D&  
  
  oauth_token=fe6265afa5c74866a62573b34ca411fb&  
  
  oauth_signature_method=HMAC-SHA1&  
  
  oauth_nonce=65352566&  
  
  oauth_timestamp=1329119724&  
  
  path=%2F%E6%B5%8B1329119724.29&  
  
  oauth_consumer_key=xcKXXgo9AkXdJuBn&  
  
  root=app_folder
```

接口返回：

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，消息体返回参数（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
path	N	string	文件或文件夹相对<path>的路径

file_id	Y	string	
root	N	string	kuaipan 或 app_folder

JSON 示例:

```
{
  "msg": "ok",
  "path": "u/^u6d4b1329119724.29",
  "root": "app_folder",
  "file_id": "433791702966"
}
```

若 HTTP 状态码不为 200，消息体返回 msg，msg 含义请参考[附录 MSG 定义](#)。

❖ /fileops/move

接口说明：移动文件，文件夹，不能移动带共享的文件，当然也不支持其他不能移动的情况（如形成环路）

接口协议：**HTTP** GET

URL 结构：<http://openapi.kuaipan.cn/<v>/fileops/move>

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
root	Y	string	app_folder 或 kuaipan
from_path	Y	string	相对于 root 的旧文件路径
to_path	Y	string	相对于 root 的新文件路径，包含文件名（和 from_path 的文件不同名的话，则重命名）

示例：

接口返回：

Name	Required	Type and limit	Description
-			

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，说明移动成功。

若 HTTP 状态码不为 200，消息体返回 msg，msg 含义请参考[附录 MSG 定义](#)。

❖ /fileops/copy

接口说明：复制文件，文件夹

接口协议：**HTTP** GET

URL 结构：<http://openapi.kuaipan.cn/<v>/fileops/copy>

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
root	Y	string	app_folder 或 kuaipan
from_path	N	string	相对于 root 的旧文件(夹)路径
to_path	Y	string	相对于 root 的新文件夹路径，包含文件名（和 from_path 的文件不同名的话，则重命名）
from_copy_ref	N	string	使用 copy_ref 接口返回的引用

请求示例（换行和缩进只是为了更好的展示）：

http://openapi.kuaipan.cn/1/fileops/copy/?

oauth_version=1.0&

oauth_signature=pJxLJfvyfW0zXC%2BVtrupWXpFSJ8%3D&

from_path=%2F%E5%A4%8D%E5%88%B6%2Ftest.wps&

oauth_token=aed1a2268d2544b095bf1a88402be494&

oauth_signature_method=HMAC-SHA1&

oauth_nonce=45735647&

oauth_timestamp=1331183490&

to_path=%2F%E5%A4%8D%E5%88%B6%2Ftest%EF%BC%88%E5%A4

%8D%E4%BB%B6%EF%BC%89.wps&

oauth_consumer_key=662d78dbbc7944a4ba7a5d491b158a0c&

root=app_folder

接口返回：

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，消息体返回参数的（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
file_id	Y	string	

JSON 示例：

```
{"file_id": "433791697421"}
```

若 HTTP 状态码不为 200，消息体返回 msg，msg 含义请参考[附录 MSG 定义](#)。

❖ /fileops/delete

接口说明：删除文件，文件夹，以及文件夹下所有文件到回收站。**删除有风险，操作需谨慎。**

接口协议：**HTTP** GET

URL 结构：<http://openapi.kuaipan.cn/<v>/fileops/delete>

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
root	Y	string	app_folder 或 kuaipan
path	Y	string	相对于 root 将被删除的文件(夹)路径
to_recycle	N	string/JSON boolean	默认 True True- 放入回收站，空间不回收； False - 彻底删除，空间相应增加。

请求示例（换行和缩进只是为了更好的展示）：

```
http://openapi.kuaipan.cn/1/fileops/delete?

