

云帆加速 CDN API 文档

版本历史

NO	版本号	修改说明	日期	修订人
21	V1.7.9	<ul style="list-style-type: none">加入按单个 key 删除修改自定义响应头的接口修改单次返回域名条数最大值为 10000	2016-09-01	王家豪

深圳市云帆加速科技有限公司

2016-09-12

(版权所有 , 翻版必究)

目 录

1. 概述.....	6
2. API.....	6
2.1 使用说明.....	6
2.2 token 校验.....	7
2.3 域名配置接口.....	8
2.3.1 新增域名配置接口.....	8
2.3.2 修改域名配置.....	11
2.3.3 域名列表查询.....	14
2.3.4 删除域名配置.....	17
2.3.5 查询域名配置.....	18
2.4 文件推送.....	22
2.4.1 文件注入.....	23
2.4.2 文件推送回调.....	23
2.5 刷新预热接口.....	23
2.5.1 文件预热.....	23
2.5.2 预热进度查询.....	25
2.5.3 URL 刷新接口.....	26
2.5.4 URL 刷新进度查询.....	28
2.5.5 预加载状态回调.....	30

2.5.6 目录刷新.....	30
2.6 缓存配置.....	31
2.6.1 新增缓存配置.....	31
2.6.2 修改缓存配置.....	34
2.6.3 覆盖缓存配置.....	36
2.6.4 缓存配置查询.....	39
2.6.5 删除缓存配置.....	41
2.7 自定义 IP 回源.....	43
2.7.1 自定义 IP 回源设置.....	43
2.7.2 覆盖自定义 IP 回源.....	46
2.7.3 查询自定义回源 IP.....	49
2.7.4 删除自定义回源 IP.....	51
2.8 安全防盗策略.....	52
2.8.1 黑白名单配置.....	52
2.8.3 通用防盗链配置.....	55
2.8.4 获取通用防盗链配置.....	58
2.9 HTTP 响应头.....	60
2.9.1 HTTP 响应头配置.....	60
2.9.2 修改 HTTP 响应头.....	62
2.9.3 按 key 索引修改自定义 HTTP 响应头.....	64

2.9.4 删除 HTTP 响应头.....	66
2.9.5 按 key 索引删除 HTTP 响应头.....	67
2.9.6 查看 HTTP 响应头.....	69
2.10 视频拖拽配置.....	71
2.10.1 新增视频拖拽配置.....	71
2.10.2 查询视频拖拽配置.....	73
2.10.3 修改视频拖拽.....	75
2.10.4 删除视频拖拽.....	78
2.11 数据统计查询.....	79
2.12 日志查询.....	83
2.13 SSL 证书设置.....	86
2.13.1 配置 SSL 证书.....	86
2.13.2 查询 SSL 证书配置.....	95
2.13.3 删除 SSL 证书配置.....	101
1. 附录.....	103
附录 1.通用防盗链.....	103
字段说明.....	103
算法.....	104
示例.....	104
附录 2.阿里云防盗链 A.....	105

字段说明.....	105
算法.....	105
示例.....	106
附录 3.七牛防盗链.....	107
算法.....	107
字段说明.....	107
算法.....	108
示例.....	108
附录 4.过期时间生效规则.....	109
字段说明 I.....	109
附录 5.返回错误码对照表.....	109
errno 错误码对照表.....	109

1.概述

云帆加速平台致力于将快播顶尖的播放技术分享应用到广大视频网站、视频 APP，帮助企业、开发者降低视频业务的开发门槛，加快开发进度，降低投入成本。快速帮助客户将源站内容发布到最近客户的边缘节点，解决 Internet 网络拥塞情况，提高用户访问速度。

云帆加速 CDN_API 是帮助开发者使用第三程序与云帆加速后台对接并实现相关功能，通过云帆加速 CDN_API 可以实现域名配置、文件注入、文件刷新预热、统计查询、缓存配置、视频拖拽配置、安全防盗链等功能。

2.API

2.1 使用说明

(1) 支持 GET、POST、PUT、DELETE 请求方式

(2) 返回结果都使用 UTF-8 字符进行编码

(3) 每一个资源管理请求均需在 HTTP 请求头部增加一个通过 token 接口校验获取到的 AccessToken 字段

(4) 接口请求地址：<http://api.yfcache.com>

(5) 接口结构：`/cdn/<version>/domains/<domain_id>/<module>/<field>`

2.2 token 校验

管理凭证

资源管理属于比较敏感的操作。原则上所有的资源管理操作均应在业务服务端进行。

每一个资源管理请求均需在 HTTP 请求头部增加一个 Authorization 字段，其值为符合管理凭证的字符串。管理凭证是云帆 CDN 用于验证管理请求合法性的机制。建议仅在业务服务器端使用这一类凭证，避免意外授权导致滥用。在管理现有资源时（如查看资源元信息、删除或移动资源等）通常需要带一个合法的管理凭证。不带凭证或带非法凭证的管理请求将

返回 HTTP 错误码 401，代表认证失败。管理凭证的作用与下载凭证类似，保证请求发起者拥有对目标空间的管理权限。保证服务端收到的管理请求内容未经中途篡改，具体包括代表管理动作的 URI 和该管理动作的参数信息均应未受到篡改。

算法

生成待签名的原始字符串：

(1) 抽取请求 URL 中 <path> 或 <path>?<query> 的部分与请求内容部分（即 HTTP Body），用 \n 连接起来。（如无请求内容，该部分必须为空字符串。）

注意：当 Content-Type 为 application/x-www-form-urlencoded 时，签名内容必须包括请求内容（即 HTTP Body）。

```
signingStr = "<path>?<query>\n"
```

或

```
signingStr = "<path>?<query>\n<body>"
```

注意 这里的 body 一般是 json 或者 key-value,需要转成字符串用?拼接

(2) 对 signingStr 使用 SecretKey 进行 HmacSHA1 签名

```
sha1Result = hmac_sha1(signingStr, "<SecretKey>")
```

(3) 对签名 sign 的 16 进制数转化为字符串, 然后对该字符串进行 Base64 编码

```
encodedSign=base64_encode(sign)
```

(4) 最后, 将 AccessKey 和 encodedSign 用 : 连接起来

```
Authorization = "<AccessKey>:<encodedSign>"
```

2.3 域名配置接口

2.3.1 新增域名配置接口

请求地址 : <http://api.yfcache.com/cdn/v1/domains>

请求方式 : POST

接口说明 : 为新增域名配置参数

输入参数

参数名	参数	说明	是否必填	默认值	备注
param		参数	必填		

domain	加速域名	必填		支持泛域名，格式： *.example.com
type	加速域名类型	必填		<ul style="list-style-type: none"> videocache: 视频加速 webcache: 页面加速 filecache: 文件下载加速 piccache: 图片下载加速
origin_type	源站类型	必填		<ul style="list-style-type: none"> ip: 源站类型为 IP 地址 (IP 类型支持多线路) domain: 源站类型为域名
origin_domain	源站	可选		
origin_custom_host	源站 Host	可选		回源时使用的 Host
origin_ip	源站 IP	可选		回源时使用的 IP
key_include_query	是否过滤参数	可选	off	<ul style="list-style-type: none"> on: 开启对不同参数缓存 off: 忽略参数来缓存
expire_default	默认过期时间	和 expire_force 只能二选一		只有当源站没有过期时间时生效
expire_force	强制过期时间	和 expire_default 只能二选一		强制覆盖源站过期时间
comment	域名备注	可选		域名备注信息
status	域名启用状态	可选	off	<ul style="list-style-type: none"> on: 启用 off: 禁用 全网生效时间 10 秒内

输出参数

参数名	说明	备注
-----	----	----

errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg	失败时的失败提示信息	
data	成功时返回新建配置	

示例

输入

<http://api.yfcache.com/cdn/v1/domains>

```
{
  "param": {
    "domain": "testyf.com",
    "type": "videocache",
    "origin_type": "domain",
    "origin_domain": "source.testyf.com",
    "origin_custom_host": "source.testyf.com",
    "key_include_query": "off",
    "create_at": 1453183000,
    "expire_default": 86400
  }
}
```

输出

```
{
  "data": {
```

```
"cname": "huddjtekwoy.testyf.yfcdn.net",
"create_at": "2016-09-29 11:18:39",
"domain": "testyf.com",
"expire_default": "86400",
"id": "1664",
"key_include_query": "off",
"origin_custom_host": "source.testyf.com",
"origin_domain": "source.testyf.com",
"origin_type": "domain",
"status": "on",
"type": "videocache"
},
"errno": 0
}
```

2.3.2 修改域名配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>

请求方式：PUT

接口说明：修改域名的基础配置，如：CNAME 地址、域名加速类型、域名启用与禁用、回源配置、默认过期时间、强制过期时间等

输入参数

参数名	参数	说明	是否必填	备注
domain_id		域名 id		将域名 ID 放入请求 URL 中
param		参数	必填	
	status	域名启用禁用	可选	<ul style="list-style-type: none"> on: 启用 off: 禁用
	type	加速域名类型	可选	<ul style="list-style-type: none"> videocache: 视频加速 webcache: 页面加速 filecache: 文件下载加速 piccache: 图片下载加速
	origin_type	源站类型	可选	<ul style="list-style-type: none"> ip: 源站类型为 IP 地址 (IP 类型支持多线路) domain: 源站类型为域名
	origin_domain	源站域名	可选	
	origin_ip	源站 IP	可选	
	origin_custom_host	源站 Host	可选	回源时使用的 Host
	key_include_query	是否过滤参数	可选	是否根据不同参数缓存
	expire_default	默认过期时间	和 expire_force 只能二选一	只有当源站没有过期时间时生效
	expire_force	强制过期时间	和 expire_default 只能二选一	强制覆盖源站过期时间(关于过期时间详见附录 过期时间生效规则)
	comment	域名备注信息		

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg	失败时的失败提示信息	
data	成功时返回修改结果	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100
```

```
{
  "param" : {
    "origin_custom_host" : "origin_custom_host.example.com",
    "cname" : "cname.example.com"
  }
}
```

输出

```
{
  "errno" : 0,
  "data" : {
    "origin_domain" : "source.example.com",
    "type" : "videocache",
    "expire_default" : 86400,
  }
}
```

```
{
  "origin_custom_host" : "source.example.com",
  "create_at" : 1453183000,
  "id" : 100,
  "domain" : "example.com",
  "origin_type" : "domain",
  "key_include_query" : "off",
  "cname" : "example.yfcdn.net"
},
"errmsg" : null
}
```

2.3.3 域名列表查询

请求地址：<http://api.yfcache.com/cdn/v1/domains>

请求方式：GET

接口说明：获取该用户可见的域名列表以及详细信息

输入参数

参数名	说明	是否必填	默认值	备注
domain	域名，多个域名用；隔开	可选		GET 参数
ids	多个域名 ID, 用；隔开	可选		GET 参数
offset	域名偏移量，用于域名分页	可选	0	GET 参数
max_domains	每次返回的域名最大数量为 [0-10000] 之间整型	可选	10000	GET 参数

注：PS: 当 domain 和 ids 两个参数都为空时，则返回当前用户下所有对应的域名（分页，每次最多 10000 条）

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg		失败时的失败信息	
data		成功时返回结果 数组	
	id	域名 ID	
	domain	域名	
	type	加速域名类型	<ul style="list-style-type: none"> videocache: 视频加速 webcache: 页面加速 filecache: 文件下载加速 piccache: 图片下载加速
	cname	域名 CNAME 地址	
	deploy_status	域名配置状态	<ul style="list-style-type: none"> true: 域名配置已完成 false: 域名配置进行中
	create_at	创建时间	
	status	域名状态	<ul style="list-style-type: none"> on: 启用 off: 禁用
	origin_type	回源类型	<ul style="list-style-type: none"> ip: IP 地址回源 domain: 域名回源
	origin_domain	源站域名	仅当回源类型为 domain 的时候生效
	origin_ip	源站 IP	仅当回源类型为 IP 的时候生效
	origin_custom_host	回源 Host	回源时使用的 Host 头，仅在特殊情况下出现
	key_include_query	是否过滤参数	是否根据不同参数缓存 ["on" , "off"]
	expire_default	默认过期时间	整个域名的默认过期时间 当源站返回的资源中没有过期时间时生效

expire_force	强制过期时间	整个域名的强制过期时间 会强制覆盖源站的过期时间
comment	域名备注	对域名增加备注信息

示例

输入

