

七牛上传服务指南

梁涛、茅力夫
@七牛云存储

2013-09-10

Contents

| | |
|------------------------------|-----------|
| 1 上传业务模型 | 2 |
| 1.1 本地直传模型 | 2 |
| 1.2 客户端通知上传模型 | 4 |
| 1.2.1 HTML表单上传模型 | 5 |
| 1.3 七牛云通知上传模型 | 6 |
| 2 上传方式 | 7 |
| 2.1 普通上传 | 7 |
| 2.2 断点续上传 (Block) | 7 |
| 2.2.1 串行传输 | 7 |
| 2.2.2 并发传输 | 8 |
| 2.2.3 细粒度传输 (Chunk) | 8 |
| 2.3 追加上传 (Append, 研发中) | 9 |
| 2.4 特性一览表 | 10 |
| 3 上传授权凭证 | 11 |
| 3.1 策略参数 | 11 |
| 3.2 签名算法 | 14 |
| 3.3 魔法变量 (Magic Variables) | 15 |
| 3.4 自定义变量 (X-Variables) | 17 |
| 4 API | 18 |
| 4.1 直传 | 18 |
| 4.1.1 直传单一资源 (文件) | 18 |
| 4.1.2 直传单一资源 (图片+魔法变量+自定义变量) | 20 |
| 4.2 断点续上传 | 21 |
| 4.2.1 微型资源上传 | 21 |
| 4.2.2 大型资源上传 (6MB) | 25 |
| 4.3 追加上传 | 28 |
| 5 FAQ | 29 |
| 6 术语 | 30 |

Chapter 1

上传业务模型

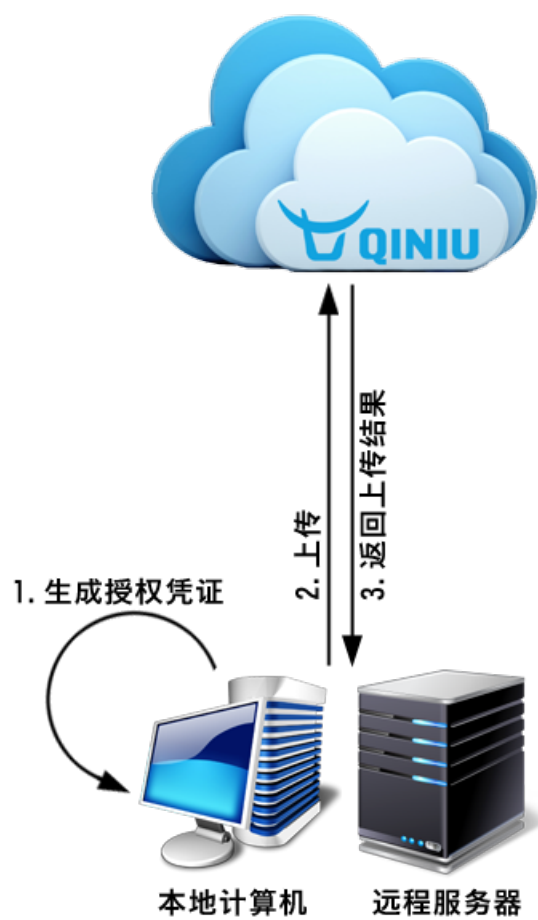
七牛云存储提供多种上传业务模型，以满足不同业务场景的需求。用户可根据实际情况择优选用，使用[官方SDK](#)/第三方SDK开发上传模块或工具。还可以直接使用[官方工具](#)便捷快速地上传资源，无需额外编程。