  oauth_version=1.0&

  oauth_signature=YK0bHhs31yeLJiYLQqI2Scq3%2BJo%3D&

  oauth_token=fe6265afa5c74866a62573b34ca411fb&

  oauth_signature_method=HMAC-SHA1&

  oauth_nonce=41998409&

  oauth_timestamp=1329120913&

  path=%2F%E5%88%A0%E9%99%A4%25E4%25BD%25A0%27%E6%B5
%8B%E8%AF%95%CE%97%CE%B6%E4%B8%93%E7%94%A81329120912
%2F%2Fd&

  oauth_consumer_key=xcXXXgo9AkXdJuBn&

  root=app_folder
```

接口返回：

Name	Required	Type and limit	Description
-			

返回 HTTP 的状态码含义，请参考 [附录 HTTP 状态码](#)。

若 HTTP 状态码为 200，说明删除成功：

❖ /fileops/thumbnail

接口说明：缩略图

接口协议：**HTTP** GET

接口 URL：<http://conv.kuaipan.cn/1/fileops/thumbnail>

接口参数：

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
width	Y	int	宽
height	Y	int	高
root	Y	string	kuaipan 或者 app_folder
path	Y	string	相对于 root 的路径，图片文件的位置

支持格式：

TYPE_IMG = ('gif', 'png', 'jpg', 'bmp', 'jpeg', 'jpe')

接口返回：

若 HTTP 状态码不是 200，HTTP status code 和 msg 的可能值(JSON 格式)：

status	msg	Description
404	file not exist	文件不存在
401	-	授权失败，参考 附录 MSG 定义- 401 类
302	-	文件在另外的服务器上，请处理好跳转
403	-	授权失败
5xx		服务器暂时不可用

若 HTTP 状态码为 200，消息体即文件内容。

Name	Required	Type and limit	Description
file	Y	binary	文件内容

❖ /fileops/documentView

接口说明：文档转换，

接口协议：**HTTP** GET

接口 URL：<http://conv.kuaipan.cn/1/fileops/documentView>

接口参数：

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
type	Y	enum()	文档的类型，支持格式如下： 'pdf', 'doc', 'wps', 'csv', 'prn', 'xls', 'et', 'ppt', 'dps', 'txt', 'rtf'
view	Y	enum()	视图格式，下面的参数会影响在浏览器显示的效果（页面大小和排版）： 'normal' - 假如是 PC 浏览器的话请选择这个； 'android' - android 客户端请选择这个； 'iPad' - iPad 客户端的请选择这个； 'iphone' - iphone 客户端。
zip	N	enum(0, 1)	默认 0 - 不压缩，返回的是一个 html 页面； 1 - 返回一个 zip 的压缩包，里面包含文档转换后的 html 页面。
root	Y	string	kuaipan 或者 app_folder
path	Y	string	相对于 root 的路径，要转换的文档的位置

支持格式：

('pdf', 'doc', 'wps', 'csv', 'prn', 'xls', 'et', 'ppt', 'dps', 'txt', 'rtf')

接口返回：

若 HTTP 状态码不是 200，HTTP status code 和 msg 的可能值(JSON 格式)：

status	msg	Description
404	file not exist	文件不存在
401	-	授权失败，参考 附录 MSG 定义- 401 类
302	-	文件在另外的服务器上，请处理好跳转
403	-	授权失败
5xx		服务器暂时不可用

若 HTTP 状态码为 200，消息体即文件内容。

Name	Required	Type and limit	Description
------	----------	----------------	-------------

file	Y	binary	文件内容
------	---	--------	------

❖ Client /upload_locate

上传流程:

1. 获取上传 url
2. 上传文件

接口说明: 获取文件上传地址

接口协议: **HTTP** GET

URL 结构: http://api-content.dfs.kuaipan.cn/1/fileops/upload_locate

接口参数

Name	Required	Type and limit	Description
<i>source_ip</i>	N	string	可选, 真实上传的 ip, 如果不填, 则使用调用接口的 ip。 服务器会根据 ip 来获取最快的上传地址, 请谨慎选择。

示例:

略

接口返回:

Name	Required	Type and limit	Description
url	Y	string	上传地址

状态码:

Status Code	Description
200	正常返回
500	上传服务暂时不可用

JSON 示例:

200:

```
{"url": "http://ufaceient.wps.cn"}
```

❖ Client /upload_file

接口说明：上传文件

接口协议：**HTTP** POST

URL 结构：[<url>/1/fileops/upload_file](#) (url 由 [upload_locate](#) 获得)

接口参数：

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
overwrite	Y	enum(True, False)	True/False 决定了如果路径下已存在同名文件，是否覆盖原文件。
file	Y	binary	文件数据，在 HTTP entity body 里面使用 Multipart/form-data 格式上传。此参数不用参加计算 OAuth 签名。
root	Y	string	kuaipan 或者 app_folder
path	Y	string	文件将要上传到的路径，必须包含文件名。

HTTP 示例（[蓝色加粗](#)的字段必须存在）：

POST

/cdlnode/1/fileops/upload_file?oauth_nonce=57895862&oauth_timestamp
=1328956550&oauth_consumer_key=xcKXXgo9AkXdJuBn&oauth_signatur
e_method=HMAC-SHA1&oauth_version=1.0&oauth_token=fe6265afa5c74
866a62573b34ca411fb&oauth_signature=eQ9BhjCaIpUAWDD3cbb%2Brtlf
Xt4%3D&path=%2Fttt1328956550.95sss%2Ftestw.wps&root=app_folder&
overwrite=True HTTP/1.1

Accept-Encoding: identity

Content-Length: 237

Host: ufaclient.wps.cn

Content-Type: multipart/form-data;

boundary=-----ThIs_Is_tHe_bouNdaRY_\$

Connection: close

User-Agent: klive

-----ThIs_Is_tHe_bouNdaRY_\$

Content-Disposition: form-data; **name="file"; filename="testw.wps"**

Content-Type: application/octet-stream

1328956550.991328956550.991328956550.991328956550.99

-----ThIs_Is_tHe_bouNdaRY_\$--

假如对 multipart 格式不熟悉，可以尝试用 curl 工具上传：

```
curl -F "filedata=@<本地的文件>" "<你生成的URL>"
```

接口返回：

若 HTTP 状态码为 200，消息体返回参数（当前只支持返回 JSON 化字符串）：

Name	Required	Type and limit	Description
file_id	Y	string	文件唯一标识 id
type	Y	enum(file, folder)	folder 为文件夹，file 为文件，
rev	Y	string	文件版本
size	Y	uint32	文件大小
create_time	N	string	YYYY-MM-DD hh:mm:ss
modify_time	N	string	YYYY-MM-DD hh:mm:ss
name	N	string	文件名
is_deleted	N	string/JSON bool	是否被删除的文件

HTTP status code 和 msg 的可能值：

status	msg	Description
200	ok	成功
401	-	授权失败，参考 附录 MSG 定义-401 类
403	file exist / forbidden	同名文件（夹）已存在
404	file not exist	父目录不存在
405	-	同名文件（夹）已存在或者路径不存在
413	file too large	文件过大
507	over space	用户空间已满
511		
512		

JSON 示例：

200:

```
{
    "msg": "ok",
    "is_deleted": False,
    "name": "\u6211\u7684\u5e94\u7528.exe",
    "rev": "1",
    "file_id": "433791696897",
    "create_time": "2012-01-05 14:18:15",
    "modify_time": "2012-01-05 14:18:15",
    "type": "file",
    "size": 30805
}
```

404:

```
{"msg": "file not exist"}
```

❖ Client /upload_file_by_id

接口说明：通过目录 id 和文件名来上传文件。默认把该文件覆盖。

接口协议：**HTTP** POST

URL 结构：<url>/1/fileops/upload_file_by_id (url 由 [upload_locate](#) 获得)

接口参数：

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
file	Y	binary	文件数据, 在 HTTP entity body 里面使用 Multipart/form-data 格式上传。此参数不用参加计算 OAuth 签名。
parent_id	Y	string	上传到的目录 id
name	Y	string	新上传的文件名, 不超过 255 字符, 只能是 windows 可创建文件名的字符。

其他同 [/upload_file](#)

❖ **Client /download_file**

接口说明：下载文件（支持 **HTTP Range Retrieval Requests**）

接口协议：**HTTP** GET

URL 结构：http://api-content.dfs.kuaipan.cn/1/fileops/download_file

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
root	Y	string	kuaipan 或者 app_folder
path	Y	string	相对于 root 的路径，包含文件名
rev	N	string	默认 0，当前版本。 非零可以下载其他版本的文件，具体可下载的版本号见接口 /history

请求示例（换行和缩进只是为了更好的展示）：

```
http://api-content.dfs.kuaipan.cn/1/fileops/download_file?