```
http://api.yfcache.com/cdn/v1/domains?domain=example.com
```

输出

```
{
  "data" : [
    {
      "status" : true,
      "domain" : "example.com",
      "deploy_status" : true,
      "id" : 100,
      "expire_default" : 86400,
      "origin_type" : "domain",
      "origin_custom_host" : "source.example.com",
      "expire_force" : null,
      "key_include_query" : "off",
      "origin_domain" : "source.example.com",
      "create_at" : "2016-04-09 12:00:00",
    }
  ]
}
```



```
    "cname" : "example.yfcdn.net",  
    "type" : "videocache",  
    "comment": "comment"  
  }  
],  
"errmsg" : null,  
"errno" : 0  
}
```

2.3.4 删除域名配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>

请求方式：DELETE

接口说明：删除域名配置，并停止该域名加速服务

输入参数

参数	类型	取值	说明
domain_id	int	域名 id	将域名 ID 放入请求 URL 中

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0

errmsg	失败时的失败提示信息	
data	成功时返回的结果数组	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100
```

输出

```
{
  "errno": 0,
  "errmsg": null,
  "data": [ 100 ]
}
```

2.3.5 查询域名配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>

请求方式：GET

接口说明：获取域名的详细配置信息

输入参数

参数	类型	取值	说明
----	----	----	----

domain_id	int	域名 id	将域名 ID 放入请求 URL 中
-----------	-----	-------	-------------------

输出参数

参数名	参数 ->	参数 ->->	说明	备注
errno			提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg			失败时的失败信息	
data			成功时返回结果键值对	
	id		域名 ID	
	domain		域名	
	type		加速域名类型	<ul style="list-style-type: none"> videocache: 视频加速 webcache: 页面加速 filecache: 文件下载加速 piccache: 图片下载加速
	deploy_status		域名配置状态	<ul style="list-style-type: none"> true: 域名配置已完成 false: 域名配置进行中
	status		域名状态	<ul style="list-style-type: none"> on: 启用 off: 禁用
	origin_type		回源类型	<ul style="list-style-type: none"> ip: IP 地址回源 domain: 域名回源
	origin_domain		源站域名	仅当回源类型为 domain 的时候生效
	origin_custom_host		回源 Host	回源时使用的 Host 头, 仅在特殊情况下使用
	key_include_que		是否过滤参数	回源时是否根据不同参数缓存

	ry			
	expire_default		默认过期时间	整个域名的默认过期时间 当源站返回的资源中没有过期时间时生效
	expire_force		强制过期时间	整个域名的强制过期时间 会强制覆盖源站的过期时间
	comment		域名备注	域名备注信息
	expire		特殊过期时间配置	JSON 数组
		expire_id	过期时间规则 ID	
		type	过期时间配置类型	<ul style="list-style-type: none"> suffix: 按后缀设置过期时间 urlPath: 按前缀匹配设置过期时间 regex: 按正则匹配设置过期时间
		addr	过期时间匹配的字符	
		expire_time	过期时间 (单位秒)	
	bwlist		黑白名单的配置	JSON 数组
		type	黑白名单类型	<ul style="list-style-type: none"> blacklist: IP 黑名单 whitelist: IP 白名单 blacklist_header: UA 限制黑名单 whitelist_header: UA 限制白名单 blacklist_country: 国别省份地区黑名单 whitelist_country: 国别省份地区白名单
		content	黑白名单内容	HTTP 头限制支持正则匹配 IP 限制支持 IP 地址段
	origin		特殊回源配置	JSON 数组
		host	回源 IP 地址	
		port	源站端口	
		route	回源路线名称	<ul style="list-style-type: none"> default: 默认路线

				<ul style="list-style-type: none"> • bgp: 多线机房 • ct: 电信 • cnc: 联通
		up	是否启用该 IP 回源	<ul style="list-style-type: none"> • true: 启用 • false: 禁用
		backup	是否作为备用源	<ul style="list-style-type: none"> • true: 作为备用源 • false: 不作为备用源
		weight	源站地址的权重	
	auth_config		通用防盗链配置	JSON 数组
		secrets	私有加密字符串	JSON 数组
		suffix	按后缀名匹配生效	JSON 数组
		algorithm	通用防盗链算法	<ul style="list-style-type: none"> • token.common: 防盗链方案一 (详见附录 通用防盗链) • token.ali_check_a: 阿里云防盗链 A 方案 (详见附录 阿里防盗链) • token.qiniu: 七牛防盗链方案 (详见附录 七牛防盗链)
		enable	是否启用该防盗链规则	<ul style="list-style-type: none"> • true: 启用防盗链 • false: 禁用防盗链

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/1664
```

输出

```
{
```

```
"data": {
  "bwlist": [],
  "cname": "huddjtekwoy.testyf.yfcdn.net",
  "create_at": "2016-09-29 11:18:39",
  "deploy_status": true,
  "domain": "testyf.com",
  "expire": null,
  "expire_default": "86400",
  "id": "1664",
  "key_include_query": "off",
  "origin_custom_host": "source.testyf.com",
  "origin_domain": "source.testyf.com",
  "origin_ip": null,
  "origin_type": "domain",
  "status": "on",
  "type": "videocache"
},
"errno": 0
}
```

2.4 文件推送

待更新

2.4.1 文件注入

待更新

2.4.2 文件推送回调

待更新

2.5 刷新预热接口

2.5.1 文件预热

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/preload

请求方式：POST

接口说明：将单条 URL 主动预热到边缘节点上，用户首次访问即可命中

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL 中
param		参数	必填		
	urls	需要预加载的 URL	必填		JSON 数组

输出参数

参数名	参数	说明	备注
-----	----	----	----

errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		任务提交成功时的任务 ID	
	r_id	预加载任务的 ID	用于跟踪查询使用

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/preload
```

```
{
  "param" : {
    "urls" : [
      "example.com/a/b.html",
      "http://example.com/index.html"
    ]
  }
}
```

输出

```
{
  "data" : {
    "r_id" : 1002
  },
}
```



```
"errmsg" : null,  
"errno" : 0  
}
```

2.5.2 预热进度查询

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/preload/<:r_id>

请求方式：GET

接口描述：查询单条 URL 主动预热到边缘节点的进度情况

输入参数

参数	类型	描述	取值
r_id	int	预加载任务 ID	将预加载任务 id 放入请求 URL 中
domain_id	int	域名 ID	将域名 ID 放入请求 URL 中

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		任务提交成功时返回数据	
	percent	预加载完成百分比	

	r_id	预加载任务的 ID	用于跟踪查询使用
--	------	-----------	----------

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/preload/1002
```

输出

```
{
  "errmsg" : null,
  "data" : {
    "r_id" : 1002,
    "percent" : "100%"
  },
  "errno" : 0
}
```

2.5.3 URL 刷新接口

请求地址：

```
http://api.yfcache.com/cdn/v1/domains/<:domain_id>/refresh_url
```

请求方式：POST

接口说明：提交单条 URL 全网刷新

输入参数

参数名	参数	说明	是否必填	默认值	备注
domian_id		域名 ID			将域名 ID 放入请求 URL 中
param		参数	必填		
	urls	需要刷新的 URL	必填		JSON 数组

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		任务提交成功时的任务 ID	
	r_id	刷新任务的 ID	用于跟踪查询使用

示例

输入

```
http://api.yfcache.com//cdn/v1/domains/100/refresh_url
{
  "param" : {
    "urls" : [
      "example.com/a/b.html",
      "http://example.com/index.html"
    ]
  }
}
```

```
]
}
}
```

输出

```
{
  "errno" : 0,
  "errmsg" : null,
  "data" : {
    "r_id" : 1001
  }
}
```

2.5.4 URL 刷新进度查询

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/refresh_url/<:r_id>

请求方式：GET

接口说明：提交 url 刷新后，查询刷新进度

输入参数

参数	类型	描述	说明
domain_id	int	域名 ID	将域名 ID 放入请求 URL 中

r_id	int	任务 ID	将刷新进度 id 放到请求 url 中
------	-----	-------	---------------------

输出参数

参数名		说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		任务提交成功时返回数据	
	percent	刷新完成百分比	
	r_id	刷新任务的 ID	用于跟踪查询使用

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/refresh_url/1001
```

输出

```
{
  "errno": 0,
  "errmsg": null,
  "data": { "percent": "100%", "r_id": 1001 }
}
```

2.5.5 预加载状态回调

待更新

2.5.6 目录刷新

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/refresh_dir

请求方式：PUT

接口说明：对某个目录下所有的文件执行刷新操作

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id, 将域名 ID 放入请求 URL 中			
param		参数	必填		
	dirs	多个目录, JSON 数组	必填		
domain_id		域名 ID, 将域名 ID 放入请求 URL 中			

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg	失败时的失败提示信息	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/refresh_dir

{
  "param": {
    "dirs": ["/video", "/test"]
  }
}
```

输出

```
{
  "errno": 0,
  "errmsg": null,
  "data": {
    "dirs": ["/video", "/test"]
  }
}
```

2.6 缓存配置

2.6.1 新增缓存配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/expire

请求方式：POST

接口说明：新增域名缓存过期时间设置

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL 中
param		参数	必填		
	type	过期时间配置类型	必填		<ul style="list-style-type: none"> suffix: 按后缀设置过期时间 urlPath: 按前缀匹配设置过期时间 regex: 按正则匹配设置过期时间
	addr	过期时间匹配的字符	必填		
	expire_time	过期时间(单位: 秒)	必填		
	force	覆盖源站过期时间	可选	false	<ul style="list-style-type: none"> true: 覆盖源站过期时间 false: 不覆盖源站过期时间
	sequence	顺序优先级	可选	false	<ul style="list-style-type: none"> true: 缓存规则优先级按照先后排序 false: 默认优先级

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg		失败时的失败提示信息	

data		成功时返回 JSON 字典	
	expire_id	缓存配置任务 id	
	expire_time	过期时间	

示例

<http://api.yfcache.com/cdn/v1/domains/100/expire>

```
{
  "param" : {
    "type" : "suffix",
    "expire_time" : 3,
    "addr" : ".m3u8"
  }
}
```

输出