1.1 本地直传模型

本模型适用于将存储在本地计算机或服务器的资源直接上传至七牛云，整个过程不涉及其它业务服务器。非常合适应用在大批量数据迁移、定期同步备份等场景下。

■ 参与实体：七牛云、本地资源存储机。

■ 官方实用工具：[qrsync](#)。



1.2 客户端通知上传模型

本模型适用于业务客户端将用户产生的内容（UGC）上传至七牛云后，将上传结果通知给业务服务器。合适附件存储、网盘、摄影摄像应用等场景。

■ 参与实体：七牛云、业务客户端、业务服务器。

■ 潜在问题：1) 上传过程与通知过程是分离的，任一过程失败都将导致业务状态不一致；2) 上传过程与通知过程都可能走低速线路，带来不必要的延迟。



1.2.1 HTML表单上传模型

本模型是客户端通知上传模型的一种特例，适用于业务客户端通过网页表单形式上传资源的场景（网站）。

■ 参与实体：七牛云、业务客户端（浏览器）、业务服务器。



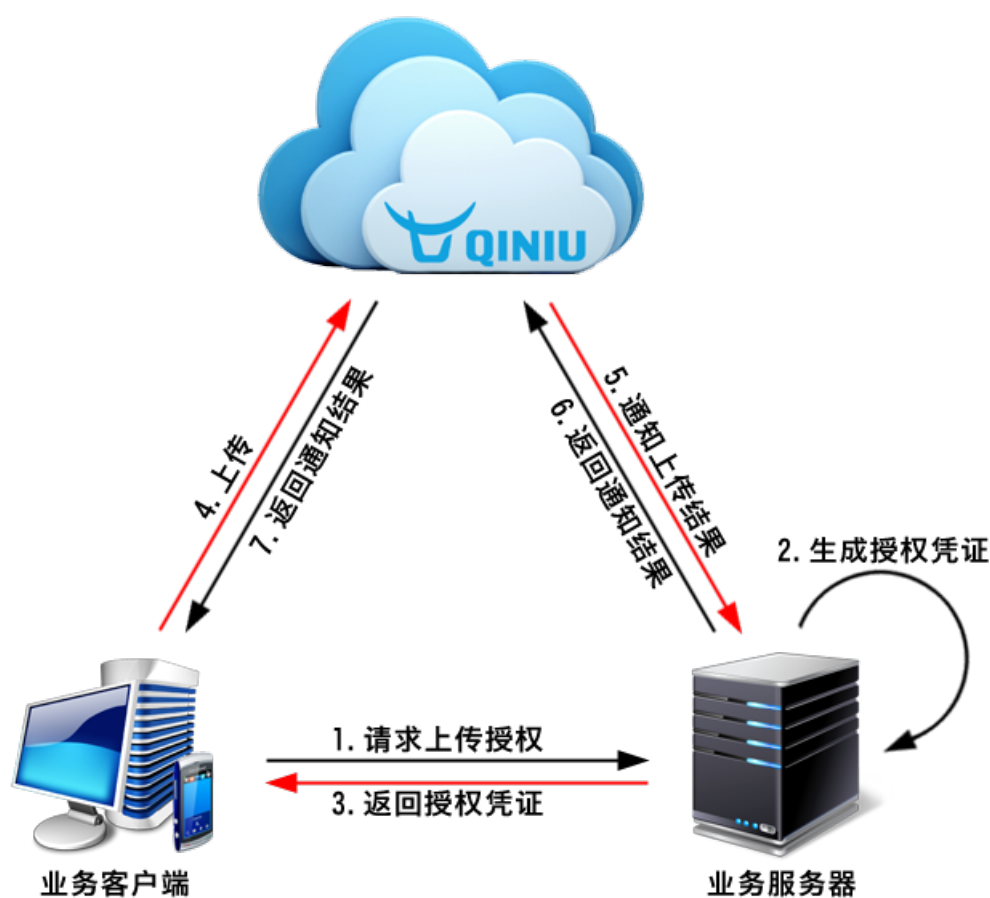
注意：红线为returnBody的传送轨迹。

1.3 七牛云通知上传模型

本模型同样由业务客户端将用户产生的内容（UGC）上传至七牛云，但改由七牛云将上传结果通知给业务服务器，并将最终结果转送给业务客户端。

■ 参与实体：七牛云、业务客户端、业务服务器。

■ 优点：尽可能避免业务客户端/服务器之间多次交互带来的状态不一致问题，降低可能引发的业务风险。



注意：红线为callbackBody的传送轨迹。

Chapter 2

上传方式

2.1 普通上传

最简单、最直观的上传方式，通过将单一资源封装在一次HTTP请求中传输至七牛云。

■ 优点：传输模型简单，API单一。

■ 缺点：对大型资源不友好，失败概率与资源大小成正比，纠错成本高（只能重传整个资源）。

■ 适用场景：大小固定的微型/小型资源上传、服务器之间资源传输。



2.2 断点续上传 (Block)