  oauth_version=1.0&

  oauth_consumer_key=adcc22219c3645fb9e2c5413371dcd4f&

  oauth_token=578e96fb73f9492d9f63fba434fde290&

  oauth_signature=8g5tMUhQl6oLtUbp0pO9za5x3fY%3D&

  oauth_nonce=4077316&

  oauth_timestamp=1312165717&

  root=kuaipan&

  path=myfolder%2Ftest.doc
```

接口返回：

若 HTTP 状态码不是 200，HTTP status code 和 msg 的可能值(JSON 格式)：

status	msg	Description
5xx	-	文件不存在或者服务器内部错误
500		文件不存在
404	file not exist	文件不存在

403	forbidden	文件不存在或者无权访问
401	-	授权失败，参考 附录 MSG 定义- 401 类
302	-	文件在另外的服务器上，请处理好跳转

若 HTTP 状态码为 200，消息体即文件内容。

Name	Required	Type and limit	Description
file	Y	binary	文件内容

❖ Client /download_file_by_id

接口说明：通过 id 下载文件（支持 **HTTP Range Retrieval Requests**）

接口协议：**HTTP** GET

URL 结构：http://api-content.dfs.kuaipan.cn/1/fileops/download_file_by_id

接口参数

Name	Required	Type and limit	Description
<oauth 参数>	Y	-	参考 <表 - oauth 参数>
file_id	Y	string	要下载的文件 id
rev	N	string	默认 0，当前版本。 非零可以下载其他版本的文件，具体可下载的版本号见接口 /history

附录

■ 表 - OAuth 参数

Name	Required	Type and limit	Description
oauth_consumer_key	Y	string	consumer_key
oauth_token	Y	string	access_token
oauth_signature_method	N	string	请填“HMAC-SHA1”
oauth_signature	Y	string	本次请求的签名
oauth_timestamp	Y	string	时间戳
oauth_nonce	Y	string	[0-9A-Za-z_]随机字符串，长度介于 16-32 字节。每次请求时请使用不同的 nonce。
oauth_version	N	string	请填”1.0”

■ OAuth 签名生成算法

Attention: 由于文档篇幅，作者水平有限，难免一些细节部分没有说清楚或者具有二义性，一切以 [RFC5849](#) 为准。

假设已经向快盘开放平台申请，拿到以下 consumer_key:

```
{
  'consumer_key': '79a7578ce6cf4a6fa27dbf30c6324df4',
  'consumer_secret': 'c7ed87c12e784e48983e3bcd6889dad'
}
```

并且拿到用户的授权，拿到以下 oauth_token:

```
{
  'oauth_token': 'fa361a4a1dfc4a739869020e586582f9',
  'oauth_token_secret': '0183ce137e4d4170b2ac19d3a9fda677'
}
```

假设服务器地址为 openapi.kuaipan.cn，现在需要向
http://openapi.kuaipan.cn/1/fileops/create_folder 用 GET 方法发出请求，请求参数 (parameters)
如下：

```
{
    'oauth_version': '1.0',

    'oauth_token': 'fa361a4a1dfc4a739869020e586582f9',

    'oauth_signature_method': 'HMAC-SHA1',

    'oauth_nonce': '58456623',

    'oauth_timestamp': 1328881571,

    'oauth_consumer_key': '79a7578ce6cf4a6fa27dbf30c6324df4',


    'path': '/test@kingsoft.com',

    'root': 'kuaipan'

}
```

首先计算字符基串（base string），参考算法（伪代码）可以为：

```
http_method + "&" +
    url_encode( base_uri ) + "&" +
    url_encode(
        "&".join(
            sort( [url_encode( k ) + "=" + url_encode( v ) for k, v in paramesters.items() ]
        )
    )
```

注意点：

- http_method 只能为大写，本例子中是 GET；
- base_uri 是不包括 "?" 号和其右边的 query 参数的 uri，本例中是 `http://openapi.kuaipan.cn/1/fileops/create_folder`，传输协议，主机地址必须用**小写**，请不要包含端口号，请求路径也是大小写区分；
- 参数（parameters）按照其**参数名的字典序**排序，本例子的顺序是[oauth_consumer_key, oauth_nonce, oauth_signature_method, oauth_timestamp, oauth_token, oauth_version, path, root]；
- urlencode 编码的字符需要用 utf8 转编码，16 进制部分需要**大写**，如等号“=”是%3D 而不是%3d；
- urlencode 规则：只有字母数字和[.-_~]（红色部分，不包括方括号）不用 encode，其他

字符都需要，例如 `urlencode("._~+*") = "._~%20%2B%2A"`。

这样我们根据算法生成字符基串如下（注意蓝色部分是%2540 而不是%40）：