```
{
  "data" : {
    "expire_id" : 10,
    "expire_time" : 3,
    "type" : "suffix",
    "addr" : ".m3u8"
  },
  "errmsg" : null,
  "errno" : 0
}
```

2.6.2 修改缓存配置

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/expire/<:expire_id>

请求方式：PUT

接口说明：修改域名缓存过期时间

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 ID			将域名 ID 放入请求 URL 中
expire_id		缓存任务配置 id			将任务配置 ID 放入请求 URL 中
param		参数	必填		
	type	缓存配置类型	可选		<ul style="list-style-type: none"> suffix: 按后缀设置过期时间 urlPath: 按前缀匹配设置过期时间 regex: 按正则匹配设置过期时间
	addr	过期时间匹配的字符	可选		
	expire_time	过期时间(单位: 秒)	可选		
	force	覆盖源站过期时间	可选		<ul style="list-style-type: none"> true: 覆盖源站过期时间 false: 不覆盖源站过期时间
	sequence	顺序优先级	可选		<ul style="list-style-type: none"> true: 缓存规则优先级按照先后排序

					<ul style="list-style-type: none">• false: 默认优先级
--	--	--	--	--	--

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">• 成功: errno = 0• 失败: errno > 0
errmsg		失败时的失败提示信息	
data		成功时返回 JSON 字典	
	expire_id	修改成功的过期时间配置	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/expire/12
```

```
{
  "param" : {
    "expire_time" : 100
  }
}
```

输出

```
{
  "data" : {
    "addr" : ".m3u8",
  }
}
```

```
"expire_id" : 12,  
"expire_time" : 100,  
"type" : "suffix"  
},  
"errmsg" : null,  
"errno" : 0  
}
```

2.6.3 覆盖缓存配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/expire_bat

请求方式：PUT

接口说明：覆盖域名的缓存配置

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL 中
param		参数：JSON 数组	必填		
	type	缓存配置类型	可选		<ul style="list-style-type: none">suffix: 按后缀设置过期时间urlPath: 按前缀匹配设置过期时间regex: 按正则匹配设置过期时间
	addr	过期时间匹配的字	可选		

		符			
	expire_time	过期时间(单位:秒)	可选		
	force	覆盖源站过期时间	可选	false	<ul style="list-style-type: none"> • true: 覆盖源站过期时间 • false: 不覆盖源站过期时间
	sequence	顺序优先级	可选	false	<ul style="list-style-type: none"> • true: 缓存规则优先级按照先后排序 • false: 默认优先级

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none"> • 成功: errno = 0 • 失败: errno > 0
errmsg	失败时的失败提示信息	
data	成功时返回 JSON 数组	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/expire_bat
```

```
{
  "param" : [
    {
      "type" : "suffix",
      "expire_time" : 100,
      "addr" : ".m3u8"
    }
  ]
}
```

```
  },  
  {  
    "type" : "urlPath",  
    "addr" : "/admin",  
    "expire_time" : 100  
  }  
]  
}
```

输出

```
{  
  "errno" : 0,  
  "data" : [  
    {  
      "domain_id" : 100,  
      "type" : "suffix",  
      "expire_time" : 100,  
      "expire_id" : 12,  
      "addr" : ".m3u8",  
      "force": false  
    },  
    {  
      "type" : "urlPath",
```

```
    "expire_time" : 100,  
    "expire_id" : 13,  
    "domain_id" : 100,  
    "addr" : "/admin",  
    "force": false  
  }  
],  
"errmsg" : null  
}
```

2.6.4 缓存配置查询

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/expire

请求方式：GET

接口说明：查看域名下缓存配置

输入参数

参数	类型	描述	说明
domain_id	int	域名 ID	将域名 ID 放入请求 URL 中

输出参数

参数名	说明	备注
-----	----	----

errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		返回结果 JSON 数组	
	type	过期时间配置类型	<ul style="list-style-type: none">suffix: 按后缀设置过期时间urlPath: 按前缀匹配设置过期时间regex: 按正则匹配设置过期时间
	expire_id	过期时间规则 ID	
	addr	过期时间匹配的字符	
	expire_time	过期时间时间(单位: 秒)	
	force	覆盖源站过期时间	<ul style="list-style-type: none">true: 覆盖源站过期时间false: 不覆盖源站过期时间

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/expire
```

输出

```
{
  "data" : [
    {
      "expire_time" : 3,
      "addr" : ".m3u8",
      "type" : "suffix",
```



```
    "expire_id" : 10,  
    "force": false  
  },  
  {  
    "type" : "urlPath",  
    "expire_id" : 11,  
    "addr" : "/admin",  
    "expire_time" : 3600,  
    "force": false  
  }  
],  
"errno" : 0,  
"errmsg" : null  
}
```

2.6.5 删除缓存配置

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/expire/<:expire_id>

请求方式：DELETE

接口说明：删除特殊配置的缓存规则、每条请求删除一个

输入参数

参数	类型	描述	说明
domain_id	int	域名 ID	将域名 ID 放入请求 URL 中
expire_id	int	缓存配置 ID	将缓存配置 ID 放入请求 URL 中

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		成功时返回 JSON 字典	
	expire_id	删除成功的缓存配置 ID	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/expire/12
```

输出

```
{
  "errno" : 0,
  "data" : {
    "expire_id" : 12
  },
}
```

```
"errmsg" : null  
}
```

2.7 自定义 IP 回源

2.7.1 自定义 IP 回源设置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/origin

请求方式：POST

接口说明：域名使用 IP 回源，可以添加多个 IP 地址，并制定回源策略

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL 中
param		JSON 数组参数	必填		
	host	回源 IP 地址	必填		
	port	源站端口	可选	80	
	route	路线名称	可选	default	<ul style="list-style-type: none">default: 默认路线bgp: 多线机房ct: 电信cnc: 联通cm: 移动
	up	是否启用该 IP	可选	true	<ul style="list-style-type: none">true: 启用false: 禁用
	backup	是否作为备用源	可选	false	<ul style="list-style-type: none">true: 备用

					<ul style="list-style-type: none"> false: 不作为备用
	weight	源站地址的权重	可选		
	max_fails	最大失败次数	可选		
	method	回源策略算法选择	可选		<ul style="list-style-type: none"> round robin : 随机回源 hash : 一致性哈希回源

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg	失败时的失败提示信息	
data	成功时返回结果	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/origin
```

```
{
  "param" : [
    {
      "host" : "127.0.0.1",
      "backup" : false,
      "weight" : 10,
      "max_fails" : 3,
      "up" : true,
      "port" : 80,
      "route" : "default"
    }
  ]
}
```

输出

```
{
  "errmsg" : null,
  "data" : [
    {
      "up" : true,
      "host" : "127.0.0.1",
      "weight" : 10,

```

```
    "route" : "default",  
  
    "port" : 80,  
  
    "backup" : true  
  
  }  
],  
  
"errno" : 0  
  
}
```

2.7.2 覆盖自定义 IP 回源

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/origin

请求方式：PUT

接口说明：覆盖当前自定义 IP 回源策略

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL 中
param		JSON 数组参数	必填		
	host	回源 IP 地址	必填		
	port	源站端口	可选	80	
	route	路线名称	可选	default	<ul style="list-style-type: none">default: 默认路线bgp: 多线机房ct: 电信cnc: 联通

46

	up	是否启用该 IP	可选	true	<ul style="list-style-type: none"> true: 启用 false: 禁用
	backup	是否作为备用源	可选	false	<ul style="list-style-type: none"> true: 备用 false: 不作为备用
	weight	源站地址的权重	可选		
	max_fails	最大失败次数	可选		
	method	回源策略算法选择	可选		<ul style="list-style-type: none"> round robin hash

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg	失败时的失败提示信息	
data	成功时返回结果	

示例

输入

```
PUT /cdn/v1/domains/100/origin
```

```
{
  "param" : [
    {
      "port" : 80,
      "backup" : false,
      "up" : true,

```

```
    "weight" : 10,  
    "route" : "default",  
    "max_fails" : 3,  
    "host" : "127.0.0.1"  
  }  
]  
}
```

输出

```
{  
  "errno" : 0,  
  "data" : [  
    {  
      "route" : "default",  
      "host" : "127.0.0.1",  
      "backup" : true,  
      "port" : 80,  
      "weight" : 10,  
      "up" : true  
    }  
  ],  
  "errmsg" : null  
}
```


2.7.3 查询自定义回源 IP

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/origin

请求方式：GET

接口说明：查看自定义 IP 回源策略

输入参数

参数	类型	说明	取值
domain_id	int	域名 id	将域名 ID 放入请求 URL

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		成功时返回的 JSON 数组	
	host	源站 IP	
	weight	源站 权重	
	port	端口	
	backup	是否做备用回源	
	up	是否启用	
	route	回源自定义线路	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/origin
```

输出

```
{
  "data" : [
    {
      "backup" : true,
      "up" : true,
      "weight" : 10,
      "host" : "127.0.0.1",
      "route" : "default",
      "port" : 80
    }
  ],
  "errno" : 0
}
```

2.7.4 删除自定义回源 IP

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/origin

请求方式：DELETE

接口说明：删除自定义 IP 回源策略

输入参数

参数	类型	说明	取值
domain_id	int	域名 id	将域名 ID 放入请求 URL

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg	失败时的失败提示信息	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/origin
```

输出

```
{  
  "errmsg" : null,  
}
```

```
"errno" : 0
}
```

2.8 安全防盗策略

2.8.1 黑白名单配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/bwlist

请求方式：PUT

接口说明：客户根据自身的需求设置 IP 黑白名单，User-Agent 黑白名单

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL
param		参数, JSON 数组	必填		
	type	黑白名单类型	必填		<ul style="list-style-type: none"> blacklist: IP 黑名单 whitelist: IP 白名单 blacklist_header: 根据 HTTP 头限制黑名单 whitelist_header: 根据 HTTP 头限制白名单 blacklist_country: 针对国别、省份、地区黑名单 whitelist_country: 针对国别、省份、地区白名单 blacklist_regex: 正则匹配 uri 黑名单

	content	限制匹配内容， JSON 数组 HTTP 头匹配时， 数据结构为字典 数据来源 ipip.net	可选		<ul style="list-style-type: none"> IP 地址支持填 IP 段 & CIDR 支持 HTTP 头匹配支持填正则匹配 国别匹配时填国家地区的英文，用; 隔开 当 content 内容为空数组或为空时，则清空所选类型的规则
--	---------	--	----	--	--

不同类型数据说明

类型	数据结构	举例	备注
blacklist whitelist	[Str, Str, Str, ...]	["1.2.3.4", "4.3.2.1/32"]	数组中多个 IP
blacklist_header whitelist_header	[[{Str: Str}, Hash, Hash, ...]	["Referer": ".*\example\\.com"], {}]	多对 Hash Header_key: Header_val
blacklist_country whitelist_country	[Str, Str, Str, ...]	["China,Guangzhou"]	数组中多个地区 地区写法详见 http://ipip.net 数据库
blacklist_regex	[Str, Str, Str]	["^/test/.*"]	数组中多个正则表达式

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg		失败时的失败提示信息	

data		成功时返回修改后的数据: JSON 数组	
	type	黑白名单类型	
	content	限制匹配内容	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/bwlist
```

```
{
  "param" : [
    {
      "content" : [
        {
          "Referer" : "http://xxx.com/"
        }
      ],
      "type" : "blacklist_header"
    }
  ]
}
```

输出