通过将资源切分为固定大小的数据块（称为Block，4MB大小），封装在多次HTTP请求中逐一传输至七牛云。

可根据实际需求灵活地调整为串行传输或并发传输。

■ 优点：对大型资源友好，纠错成本小（只重传失败的数据片，而不是数据块）。

■ 缺点：传输模型复杂，API多。

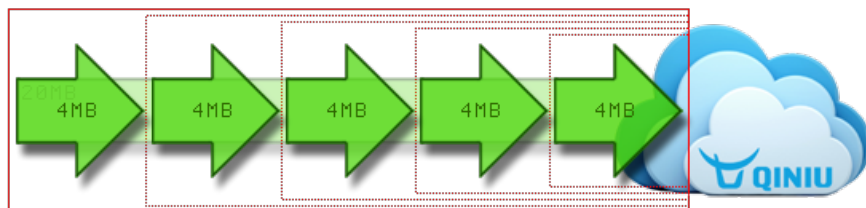
■ 适用场景：大小固定的中型/大型资源上传，弱信号通信线路的上传，频繁切换接入点/线路的上传。

2.2.1 串行传输

断点续上传的串行传输与单一资源的直传类似，只是传输单元更小、纠错成本更小。相对地，多次HTTP请求会带来传输成本的上升。

■ 优点：逻辑简单，实现容易。

■ 适用场景：需要快速编程实现业务逻辑的场景。



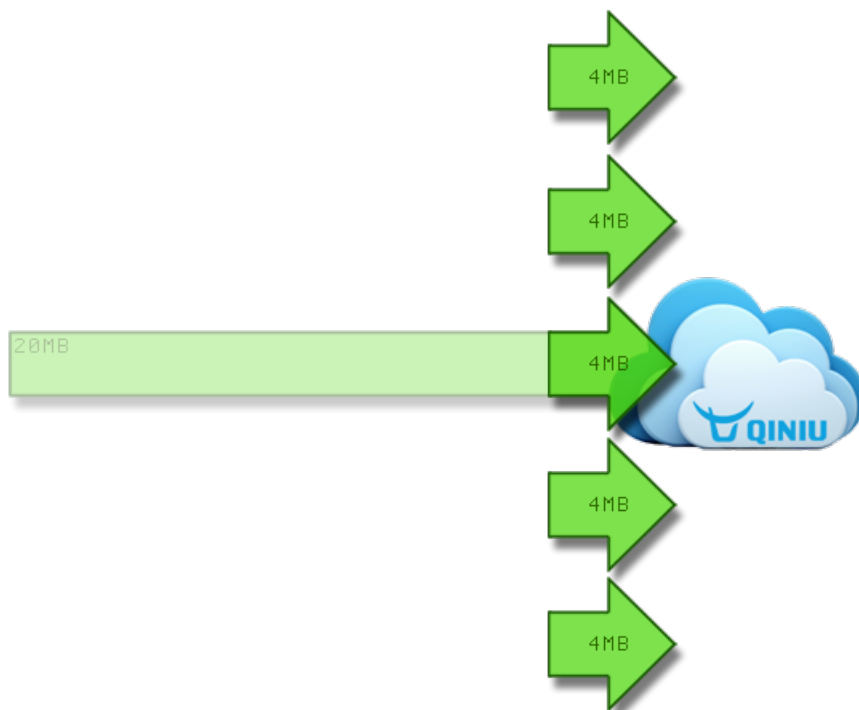
虚线框：上传成功的数据范围。

实线框：可以读取的数据范围。

2.2.2 并发传输

断点续上传的并发传输与多资源的并发直传类似，用于加快单一大型资源的传输速度。

- 优点：提供最高的带宽利用率。
- 缺点：涉及多线程技术，增加了调试难度。
- 适用场景：需要调优以提高带宽利用率的场景。

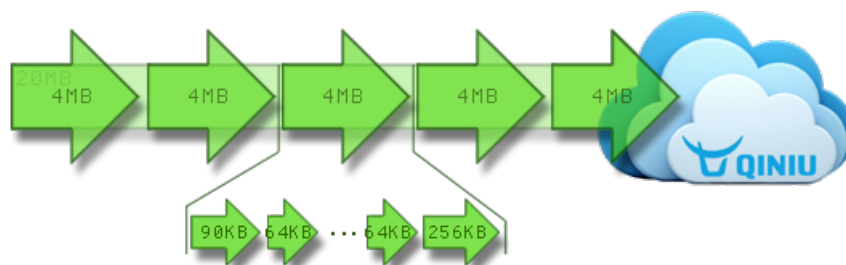


2.2.3 细粒度传输 (Chunk)

断点续上传的细粒度传输构建在对数据块的进一步切分之上，也即将数据块 (Block) 切分为非固定大小的数据片 (称为Chunk，小于4MB)，更灵活地贴合传输环境的限制，甚至智能调整传输速率以提供更好的用户体验。

每个数据片封装成一个HTTP请求，过小的数据片将带来额外的传输成本。另外数据片不能并发上传。

■ 适用场景：频繁切换接入点的环境，如地铁等信号弱、频繁重新接入网络的封闭空间。



2.3 追加上传 (Append, 研发中)

类似于操作系统打开文件时使用的Append参数，七牛云也提供相应语义的追加上传方式，在已保存文件的尾部追加数据。

本方式同样采用切分数据、逐一上传的思路，但与断点续上传不同的是a) 只切分成非固定大小的数据片 (Chunk)，b) 每个数据片上传后即可立刻读取。

■ 优点：资源大小无限制，支持流式传输，支持即写即读。

■ 缺点：传输模型复杂，不支持并发上传。

■ 适用场景：流式资源的上传，如监控音频/视频流、直播视频流存储，无长度限制的日志。



实线框：上传成功/可以读取的数据范围。

2.4 特性一览表

| 特性 | 普通上传 | 断点续上传 | 追加上传 |
|---------|---------|-------|---------|
| API数量 | 1 | 3 | 3 |
| 单次上传数据量 | 最大500MB | 最大4MB | 最大500MB |