```
GET&http%3A%2F%2Fopenapi.kuaipan.cn%2F1%2Ffileops%2Fcreate_folder
&oauth_consumer_key%3D79a7578ce6cf4a6fa27dbf30c6324df4%26oauth_
nonce%3D58456623%26oauth_signature_method%3DHMAC-SHA1%26ou
th_timestamp%3D1328881571%26oauth_token%3Dfa361a4a1dfc4a739869
020e586582f9%26oauth_version%3D1.0%26path%3D%252Ftest%2540king
soft.com%26root%3Dkuaipan
```

然后生成签名加密的密钥（记得有个&），注意假如没有 `oauth_token` 的话，蓝色部分是不用包含的：

```
c7ed87c12e784e48983e3bcd6889dad&0183ce137e4d4170b2ac19d3a9fda
677
```

假如没有 `oauth_token` 的话，密钥为 `c7ed87c12e784e48983e3bcd6889dad&`，在本例子中，有 `oauth_token`。

使用密钥通过 HMAC-SHA1 算法签名字符基串，生成签名（先生成数字签名，然后再用 base64 encode）：

```
pa7Fuh9GQnsPc+Lcn+Qu6G7LVEU=
```

最后把 `urlencode` 后的签名作为 `oauth_signature` 的值，向连接发出请求：

```
curl -k
```

```
"http://openapi.kuaipan.cn/1/fileops/create_folder?oauth_version=1.0&oauth_signature=pa7Fuh9GQnsPc%2BLcn%2BQu6G7LVEU%3D&oauth_token=fa361a4a1dfc4a739869020e586582f9&oauth_signature_method=HMAC-SHA1&oauth_nonce=58456623&oauth_timestamp=1328881571&path=%2Ftest%40kingsoft.com&oauth_consumer_key=79a7578ce6cf4a6fa27dbf30c6324df4&root=kuaipan"
```

■ HTTP 状态码

Status	Description
200	正常返回
202	接口执行错误，错误信息可以看 msg
400	参数错误
401	请求验证错误（时间戳，nonce，签名，token 等验证错误）
403	无权限操作
404	无此文件
406	同时操作太多文件
413	文件太大
500	服务器内部错误
507	空间不够
5xx	内部错误，服务暂时不可用

■ MSG 定义

202 类：接口内部逻辑错误

Message	Description
account server error	账号服务器查询失败，请过一段时间重试
cannot create app folder	不能创建用户目录，请重新获取 access token
login fail	登录失败，可能用户密码不对或者无此账号

wrong verification code	注册验证码错误或者失效
commit fail	上传失败，请过一段时间重试
bad openid	不能用此 openid 注册

400 类：请求格式不正确

Message	Description
bad parameters	参数不正确
bad request	请求 url、参数不对
no such api implemented	无此 API

401 类：授权验证出错

Message	Description
bad signature	签名不正确
request expired	请求过期（timestamp 或者 nonce 不正确）
bad consumer key	没有这个 consumer key（你确定申请了应用？）
not supported auth mode	你不能使用这种授权模式
authorization expired	用户的授权过期，请重新获取 access token
api daily limit	你今天调用接口次数太多了吧
no right to call this api	无权限调用此 api
reused nonce	nonce 被重用了
bad verifier	verifier 错误
authorization failed	授权失败（不是生产状态的应用）

403 类：

Message	Description
file exist	同名文件（夹）已经存在
forbidden	无权限访问，或者无权进行此操作

404 类：

Message	Description
file not exist	文件（夹）不存在，父目录不存在

406 类：

Message	Description
too many files	一次操作太多文件，请使用我们的客户端执行此操作

413 类：

Message	Description
file too large	文件大小超过能上传的上限

500 类：服务器内部错误

Message	Description
server error	服务器错误，应该是我们遇到 bug 了

507 类：

Message	Description
over space	用户空间已满