```
{
  "errmsg" : null,
```

```
"errno" : 0,
"data" : [
  {
    "content" : [
      {
        "Referer" : "http://xxx.com/"
      }
    ],
    "type" : "blacklist_header"
  }
]
```

2.8.3 通用防盗链配置

请求地址：

http://api.yfcache.com/cdn/v1/domains/<domain_id>/auth_config

请求方式：GET

接口说明：为域名配置通用防盗链

输入参数

参数名	参数 ->	说明	是否必填	默认值	备注
domian_id		域名 id			将域名 ID 放入请求 URL

param		参数	必填		JSON 数组
	secrets	私有加密字符串	必填		JSON 数组 其中有且最多有两个用于加密的字符串
	suffix	按后缀名匹配生效	可选	所有后缀	JSON 数组
	algorithm	算法	可选	common	<ul style="list-style-type: none"> token.common: 防盗链方案一 (详见附录 通用防盗链) token.ali_check_a: 阿里云防盗链 A 方案 (详见附录 阿里防盗链) token.qiniu: 七牛防盗链方案 (详见附录 七牛防盗链)
	enable	是否启用	可选	false	<ul style="list-style-type: none"> true: 启用防盗链 false: 禁用防盗链

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg	失败时的失败提示信息	
data	成功时返回修改后数据	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/auth_config
```

```
{
  "param" : [
    {
```



```
"suffix" : [  
  ".m3u8"  
],  
"algorithm" : "common",  
"secrets" : [  
  "123456",  
  "123"  
],  
"enable" : false  
}  
]  
}
```

输出

```
{  
  "data" : [  
    {  
      "secrets" : [  
        "123456",  
        "123"  
      ],  
      "suffix" : [  
        ".m3u8"  
      ]  
    }  
  ]  
}
```

```
    ],  
    "enable" : false,  
    "algorithm" : "common"  
  }  
],  
"errmsg" : null,  
"errno" : 0  
}
```

2.8.4 获取通用防盗链配置

请求地址：

http://api.yfcache.com/cdn/v1/domains/<domain_id>/auth_config

请求方式：GET

接口说明：获取通用防盗链配置

输入参数

参数	类型	说明	取值
domain_id	int	域名 id	将域名 ID 放入请求 URL

输出参数

参数名	参数	说明	备注
-----	----	----	----

errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data	成功时返回的数据		
	secrets	私有加密字符串	
	suffix	按后缀名匹配生效	
	algorithm	防盗链算法名称	
	enable	是否启用防盗链	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/auth_config
```

输出

```
{
  "data" : [
    {
      "algorithm" : "common",
      "secrets" : [
        "123456",
        "123"
      ],
      "suffix" : [
```

```
        ".m3u8"  
    ],  
    "enable" : false  
  }  
],  
"errno" : 0  
}
```

2.9 HTTP 响应头

2.9.1 HTTP 响应头配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/header

请求方式：POST

接口说明：自定义对 http 响应头内容进行配置

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL
param		参数	必填		
	key	HTTP header 的 key	必填		
	val	HTTP header 的内容	必填		

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		JSON 字典	
	id	该条规则的 id	
	key	res_header	
	val	http 头自定义规则	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/header
```

```
{
  "param" : {
    "val" : "video/mp4",
    "key" : "content_type"
  }
}
```

输出

```
{
```

```
"data" : {
  "val" : {
    "content_type" : "video/mp4"
  },
  "id" : 99,
  "key" : "res_header"
},
"errno" : 0
}
```

2.9.2 修改 HTTP 响应头

请求地址：PUT

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/header/<:rule_id>

请求方式：PUT

接口说明：修改自定义的 HTTP 响应头

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id ,			将域名 ID 放入请求 URL
rule_id		修改 http 响应头任务 id			将 http 响应头任务 ID 放入请求 URL
param		参数	必填		

	key	HTTP header 的 key	必填		
	val	HTTP header 的内容	必填		

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg		失败时的失败提示信息	
data		JSON 字典	
	id	http 响应头规则的 id	
	key	res_header	
	val	http 头自定义规则	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/header/99
```

```
{
  "param" : {
    "id" : 99,
    "key" : "content_type",
    "val" : "text/plain"
  }
}
```

输出

```
{
  "data" : {
    "id" : 99,
    "key" : "res_header",
    "val" : {
      "content_type" : "text/plain"
    }
  },
  "errno" : 0
}
```

2.9.3 按 key 索引修改自定义 HTTP 响应头

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/header_by_key

请求方式：PUT

接口说明：按照 key 对 http 响应头进行设置

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL

param		参数	必填		
	key	HTTP header 的 key	必填		要修改的 key 值, 必须已存在
	val	HTTP header 的 内容	必填		

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg		失败时的失败提示信息	
data		JSON 字典	
	key	res_header	
	val	http 头自定义规则	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/header_by_key
```

```
{
  "param" : {
    "key" : "content_type",
    "val" : "application/json"
  }
}
```

输出

```
{
  "data" : {
    "key" : "res_header",
    "val" : {
      "content_type" : "application/json"
    }
  },
  "errno" : 0
}
```

2.9.4 删除 HTTP 响应头

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/header/<:rule_id>

请求方式：DELETE

接口说明：删除自定义设置的 HTTP 响应头

输入参数

参数	类型	说明	取值
domain_id	int	域名 id	将域名 ID 放入请求 URL
rule_id	int	http 头规则任务 id	将 http 响应头任务 ID 放入请求 URL

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/header/99
```

输出

```
{  
  "data" : [],  
  "errno" : 0  
}
```

2.9.5 按 key 索引删除 HTTP 响应头

请求地址：

```
http://api.yfcache.com/cdn/v1/domains/<:domain_id>/header_by_key
```

请求方式：DELETE

接口说明：按 key 索引删除自定义响应头

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL
param		参数	必填		
	key	HTTP header 的 key	必填		要修改的 key 值，必须已存在

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg	失败时的失败提示信息	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/header_by_key

{
  "param": {
    "key": "content_type"
  }
}
```

输出

```
{
```

```
"data" : [],  
"errno" : 0  
}
```

2.9.6 查看 HTTP 响应头

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/header

请求方式：GET

接口说明：查看自定义设置的 HTTP 响应头基本信息

输入参数

参数	类型	说明	取值
domain_id	int	域名 id	将域名 ID 放入请求 URL

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		JSON 数组	
	id	该条规则的 id	
	key	res_header	
	val	http 头自定义规则	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/header
```

输出

```
{
  "errno" : 0,
  "data" : [
    {
      "key" : "res_header",
      "val" : {
        "content_type" : "text/plain"
      },
      "id" : 99
    }
  ]
}
```

2.10 视频拖拽配置

2.10.1 新增视频拖拽配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/streaming

请求方式：POST

接口说明：配置域名下视频拖动的类型、参数等

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL
param		参数	必填		
	type	需要拖动视频的类型	必填		<ul style="list-style-type: none"> mp4: 对 mp4 封装拖动的支持 flv: 对 flv 封装拖动的支持
	sarg	视频拖动的开始参数 (秒)	必填		
	earg	视频拖动的结束参数 (秒)	必填		
	suff	需要支持的视频文件后缀名	必填		
	mode	支持的模式			<ul style="list-style-type: none"> second: 按秒拖动 byteAV: 按字节拖动 byte: 按字节拖动

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		JSON 字典	
	id	视频拖动配置任务 id	
	type	需要拖动视频的类型	
	sarg	视频拖动的开始参数(秒)	
	earg	视频拖动的结束参数(秒)	
	suff	需要支持的视频文件后缀名	
	mode	拖拽的模式	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/streaming
```

```
{
  "param" : {
    "suff" : "mp4",
    "sarg" : "start",
    "type" : "mp4",
    "mode" : "second",
    "earg" : "end"
  }
}
```


输出参数

```
{
  "data" : {
    "val" : {
      "suff" : "mp4",
      "mode" : "second",
      "earg" : "end",
      "type" : "mp4",
      "sarg" : "start"
    },
    "id" : 99,
    "key" : "streaming"
  },
  "errno" : 0
}
```

2.10.2 查询视频拖拽配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/streaming

请求方式：GET

接口说明：查看视频拖拽的配置

输入参数

参数	类型	说明	取值
domain_id	int	域名 id	将域名 ID 放入请求 URL

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg		失败时的失败提示信息	
data		JSON 数组	
	id	视频拖拽配置任务 id	
	type	需要拖动视频的类型	
	sarg	视频拖动的开始参数(秒)	
	earg	视频拖动的结束参数(秒)	
	suff	需要支持的视频文件后缀名	
	mode	拖拽的模式	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/streaming
```

输出

```
{
```

```
"data" : [  
  {  
    "id" : 99,  
    "val" : {  
      "sarg" : "start",  
      "type" : "mp4",  
      "suff" : "mp4",  
      "earg" : "end",  
      "mode" : "second"  
    },  
    "key" : "streaming"  
  }  
],  
"errno" : 0  
}
```

2.10.3 修改视频拖拽

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/streaming/<:rule_id>

请求方式：PUT

接口说明：修改视频拖拽配置参数

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL
rule_id		视频拖拽任务 id			将视频拖拽配置任务 ID 放入请求 URL
param		参数	必填		
	type	需要拖动视频的类型	可选		<ul style="list-style-type: none"> mp4: 对 mp4 封装拖动的支持 flv: 对 flv 封装拖动的支持
	sarg	视频拖动的开始参数 (秒)	可选		
	earg	视频拖动的结束参数 (秒)	可选		
	suff	需要支持的视频文件后缀名	可选		
	mode	支持的模式			<ul style="list-style-type: none"> second: 按秒拖动 byteAV: 按字节拖动 byte: 按字节拖动

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg		失败时的失败提示信息	
data		JSON 字典	
	id	视频拖拽配置任务 id	

	type	需要拖动视频的类型	
	sarg	视频拖动的开始参数(秒)	
	earg	视频拖动的结束参数(秒)	
	suff	需要支持的视频文件后缀名	
	mode	拖拽的模式	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/streaming/99
```

```
{
  "param" : {
    "suff" : "flv"
  }
}
```

输出

```
{
  "data" : {
    "val" : {
      "sarg" : "start",
      "earg" : "end",
      "suff" : "flv",
    }
  }
}
```

```
    "type" : "mp4",
    "mode" : "second"
  },
  "id" : 99,
  "key" : "streaming"
},
"errno" : 0
}
```

2.10.4 删除视频拖拽

请求地址：

http://api.yfcache.com/cdn/v1/domains/<:domain_id>/streaming/<:rule_id>

请求方式：DELETE

接口说明：删除自定义设置的视频拖拽配置

输入参数

参数	类型	说明	取值
domain_id	int	域名 id	将域名 ID 放入请求 URL
rule_id	int	视频拖拽配置任务 id	将视频拖拽配置任务 ID 放入请求 URL

输出参数

参数名	说明	备注
-----	----	----

errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg	失败时的失败提示信息	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/streaming/99
```

输出参数

```
{
  "data" : [],
  "errno" : 0
}
```

2.11 数据统计查询

请求地址 : <http://api.yfcloud.com/web/api/my/flow>

请求方式 : GET/POST

接口说明 : 查询带宽、流量基本数据统计

输入参数

参数	是否必须	类型	说明
domain	是	string	需要获取数据的频道,可有多个频道,多个频道值请用英

			文分号 “;” 分隔开,频道内域名与业务号以 “:” 分割开
starttime	是	string	开始时间 格式:年月日-时分,如: 20160901-0000
endtime	否	string	结束时间 格式:年月日-时分,如: 20160901-0800
isp	否	string	运营商 CHN(电信),CNC(联通),CMN(移动),GER(教育网),GWB(长宽),WAS(华数),CTT(铁通),例如输入 CHN,会返回电信有关数据
province	否	string	省份 输入省份首字母大写,例如广东,输入 GD

输出参数

参数	类型	取值	说明
errorCode	string		错误码
errorID	int		错误 id
errorMsg	string		错误信息描述
backsource	string	单位:字节	回源流量
flow	string	单位:字节	流量
domain	string		域名

示例

输入

```
http://api.yfccloud.com/web/api/my/flow?domain=dl.cdn.xxx.com&starttime=20160901-0000
```

输出