| 单个资源数据量 | 最大500MB | 无限制 | 无限制 |
| 数据块并发上传 | × | √ | × |
| 即传即读 | × | × | √ |

Chapter 3

上传授权凭证

为防止恶意上传垃圾资源（消耗存储空间，引发额外计费），或篡改/损坏/伪造资源，每个上传请求都需要随附一份上传授权凭证（uploadToken），用于验证上传者与上传请求的合法性。同时，需要在该凭证内指定上传策略参数。

上传授权凭证通常由业务服务器签发，避免签名用的密钥泄露。

3.1 策略参数

■ 参数格式：JSON。

```
{
  "scope ":           "<Bucket                string >",
  "deadline ":        "<UnixTimestamp        int64 >",
  "insertOnly ":      "<AllowFileUpdating    int >",

  "saveKey ":         "<SaveKey                string >",

  "endUser ":         "<EndUserId            string >",

  "returnUrl ":       "<RedirectURL          string >",
  "returnBody ":      "<ResponseBodyForAppClient string >",

  "callbackUrl ":     "<RequestUrlForAppServer string >",
  "callbackHost ":    "<RequestHostForAppServer string >",
  "callbackBody ":    "<RequestBodyForAppServer string >",

  "persistentOps ":   "<persistentOpsCmds     string >",
  "persistentNotifyUrl ": "<persistentNotifyUrl   string >",
  "persistentPipeline ": "<persistentPipeline     string >",

  "fsizeLimit ":      "<FileSizeLimit        int64 >",

  "detectMime ":      "<AutoDetectMimeType    int >",

  "mimeLimit ":       "<MimeLimit            string >"
}
```

■ 参数说明:

| 名称 | 说明 | 备注 |
|----------------------|--|---------------------|
| scope | 指定存储目标 "<Bucket>": 目标资源空间, 如资源存在则失败 "<Bucket>:<Key>": 目标资源标识, 如资源存在则覆盖, 若希望上传指定key的文件, 并且不允许修改, 那么可以将下面的 insertOnly属性值设为 1 | 必填 |
| deadline | 凭证失效期限, UNIX Epoch格式 (绝对时间), 单位: 秒 | 必填 |
| insertOnly | 如果设置为非0值, 则无论scope设置为什么形式, 仅能以新增模式上传文件 | |
| saveKey | 自定义资源名, 支持魔法变量及自定义变量。这个字段仅当用户上传的时候没有主动指定key的时候起作用 | |
| endUser | 唯一属主标识, 用于特殊操作, 如作为业务客户端标识, 为图片或视频打水印 | |
| returnUrl | 指定上传成功后执行HTTP 301转跳的目标URL, 多用于HTML表单上传场景。最终URL为<returnUrl>?<returnBody> 在凭证中指定本参数会触发七牛云返回301响应 | 不可与 callback-Url共用 |
| returnBody | 指定上传成功后七牛云返回给业务客户端的上传结果。支持魔法变量与自定义变量 | 不可与 callback-Body共用 |
| callbackUrl | 指定上传成功后执行HTTP POST请求的目标URL, 多用于向业务服务器推送上传结果。必须是公网上能正常返回HTTP 200响应的URL | 不可与 returnUrl 共用 |
| callbackBody | 指定上传成功后推送的上传结果, 支持魔法变量与自定义变量 | 不可与 return-Body共用 |
| persisitentOps | 资源上传成功后触发执行的预转持久化处理指令列表, 多个指令以分号 (;) 分隔 | |
| persisitentNotifyUrl | 接收预转持久化结果通知的URL, 必须是公网上可以正常进行POST请求并能响应HTTP/1.1 200 OK的有效URL | |
| persistentPipeline | 转码队列名, 资源上传成功后, 触发转码时指定独立的队列进行转码。为空则表示使用公用队列, 处理速度比较慢。建议使用专用队列 | |
| fsizeLimit | 限定上传文件的大小, 单位: 字节 (Byte) ; 超过限制的上传内容会被判为上传失败, 返回413状态码 | |

| 名称 | 说明 | 备注 |
|------------|--|----|
| detectMime | <p>开启MimeType侦测功能</p> <p>设为非0值，则忽略上传端传递的文件MimeType信息，使用七牛服务器侦测内容后的判断结果</p> <p>默认设为0值，如上传端指定了MimeType则直接使用该值，否则按如下顺序侦测MimeType值：</p> <ol style="list-style-type: none"> 1. 检查文件扩展名； 2. 检查Key扩展名； 3. 侦测内容。 <p>如不能侦测出正确的值，会默认使用 <code>application/octet-stream</code></p> | |
| mimeLimit | <p>限定用户上传的文件类型，指定本字段值，七牛服务器会侦测文件内容以判断MimeType，再用判断值跟指定值进行匹配，匹配成功则允许上传，匹配失败返回403状态码。</p> <ol style="list-style-type: none"> 1. "image/*"表示只允许上传图片类型； 2. "image/jpeg;image/png"表示只允许上传`.jpg`和`.png`类型的图片； 3. "!application/json;text/plain"表示禁止上传`.json`文本和纯文本（注意最前面的感叹号） | |