```
{  
  
  "provider": "YunFanCDN", //CDN 厂商  
  
  "curtime": "2016-09-01 11:11:11", //服务器响应时间点  
  
  "ok  
": true, //状态结果,true:正常 false:异常 异常具体可参考错误信息  
  
  "errorCode": 0,  
  "errorID": "",  
  "errorMsg": "",  
  "data": [  
    {  
      "data": {  
        "2016-08-12 15:00": {  
          "backsource": 0, //回源流量(字节)  
  
          "flow": 0 //流量(字节)  
        },  
        "2016-08-12 15:05": {  
          "backsource": 956246678, //回源  
  
          "flow": 90896159390 //流量  
        }  
      }  
    }  
  ]  
}
```

```
    },  
    "domain": "dl.cdn.xxx.com" //域名  
  },  
  {  
    "data": {  
      "2016-08-12 15:00": {  
        "backsource": 0, //回源流量(字节)  
        "flow": 0 //流量(字节)  
      },  
      "2016-08-12 15:05": {  
        "backsource": 956246678, //回源  
        "flow": 90896159390 //流量  
      }  
    },  
    "domain": "yf.v.xxx.com" //域名  
  }  
]  
}
```

2.12 日志查询

请求地址：http://api.yfccloud.com/web/api/my/log

请求方式：GET/POST

接口说明：查询域名下日志情况

输入参数

参数	是否必须	类型	说明
domain	是	string	需要获取数据的频道,可有多个频道,多个频道值请用英文分号 ";" 分隔开,频道内域名与业务号以 ":" 分割开
starttime	是	string	开始时间 格式:年月日-时分,如:20160901-0000
endtime	是	string	结束时间 格式:年月日-时分,如:20160901-0800
type	否	string	日志类型, day(天) min(5分钟) hour(小时,默认)
mtype		string	edge(边缘) backsource(回源)

输出参数

参数	类型	取值	说明
provider	string		CDN 厂商
curtime	string		服务器响应时间点
ok	string		状态结果,true:正常 false:异常 异常具体可参考错误信息
errorCode	string		错误代码

errorID	int		错误 ID
errorMsg	string		错误信息
data			数据
uri	string		http://log.xxx.net/user_log/custom/20160818/1600/yf.v.xxx.com_2016_08_18_16.gz
size	string		文件大小
domain	string		yf.v.xxx.com

示例

输入

```
http://api.yfcloud.com/web/api/my/log?domain=yf.v.xxx.com&starttime=20160818-1500&endtime=20160918-1600
```

输出

```
{  
  "provider": "YunFanCDN", //CDN 厂商  
  "curtime": "2016-09-01 11:11:11", //服务器响应时间点  
  "ok": true, //状态结果,true:正常 false:异常 异常具体可参考错误信息  
  "errorCode": 0,  
  "errorID": "",  
  "errorMsg": "",
```

```
"data": [  
  {  
    "data": {  
      "2016-08-18 15:00": {  
        "uri":  
"http://log.xxx.net/user_log/custom/20160818/1500/yf.v.xxx.com_2016_08_  
18_15.gz",  
        "size": 0  
      },  
      "2016-08-18 16:00": {  
        "uri":  
"http://log.xxx.net/user_log/custom/20160818/1600/yf.v.xxx.com_2016_08_  
18_16.gz",  
        "size": 0  
      }  
    },  
    "domain": "yf.v.xxx.com"  
  }  
]  
}
```

2.13 SSL 证书设置

2.13.1 配置 SSL 证书

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/ssl

请求方式：PUT

接口说明：给制定域名配置 SSL 证书

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL
param		参数	必填		
	cert_chain	证书	必填		
	pkey	私钥	必填		
	cert_typ	证书类型	必填		pem der
	pkey_typ	私钥类型	必填		pem der
	passphrase	私钥密码	选填		
	enable	是否启用 https	选填	true	<ul style="list-style-type: none"> true: 启用 https false: 禁用 https
	force_ssl	强制使用 https	选填	false	<ul style="list-style-type: none"> true: 强制 https false: 不强制 https

输出参数

参数名	说明	备注
errno	提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg	失败时的失败提示信息	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/ssl
```

```
{
  "param": {
    "cert_chain": "
    -----BEGIN CERTIFICATE-----
    MIIF8jCCBNqgAwIBAgIQDmTF+8I2reFLFyrrQceMsDANBgkqhkiG9w0BAQsFADBw
    MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
    d3cuZGlnaWNlcnQuY29tMS8wLQYDVQQDEyZEaWdpQ2VydCBTSEEyIEhpZ2ggQXNz
    dXJhbmNlIFNlcnZlcjBDQTAEFw0xNTEwMDAwMDBaFw0xODEwMjg0MjAwMDBa
    MIGlMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTEUMBIGA1UEBxML
    TG9zIEFuZ2VsZXN0PDA6BgNVBAoTM01udGVybmV0IENvcnBvcmlzIGZvciBB
    c3NpZ25lZCB0eW11cyBhbmQgTnVtYmVycyETMBEGA1UECxMKVG9jaG5vbG9neTEY
    MBYGA1UEAxMPd3d3LmV4YW1wbGUub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
    MIIBCgKCAQEAs0CWL2FjPiXB1611RfvvE0KzLJmG9LWAC3bcBjgsH6NiVVo2dt6u
    Xfzi5bTm7F3K7srFUBYkL078mraM9qizrHoIeyofrV/n+pZZJauQsPjCPxMEJnRo
    D8Z4KpWKX0LyDu1SputoI4n1Q/htEhtiQnuoBfNZxF7WxcxGwEsZuS1KcXIkH15V
    RJ0reKFHTaXcB1qcZ/QRaBIv0yhxvK1yBTwddT4cli6GfHcCe3xGmaSL328Fgs3
```

```
jYrvG29PueB6VJi/tbbPu6qTfwp/H1brq djh29U52Bhb0fJkM9DwXCP/Cattcc7a
z8EXnCO+LK8vkhw/kAiJWPKx4RBvgy73nwIDAQABo4ICUDCCAkwwHwYDVR0jBBgw
FoAUUWj/kK8CB3U8zN1LZGKiErhZcjswhQYDVR00BBYEFKZPYB4fLdHn8S0gKpUW
50ia6m5IMIGBBgNVHREeEjB4gg93d3cuZXhxbXsZS5vcmeCC2V4YW1wbGUuY29t
ggtleGFtcGx1LmVkdYILZXhxbXsZS5uZXSCC2V4YW1wbGUub3Jngg93d3cuZXhh
bXsZS5jb22CD3d3dy5leGFtcGx1LmVkdYIPd3d3LmV4YW1wbGUubmV0MA4GA1Ud
DwEB/wQEAWIFoDAdBgNVHSUEFjAUBgggrBgEFBQCcDAQYIKwYBBQUHAWIwdQYDVR0f
BG4wbDA0oDKgMIYuaHR0cDovL2NybdMuZGlnaWN1cnQuY29tL3NoYTIItaGEtc2Vy
dmVyLWc0LmNybdA0oDKgMIYuaHR0cDovL2NybdQuZGlnaWN1cnQuY29tL3NoYTIIt
aGEtc2VydMvyLWc0LmNybdBMBgNVHSAERTBDMDcGCWCGSAGG/wWBATAqMCgGCCsG
AQUFBwIBFhxodHRwczovL3d3dy5kaWdpY2VydC5jb20vQ1BTMAgGBmeBDAECAjCB
gWYIKwYBBQUHAQEEdzB1MCQGCCsGAQUFBzABhhodHRwOi8vb2NzcC5kaWdpY2Vyd
dC5jb20wTQYIKwYBBQUHMAKGQWh0dHA6Ly9jYWN1cnRzLmRpb21jZXJ0LmNvbS9E
aWdpQ2VydFNIQTJJaWdoQXNzdXJhbmN1U2VydMvyQ0EuY3J0MAwGA1UdEwEB/wQC
MAAwDQYJKoZIhvcNAQELBQADggEBAISomhGn2L0LJn5SJHuyVZ3qMI1RCIdvqe0Q
61s+C8ctRwR03UU3x8q8OH+2ahx1QmpzdC5a14XQzJLiLjiJ2Q1p+hub8MFimMVP
PZjb2tZm2ipWuMRM+zgpRVM6nVJ9F3vFfUSH0b4/JsEIUvPY+d8/Krc+kPQwLvy
ieqRbcuFjmqfyPmUv1U9QoI4TQikpw7TZU0zYZANP4C/gj4Ry48/znmUaRvy2kvI
17gRQ21qJTK5suoiYoYNo3J9T+pXPGU7Lydz/HwW+w0DpArtAauKI8aNX4ohFUKS
wDSiIIWIWjiJGbEeIO0TIFwEVWTObn1/faPXpk5IRXicapqiII=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEsTCCA5mgAwIBAgIQBOHnpNxc8vNtwCtCuF0VnzANBgkqhkiG9w0BAQsFADB5
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUN1cnQgSW5jMRkwFwYDVQQLEwB3
```


d3cuZG1naWN1cnQuY29tMSswKQYDVQDEyJEaWdpQ2VydCBlaWdoIEFzc3VyYW5j
ZSBFViBSb290IENBMB4XDTEzMTAyMjEyMDAwMFOXDTI4MTAyMjEyMDAwMFowcDEL
MAkGA1UEBhMCVVMxFTATBgNVBAoTDERpZ21DZXJ0IEluYzEZMBCGA1UECXMqd3d3
LmRpZ21jZXJ0LmNvbTEvMC0GA1UEAxMmRGlnaUN1cnQgU0hBMiBIaWdoIEFzc3Vy
YW5jZSBTZXJ2ZXIgc0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC2
4C/CJAbIbQRf1+8KZAayfSImZRauQkCbztyfn3YHPsMwVYcZuU+UD1qUH1VwtMIC
Kq/Qm04LQNF0DtyyBSe75CxEamu0si4QzrZCwvV1ZX1QK/IHe1NnF9Xt4ZQaJn1
itrSxwUfqJfJ3KSxgoQtqxq2lnMcZgqaFD15EWCo3j/018QsIJzJa9buLnqS9UdAn
4t07Qj0jBSjEuyjMmqwrIw14xnmXnG3Sj4I+4G3FhahnSMSTeXXkgisdaScus0X
sh5ENWV/UyU50RwKmmMbGZJ0aAo3wsJSSMs5WqK24V3B3aAguCGikyZvFEohQcft
bZvySC/za/WiaJJTL17jAgMBAAGjggFJMIIBRTASBgNVHRMBAf8ECDAGAQH/AgEA
MA4GA1UdDwEB/wQEAWIBhjAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAWIw
NAYIKwYBBQUHAQEEDAmMCQGCCsGAQUFBzABhhodHRwOi8vb2NzcC5kaWdpY2Vy
dC5jb20wSwYDVR0fBEQwQjBAoD6gPIY6aHR0cDovL2NybdQuZGlnaWN1cnQuY29t
L0RpZ21DZXJ0SGlnaEFzc3VyYW5jZUVWUm9vdENBLmNybdQuZGlnaWN1cnQuY29t
BFUdIAAwKjAoBggrBgEFBQcCARYcaHR0cHM6Ly93d3cuZGlnaWN1cnQuY29tL0NQ
UzAdBgNVHQ4EFgQUUWj/kK8CB3U8zN11ZGKiErhZcjswhwYDVR0jBBgwFoAUsT7D
aQP4v0cB1JgmGggC72NkK8MwDQYJKoZIhvcNAQELBQADggEBABiK1YkD5m3FXPwD
aOpKj4PWUS+Na0QWnqxj9dJubISZi6qBcYRb7TR0sLd5kinMLYBq8I4g4Xmk/gNH
E+r1hspZcX30BJZr011YPf7TMSVcGDiEo+afgv2Mw5gxTs14nhr9hctJqvIni5ly
/D6q1UEL2tU2ob8cbkdJf17ZSHwD2f2LSaCYJkJA69aSEaRkClDUXPUD1gJea6zu
xICaEnL6VpPX/78whQYwvwt/Tv9XBZ0k7YXDK/umdaisLRbvFXknsuvCnQsH6qqF
0wGjIChBWUMo0hjqvbszt3tkBigAVBRQHvFwY+3sAzM2fTYS5yh+Rp/BIAV0Ae
cPUeybQ=

```
-----END CERTIFICATE-----  
  
",  
"pkey": "  
-----BEGIN RSA PRIVATE KEY-----  
  
MIIEogIBAAKCAQEAKjeRsnYWguVVGnCCcx9111h8h6Qm+5JEvew2mubiqaguvIAF  
QYaqHKS+fSARj6vgeitNtv9611BNX5XvTmuEJzN1DnccyPEsq+NixTQ5tuIx2hy  
1QGJmXXC9Lk/wwcZ4NHBXzS0kIi76cPAVzzm8PPpjrKYior/mdpmwGHmH2D+Uuur  
Gd0mhJdfyr5CjRVF5ywDhKQBLaUh7UveVh/+PLLIhSWJ2GcSFCPPLOY+eOwIkCmr  
eSypo9SiFKBZTH0t/gCHU6p7wZn3pTep+DeNjcApwuNLU9YBYexY16UaZfXGcF+W  
wySI/42nQExdPt6BCizCwpgxoZ6m0sg3XnKcFQIDAQABAOIBAAdA0G5s4zL9E1qW  
xUyZUQwf+ZV09CYynGZbeF34xBesGJHr09eDSZ8AIR7t31EkMo985GndzIcjFqN+  
OY/HT5fu0b4c9IWXSSMBtZ2YuKA02ko6kDh2yg+dwt0G733KNWNspE4HvE1LLx8q  
gbkSIVakktpsS5rbyFAmgGBcuuClint5jRgV8Ggs1WRS2c7rmHwAoCbMeTxza/W0  
Wi+a7vL5AjhrVswfwM56Fhkoi+BF+UwE+DvWibjFzUIoN4sJUQMAbOmHVxS1eiE2  
8/xvS98AD+UUI1PUuOvZUJMOawLKxHvAMjJYeWq7fT1SCHItIEJKyxX330PBjhi4  
4RYfUQECgYEAzrzP3f109U0Imz5X/p7v5sYs+Zzb//VUxe4ng518323fEhzMg7vI  
IuNxLM8FLrxQLjUyKLTN3c1I1kVImu7t6LXz2MrsKtDRmExpawbTkaUgQPZL/LzE  
Q0or5E278apTCCGBzaTmS10VRT15sa2wjnbICDqXheZVh/QZjPE+3j0CgYEAtQ7z  
M+r4F/hnqmmTN+Z51UctzKyyCDv2jSn28FsQD01z4zGnmCW2P5Id04VH1bI57Znt  
26sISkcnwuiPRK8xMdiDHvxS0uScr8US0SRQ+33aCAzEFLcdWMtcQhzZe8rZ1jwA  
rH21Pj+S7BD1115D4hXKhCMiYodbjvsc4qKuUECgYBvAedDs3h4MtLqQQGa0NDU  
Q5kmtt1SiPLqDMTe3u20MxTSbvBJ0+zU/vdoLgkEXU1YymwyU0MvU+kj9zPTLXRF  
v12fD0be3p2Mf4sJutkLnXTCLP4ESX+/s3256xt92g0YcFU27NJTqTmgoi/T/gWN  
ma4y7PMrjNBkXefSTD+LqQKBgFe9ge/j6ww6rYnw1Y4eW2Ywk0Q/YBsLa+UwTEGX
```

```
74Vq5EDyHCRoChRua0ofwVvyvY26xx8P1/ne29n2J6kIOQ/Nu2T1o6e0h5XKK987
+R6Wk2UZQqBDSK5yBihnsKED4WKn0zBbzsv2p01PFuMy1TTnmHdGS3Bx5dW5LMV1
GtrBAoGAR66Ke6w1vfJ/J8gk+sa3PQh1C/nZiXDBAYJeFJ2voIF4Hia8rCgE3kCd
tDyhhiKL/pxaaWJB4Q74chtKMXMXzqBkaaF8RNtKRFHuHwveouc7gCLuREWap4ab
LvagXef80NLLUd9phN9m+XQQMNKZ3fKt/LBxn6P2/mk/BirWmvQ=
-----END RSA PRIVATE KEY-----
",
  "cert_typ": "pem",
  "pkey_typ": "pem"
}
}
```

输出

```
{
  "errno": 0,
  "errmsg": null,
  "data": [{
    "cert_chain": "-----BEGIN CERTIFICATE-----
MIIF8jCCBNqgAwIBAgIQDmTF+8I2reFLFyrrQceMsDANBgkqhkiG9w0BAQsFADBw
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRG1naUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMS8wLQYDVQQDEyZEaWdpQ2VydCBTSEEyIEhpZ2ggQXNz
dXJhbmNlIFNlcnZlcjBDQTAEfw0xNTEyMDAwMDBaFw0xODEyMjE1MjE1MjE1MjE1
MIGlMQswCQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcml5pYTEUMBIGA1UEBxML
```

TG9zIEFuZ2VsZXMxPDA6BgNVBAoTM01udGVybmV0IENvcnBvcnF0aW9uIGZvcibB
c3NpZ25lZCBOYW1lcYBhbmQgTnVtYmVyczETMBEGA1UECxMKVGVjaG5vbG9neTEY
MBYGA1UEAxMPd3d3LmV4YW1wbGUub3JnMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAs0CWL2FjPiXB1611RfvvE0KzLJmG9LWAC3bcBjgsH6NiVVo2dt6u
Xfzi5bTm7F3K7srFUBYkL078mraM9qizrHoIeyofrV/n+pZZJauQsPjCPxMEJnRo
D8Z4KpWKX0LyDu1SputoI4n1Q/htEhtiQnuoBFNZxF7WxcxGwEsZuS1KcXIKH15V
RJ0reKFHTaXcB1qcZ/QRaBIv0yhxvK1yBTwddT4cli6GfHcCe3xGmaSL328Fgs3
jYrvG29PueB6VJi/tbbPu6qTfwp/H1brqjdjh29U52Bhb0fJkM9DwXCP/Cattcc7a
z8EXnCO+LK8vkhw/kAiJWPKx4RBvgy73nwIDAQABo4ICUDCCAkwwHwYDVR0jBBgw
FoAUUWj/kK8CB3U8zN1lZGKiErhZcjswhQYDVR00BBYEFKZPYB4fLdHn8S0gKpUW
50ia6m5IMIGBBgNVHREeEjB4gg93d3cuZXhbbXBsZS5vcmeCC2V4YW1wbGUuY29t
ggtleGFtcGx1LmVkdYILZXhbbXBsZS5uZXSCC2V4YW1wbGUub3Jngg93d3cuZXhh
bXBsZS5jb22CD3d3dy5leGFtcGx1LmVkdYIPd3d3LmV4YW1wbGUubmV0MA4GA1Ud
DwEB/wQEAwIFoDAdBgNVHSUEFjAUBgggrBgEFBQcDAQYIKwYBBQUHAWIwdQYDVR0f
BG4wbDA0oDKgMIYuaHR0cDovL2NybdMuZGlnaWN1cnQuY29tL3NoYTIItaGEtc2Vy
dmVyLWc0LmNybdA0oDKgMIYuaHR0cDovL2NybdQuZGlnaWN1cnQuY29tL3NoYTIIt
aGEtc2VydmVyLWc0LmNybdBMBgNVHSAERTBDMDcGCWCGSAGG/wwBATAqMCgGCCsG
AQUFBwIBFhxodHRwczovL3d3dy5kaWdpY2VydC5jb20vQ1BTMAgGBmeBDAECAjCB
gwYIKwYBBQUHAQEEdzB1MCQGCCsGAQUFBzABhhhodHRwOi8vb2Nzc5kaWdpY2Vyd
dC5jb20wTQYIKwYBBQUHMAKGQWh0dHA6Ly9jYWN1cnRzLmRpZ21jZXJ0LmNvbS9E
aWdpQ2VydFNIQTJJIaWdoQXNzdXJhbmNlU2VydmVyQ0EuY3J0MAwGA1UdEwEB/wQC
MAAwDQYJKoZIhvcNAQELBQADggEBAISomhGn2L0LJn5SJHuyVZ3qMI1RCIdvqe0Q
61s+C8ctRwR03UU3x8q8OH+2ahx1QmpzdC5a14XQzJLiLjiJ2Q1p+hub8MFimMVP
PZjb2tZm2ipWuMRM+zgpRVM6nVJ9F3vFfUSH0b4/JsEIUvPY+d8/Krc+kPQwLvy

```
ieqRbcuFjmqfyPmUv1U9QoI4TQikpw7TZU0zYZANP4C/gj4Ry48/znmUaRvy2kvI
17gRQ21qJTK5suoYoYNo3J9T+pXPGU7Lydz/HwW+w0DpArtAauKI8aNX4ohFUKS
wDSiIIWIWJiJGbEeIO0TIFwEVWTONbNl/faPXpk5IRXicapqiII=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEsTCCA5mgAwIBAgIQBOHnpNxc8vNtwCtCuF0VnzANBgkqhkiG9w0BAQsFADB3
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWNlcnQuY29tMSswKQYDVQQDEyJEaWdpQ2VydCBlaWdoIEFzc3VyYW5j
ZSBFViBsb290IENBMB4XDTEzMTAyMjEyMDAwMFoXDTEzMTAyMjEyMDAwMFowDEL
MAkGA1UEBhMCVVMxFTATBgNVBAoTDERpZ21DZXJ0IEluYzEZMBcGA1UECmQd3d3
LmRpZ21jZXJ0LmNvbTEvMC0GA1UEAxMmRGlnaUNlcnQgU0hBMiBiIaWdoIEFzc3Vy
YW5jZSBTZjZ2ZXIgc0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC2
4C/CJAbIbQRf1+8KZAayfSImZRauQkCbztyfn3YHPsMwVYcZuU+UDlqUH1VwtMIC
Kq/Qm04LQNfE0DtyyBSe75CxEamu0si4QzrZCwvV1ZX1QK/IHe1NnF9Xt4ZQaJn1
itrSxwUfqJfJ3KSxgoQtqxq2lnMcZgqaFD15EWCo3j/018QsIJzJa9buLnqS9UdAn
4t07Qj0jBSjEuyjMmqwrIw14xnmXnG3Sj4I+4G3FhahnSMSTeXXkgisdaScus0X
sh5ENWV/UyU50RwKmmMbGZJ0aAo3wsJSSMs5WqK24V3B3aAguCGikyZvFEohQcft
bZvySC/zA/WiaJJTL17jAgMBAAGjggFJMIIBRTASBgNVHRMBAf8ECDAGAQH/AgEA
MA4GA1UdDwEB/wQEAwIBhjAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIw
NAYIKwYBBQUHAQEEDAmMCQGCCsGAQUFBzABhhodHRwOi8vb2NzcC5kaWdpY2V5
dC5jb20wSwYDVVR0fBEQwQjBAoD6gPIY6aHR0cDovL2NybdQuZGlnaWNlcnQuY29t
L0RpZ21DZXJ0SGlnaEFzc3VyYW5jZUVWUm9vdENBLmNybdQuZGlnaWNlcnQuY29t
L0N0
BFUdIAAwKjAoBggrBgEFBQcCARYcaHR0cHM6Ly93d3cuZGlnaWNlcnQuY29tL0N0
UzAdBgNVHQ4EFgQUUWj/kK8CB3U8zN1lZGKiErhZcjswhwYDVVR0jBBgwFoAUsT7D
```

```
aQP4v0cB1JgmGggC72NkK8MwDQYJKoZIhvcNAQELBQADggEBABiK1YkD5m3fXPwd
aOpKj4PWUS+Na0QWnqxj9dJubISZi6qBcYRb7TR0sLd5kinMLYBq8I4g4Xmk/gNH
E+r1hspZcX30BJZr011YPf7TMSVcGDiEo+afgv2MW5gxTs14nhr9hctJqvIni5ly
/D6q1UeL2tU2ob8cbkdJf17ZSHwD2f2LSaCYJkJA69aSEaRkClDUxPUd1gJea6zu
xICaEnL6VpPX/78whQYwvwt/Tv9XBZ0k7YXDK/umdaisLRbvfXknsuvCnQsH6qqF
0wGjIChBWUMo0oHjqvbszt3tkBigAVBRQHvFwY+3sAz2fTYS5yh+Rp/BIAV0Ae
cPUeybQ=
-----END CERTIFICATE-----",
    "pkey": "-----BEGIN RSA PRIVATE KEY-----
MIIIEogIBAAKCAQEAKjeRsnYWguVVGnCCX9111h8h6Qm+5JEvew2mubiqaguvIAF
QYaqHKS+fSARj6vgeitNTv9611BNX5XvTmuEJzN1DnccyPEsq+NixTQ5tuIx2hyy
1QGJmXXC9Lk/wwcZ4NHBXzS0kIi76cPAVzzm8PPpjrKYior/mdpmwGHmH2D+Uuur
Gd0mhJdfyr5CjRVF5ywdHKQBLaUh7UveVh/+PLLIhSWJ2GcSFCPPLOy+e0wIkCmr
eSypo9SiFKBZTH0t/gCHU6p7wZn3pTep+DeNjcApwuNLU9BYexY16UaZfXGcF+W
wySI/42nQExdPt6BCizCwpgxoZ6m0sg3XnKCfQIDAQABAoIBAAdA0G5s4zL9E1qW
xUyZUQwf+ZV09CYynGZbeF34xBesGJHr09eDSZ8AIR7t31EkMo985GndzIcJfQn+
OY/HT5fu0b4c9IWXSSMBtZ2YuKA02ko6kDh2yg+dwt0G733KNWNspE4HvE1LLx8q
gbkSIVaktpsS5rbyFAmgGBcuuClInt5jRgV8Ggs1WRS2c7rmHwAoCbMeTxza/W0
Wi+a7vL5AjhrVswfwM56Fhkoi+BF+UwE+DvWibjFzUIoN4sJUQMAbOmHVxS1eiE2
8/xvS98AD+UUI1PUuOvZUJMOawLKxHvAMjJYewq7fT1SCHItIEJKyxX330PBjhi4
4RYfUQECgYEAzrzP3f109U0Imz5X/p7v5sYs+Zzb//VUxe4ng518323fEhzMg7vI
IuNxLM8FLrxQLjUyKLTN3c1I1kVImu7t6LXz2MrsKtDRmExpawbTkaUgQPZL/LzE
Q0or5E278apTCCGBzaTmS10VRT15sa2wjnbICDqXheZVh/QZjPE+3j0CgYEAtQ7z
M+r4F/hnqmmTN+Z51UctzKyyCDv2jSn28FsQD01z4zGnmCW2P5Id04VH1bI57ZNt
```

```
26sISkcnwuiPRK8xMdiDHvxS0uScr8US0SRQ+33aCAzEFLcdWMtcQhzZe8rZ1jwA
rH21Pj+S7BD1115D4hXKhCMiYodbjvscc4qKuUECgYBvAedDs3h4MtLq0QGa0NDU
Q5kmtt1SiPLqDMTe3u20MxTSbvBJ0+zU/vdoLgkEXU1YmwyU0MvU+kj9zPTLXRF
v12fD0be3p2Mf4sjuTkLnXTCLP4ESX+/s3256xt92g0YcfU27NJTqTmgoi/T/gWN
ma4y7PMrjNBkXefSTD+LqQKBgFe9ge/j6ww6rYnw1Y4eW2Ywk0Q/YBsLa+UwTEGX
74Vq5EDyHCRoChRua0ofwVvyvY26xx8P1/ne29n2J6kIOQ/Nu2T1o6e0h5XKK987
+R6Wk2UZQqBDSK5yBihnsKED4wKnOzBbzsv2p01PFuMy1TTnmHdGS3Bx5dw5LMV1
GtrBAoGAR66Ke6w1vfJ/J8gk+sa3PQh1C/nZiXDBAYJeFJ2voIF4Hia8rCgE3kCd
tDyhhiKL/pxaaWJB4Q74chtKMXMXzqBkaaF8RNtKRFHuHWveouc7gCLuREWap4ab
LvagXef80NLLUd9phN9m+XQQMNKZ3fKt/LBxn6P2/mk/BirWmvQ=
-----END RSA PRIVATE KEY-----",
    "cert_typ": "pem",
    "pkey_typ": "pem",
    "enable": true,
    "force_ssl": false
  }
}
```

2.13.2 查询 SSL 证书配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/ssl

请求方式：GET

接口说明：获取指定域名 SSL 证书配置

输入参数

参数名	参数	说明	是否必填	默认值	备注
domain_id		域名 id			将域名 ID 放入请求 URL
param		参数	必填		

输出参数

参数名	参数	说明	备注
errno		提交任务是否成功	<ul style="list-style-type: none"> 成功: errno = 0 失败: errno > 0
errmsg		失败时的失败提示信息	
data		返回值	
	cert_chain	证书	
	pkey	私钥	
	cert_typ	证书类型	pem der
	pkey_typ	私钥类型	pem der
	passphrase	私钥密码	
	enable	是否启用	
	force_ssl	是否强制使用 https	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/ssl
```



```
50ia6m5IMIGBBgNVHREeEjB4gg93d3cuZXhbbXBsZS5vcmeCC2V4YW1wbGUuY29t
ggtleGFtcGx1LmVkdYILZXhbbXBsZS5uZXSCC2V4YW1wbGUub3Jngg93d3cuZXhh
bXBsZS5jb22CD3d3dy5leGFtcGx1LmVkdYIPd3d3LmV4YW1wbGUubmV0MA4GA1Ud
DwEB/wQEAwIFoDAdBgNVHSUEFjAUBgggrBgEFBQcDAQYIKwYBBQUHAWIwdQYDVR0f
BG4wbDA0oDKgMIYuaHR0cDovL2NybdMuZGlnaWN1cnQuY29tL3NoYTItaGEtc2Vy
dmVyLWc0LmNybdA0oDKgMIYuaHR0cDovL2NybdQuZGlnaWN1cnQuY29tL3NoYTIta
aGEtc2VydMvYlWc0LmNybdBMBgNVHSAERTBDMDcGCWCGSAGG/WwBATAqMCgGCCsG
AQUFBwIBFhxodHRwczovL3d3dy5kaWdpY2Vydc5jb20vQ1BTMAgGBmeBDAECAjCB
gWYIKwYBBQUHAQEEdzB1MCQGCCsGAQUFBzABhhodHRwOi8vb2NzcC5kaWdpY2Vy
dC5jb20wTQYIKwYBBQUHMAKGQWh0dHA6Ly9jYWN1cnRzLmRpZ21jZXJ0LmNvbS9E
aWdpQ2VydFNIQTJJIaWdoQXNzdXJhbmN1U2VydmVyQ0EuY3J0MAwGA1UdEwEB/wQC
MAAwDQYJKoZIhvcNAQELBQADggEBAISomhGn2L0LJn5SJHuyVZ3qMI1RCIdvqe0Q
61s+C8ctRwR03UU3x8q80H+2ahx1QmpzdC5a14XQzJLiLjiJ2Q1p+hub8MFimMVP
PZjb2tZm2ipWuMRM+zgpRVM6nVJ9F3vFfUSH0b4/JsEIUvPY+d8/Krc+kPQwLvy
ieqRbcuFjmqfyPmUv1U9QoI4TQikpw7TZU0zYZANP4C/gj4Ry48/znmUaRvy2kvI
17gRQ21qJTK5suoiYoYNo3J9T+pXPGU7Lydz/Hww+w0DpArtAaukI8aNX4ohFUKS
wDSiIIWIWjiJGbEeI00TIFwEVWTONbNl/faPXpk5IRXicapqiII=
```

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

```
MIIEsTCCA5mgAwIBAgIQBOHnpNxc8vNtwCtCuF0VnzANBgkqhkiG9w0BAQsFADBs
MQswCQYDVQQGEwJVUzEVMBMGA1UEChMMRGlnaUN1cnQgSW5jMRkwFwYDVQQLExB3
d3cuZGlnaWN1cnQuY29tMSswKQYDVQQDEyJEaWdpQ2VydCB1aWdoIEFzc3VyYW5j
ZSBFViBsb290IENBMB4XDTEzMTAyMjEyMDAwMFoXDTEzMTAyMjEyMDAwMFowDEL
MAkGA1UEBhMCVVMxFTATBgNVBAoTDERpZ21DZXJ0IEl1eZEMBcGA1UECxMQd3d3
```

```
LmRpZ21jZXJ0LmNvbTEvMC0GA1UEAxMmRGlnaUNlcnQgU0hBMiBIaWdoIEFzc3Vy
YW5jZSBTZXJ2ZXIgc0EwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC2
4C/CJAbIbQRf1+8KZAayfSImZRauQkCbztyfn3YHPsMwVYcZuU+UDlqUH1VwtMIC
Kq/Qm04LQNfE0DtyyBSe75CxEamu0si4QzrZCwvV1ZX1QK/IHe1NnF9Xt4ZQaJn1
itrSxwUfqJfJ3KSxgoQtqxq2lnMcZgqaFD15EWCo3j/018QsIJzJa9buLnqS9UdAn
4t07Qj0jBSjEuyjMmqwrIw14xnmXnG3Sj4I+4G3FhahnSMSTeXXkgisdaScus0X
sh5ENWV/UyU50RwKmmMbGZJ0aAo3wsJSSMs5WqK24V3B3aAguCGikyZvFEohQcft
bZvySC/zA/WiaJJTL17jAgMBAAGjggFJMIIBRTASBgNVHRMBAf8ECDAGAQH/AgEA
MA4GA1UdDwEB/wQEAWIBhjAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAWIw
NAYIKwYBBQUHAQEEDAmMCQGCCsGAQUFBzABhhodHRwOi8vb2NzcC5kaWdpY2Vy
dC5jb20wSwYDVR0fBEQwQjBAoD6gPIY6aHR0cDovL2NybdQuZGlnaWNlcnQuY29t
L0RpZ21DZXJ0SGlnaEFzc3VyYW5jZUVWUm9vdENBLmNybDA9BgNVHSAENjA0MDIG
BFUdIAAwKjAoBggrBgEFBQcCARYcaHR0cHM6Ly93d3cuZGlnaWNlcnQuY29tL0NQ
UzAdBgNVHQ4EFgQUUWj/kK8CB3U8zN1lZGKiErhZcjswHwYDVR0jBBgwFoAUsT7D
aQP4v0cB1JgmGggC72NkK8MwDQYJKoZIhvcNAQELBQADggEBABiK1YkD5m3fXPwd
aOpKj4PWUS+Na0QWnqxj9dJubISZi6qBcYRb7TROsLd5kinMLYBq8I4g4Xmk/gNH
E+r1hspZcX30BJZr011YPf7TMSVcGDiEo+afgv2MW5gxTs14nhr9hctJqvIni5ly
/D6q1UEL2tU2ob8cbkdJf17ZSHwD2f2LSaCYJkJA69aSEaRkCldUxPud1gJea6zu
xICaEnL6VpPX/78whQYvwwt/Tv9XBZ0k7YXDK/umdaisLRbvFXknsuvCnQsH6qqF
0wGjIChBWUMo0oHjqvbszt3tkBigAVBRQHvFwY+3sAzm2fTYS5yh+Rp/BIAV0Ae
cPUeybQ=
```

-----END CERTIFICATE-----

”，

“pkey”： ”

```
-----BEGIN RSA PRIVATE KEY-----  
MIIIEogIBAAKCAQEAKjeRsnYWguVVGNCCCX9111h8h6Qm+5JEvew2mubiqaguvIAF  
QYaqHKS+fSARj6vgeitNTv9611BNX5XvTmuEJzN1DnccyPEsq+NixTQ5tuIx2hyy  
1QGJmXXC9Lk/wwcZ4NHBXzS0kIi76cPAVzzm8PPpjrKYior/mdpmwGHmH2D+Uuur  
Gd0mhJdfyr5CjRVF5ywdHKQBLaUh7UveVh/+PLLIhSWJ2GcSFCPPLOY+eOwIkcmr  
eSypo9SiFKBZTH0t/gCHU6p7wZn3pTep+DeNjcApwuNLU9YBYexY16UaZfXGcF+W  
wySI/42nQExdPt6BCizCwpgxoZ6m0sg3XnKCfQIDAQABAoIBAAdA0G5s4zL9E1qW  
xUyZUQwf+ZV09CYYnGZbeF34xBesGJHr09eDSZ8AIR7t31EkMo985GndzIcjFqN+  
OY/HT5fu0b4c9IWXSSMBtZ2YuKA02ko6kDh2yg+dwt0G733KNWNspE4HvE1LLx8q  
gbkSIVaktpsS5rbyFAmgGBcuuClint5jRgV8Ggs1WRS2c7rmHwAoCbMeTxza/W0  
Wi+a7vL5AjhrVswfwM56Fhkoi+BF+UwE+DvWibjFzUIoN4sJUQMAbOmHVxS1eiE2  
8/xvS98AD+UUI1PUuOvZUJMOawLKxHvAMjJYewq7ft1SCHItIEJKyxX330PBjhi4  
4RYfUQECgYEAzrzP3f109U0Imz5X/p7v5sYs+Zzb//VUxe4ng518323fEhzMg7vI  
IuNxLM8FLrxQLjUykLTN3c1I1kVImu7t6LXz2MrsKtDRmExpawbTkaUgQPZL/LzE  
Q0or5E278apTCCGBzaTmS10VRT15sa2wjnbICDqXheZVh/QZjPE+3j0CgYEAtQ7z  
M+r4F/hnqmmTN+Z51UctzKyyCDv2jSn28FsQD01z4zGnmCW2P5Id04VH1bI57ZNt  
26sISkcwnuipRK8xMdiDHvxS0uScr8US0SRQ+33aCAzEFLcdWMtcQhzZe8rZ1jwA  
rH21Pj+S7BD1115D4hXKhCMiYodbjvsc4qKuUECgYBvAedDs3h4MtLq0QGa0NDU  
Q5kmtt1SiPLqDMTe3u20MxTSbvBJ0+zU/vdoLgkEXU1YymwyU0MvU+kj9zPTLXRF  
v12fD0be3p2Mf4sjUtKLnXTCLP4ESX+/s3256xt92g0YcfU27NJTqTmgoi/T/gWN  
ma4y7PMrjNBkXefSTD+LqQKBgFe9ge/j6ww6rYnw1Y4eW2Ywk0Q/YBsLa+UwTEGX  
74Vq5EDyHCRoChRua0ofwVvyvY26xx8P1/ne29n2J6kIOQ/Nu2T1o6e0h5XKK987  
+R6Wk2UZQqBDSK5yBihnsKED4WKnOzBbzsv2p01PFuMy1TTnmHdGS3Bx5dw5LMV1  
GtrBAoGAR66Ke6w1vfJ/J8gk+sa3PQh1C/nZiXDBAYJeFJ2voIF4Hia8rCgE3kCd
```

```
tDyhhiKL/pxaaWJB4Q74chtKMXMXzqBkaaF8RNtKRFHuHwveouc7gCLuREWap4ab
LvagXef80NLLUd9phN9m+XQQMNKZ3fKt/LBxn6P2/mk/BirWmvQ=
-----END RSA PRIVATE KEY-----
",
  "cert_typ": "pem",
  "pkey_typ": "pem",
  "enable": true,
  "force_ssl": false
}]
}
```

2.13.3 删除 SSL 证书配置

请求地址：http://api.yfcache.com/cdn/v1/domains/<:domain_id>/ssl

请求方式：DELETE

接口说明：删除自定义设置的 SSL 证书配置

输入参数

参数	类型	说明	备注
domain_id	int	域名 id	将域名 ID 放入请求 URL

输出参数

参数名	说明	备注
-----	----	----

errno	提交任务是否成功	<ul style="list-style-type: none">成功: errno = 0失败: errno > 0
errmsg	失败时的失败提示信息	

示例

输入

```
http://api.yfcache.com/cdn/v1/domains/100/ssl
```

输出

```
{  
  "errmsg" : null,  
  "errno" : 0,  
  "data" : []  
}
```

1. 附录

对文档中的内容进行补充，目前包括：

- 防盗链算法
- 错误码说明
- 过期时间优先级说明

附录 1.通用防盗链

字段说明

`http://<domain>/<uri>?upt=<arg_upt>`

字段	说明	备注
domain	访问域名	
uri	访问 uri	

arg_upt	防盗链校验参数	
secret_key	双方协商的私钥	
expire	过期时间戳 (秒)	

算法

其中 `<arg_upt>` 算法为:

$MD5(\text{'<secret_key>' + '\&' + \text{'<expire>' + '\&' + \text{'<uri>'}}) \text{(截断 13 到 20 位)} + \text{'<expire>'}$

示例

1. 访问 url 为 `http://www.example.com/index.html`, 协商 `<secret_key>` 为 `12345678`, 过期时间戳为 `1464710400` (实际为 2016/06/01 00:00:00)
2. 按照如上算法进行计算 (js):

```
var uri      = "/index.html";  
  
var secret_key = "12345678"  
  
var expire    = "1464710400";  
  
var str      = secret_key + '&' + expire + '&' + uri
```

此时拼接的字符串为 `12345678&1464710400&/index.html`


```
var hash_str = md5(str)
```

```
var sstring = substring(hash_str, 12, 20)
```

```
var arg_upt = sstring + expire
```

结果为 76ab9c161464710400

1. 实际访问时使用

```
http://www.example.com/index.html?upt=76ab9c161464710400
```

附录 2. 阿里云防盗链 A

字段说明

```
http://domain/uri?auth_key=<arg_auth_key>
```

字段	说明	备注
domain	访问域名	
uri	访问 uri	含 /
arg_auth_key	防盗链校验参数	
secret_key	双方协商的私钥	
etime	防盗链过期时间戳 (秒)	
rand	随机数	一般设置成 0
uid	保留字段	设置为 0

算法

其中 <arg_upt> 算法为:

```
`<etime>`-`<rand>`-`<uid>`-MD5(`<uri>`-`<etime>`-`<rand>`-`<uid>`-`<secret_key>`)
```

示例

1. 访问 url 为 `http://www.example.com/index.html`, 协商 `<secret_key>` 为 `12345678`, 过期时间戳为 `1464710400` (实际为 2016/06/01 00:00:00), `<rand>` 设置为 0, `<uid>` 设置为 0
2. 按照如上算法进行计算 (js):

```
var uri      = "/index.html";  
var secret_key = "12345678"  
var etime     = "1464710400";  
var rand      = "0"  
var uid       = "0"  
var sstring   = uri + "-" + etime + "-" + rand + "-" + uid + "-" + secret_key  
ey
```

此时 `sstring` 为 `/index.html-1464710400-0-0-12345678`

```
var hashvalue = md5(sstring)  
var arg_auth_key = etime + "-" + rand + "-" + uid + "-" + hashvalue
```

结果为 1464710400-0-0-f5bf4b6290c23963a4f219400126d542

1. 实际访问时使用

```
http://www.example.com/index.html?auth_key=1464710400-0-0-f5bf4b6290  
c23963a4f219400126d542
```

附录 3. 七牛防盗链

算法

基于时间戳的防盗链是通过对时间有关的字符串进行签名，将时间、签名通过一定的方式传递给 CDN 服务器作为判定依据，CDN 边缘节点根据约定的算法判断来访 URL 是否有访问权限。

通过，执行下一步；不通过，响应 HTTP status code 403。

字段说明

```
http://domain/uri?sign=<arg_sign>&t=<arg_t>
```

字段	说明	备注
domain	访问域名	
uri	访问 uri	含 /
secret_key	双方协商的私钥	数组，支持多个 key，任意 key 通过则视作防盗链通过
arg_t	防盗链过期时间戳（秒）	十六进制
arg_sign	防盗链验证串	

算法

其中 `<arg_sign>` 算法为:

```
MD5( ('<secret_key>' '<uri>' '<arg_t>').to_lower() )
```

示例

1. 访问 url 为 `http://www.example.com/index.html`, 协商 `<secret_key>` 为 `12345678`, 过期时间戳为 `1464710400` (实际为 2016/06/01 00:00:00)
2. 按照如上算法进行计算 (js):

```
var uri      = "/index.html";  
  
var secret_key = "12345678"  
  
var etime    = "1464710400";  
  
var arg_t    = etime.toString(16);  
  
var sstring  = ( secret_key + uri + arg_t ).toLowerCase()
```

此时 `sstring` 为 `12345678/index.html574db500`

```
var arg_sign = md5(sstring)
```

结果为 `c107f93d0f01c527e9b9a5e6438260a3`

1. 实际访问时使用

```
http://www.example.com/index.html?t=574db500&sign=c107f93d0f01c527e  
9b9a5e6438260a3
```

附录 4.过期时间生效规则

字段说明 I

数字越大，则优先级越高

字段	说明	优先级
expire_default	默认过期时间	1
用户源站过期时间设定	符合 HTTP 标准的缓存设置， 如：Expires,Cache-Control,Last-Modified	2
expire_force	强制过期时间	3

附录 5.返回错误码对照表

errno 错误码对照表

错误码	类型	备注
1	参数缺失	GET / POST 参数缺失
2	JSON 中缺少必要参数	
3	JSON 中枚举字段出现错误	如：传参的值不是枚举字段允许的值，则报错
4	操作的对象不存在	如：操作的域名不存在，操作的某条规则不存在
5	数据库操作出错	如：数据库 <i>insert</i> 操作失败
6	JSON 中有数据结构不符合要求	如：要求是数组类型，得到的却是字典类型
7	获取 设备列表 错误	
8	新建域名时，域名已存在	
99	接口内部错误	具体原因见 <i>errmsg</i>