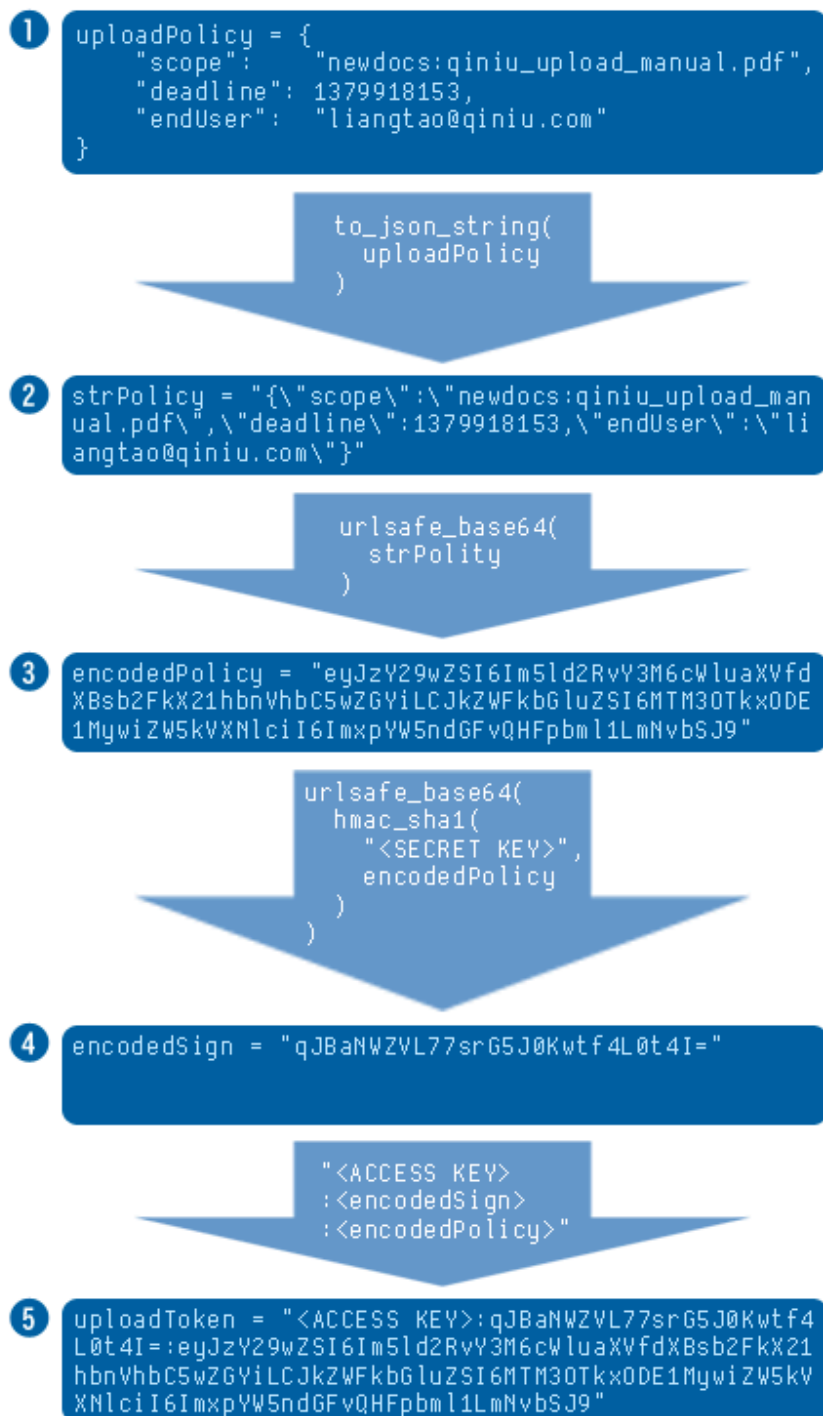
示例 3.1 uploadToken实例（覆盖语义）

```

{
  "scope": "qiniu-ts-demo",
  "deadline": 1390528576,
  "persistentOps": "avthumb/mp4;avthumb/m3u8/segtime/15/preset/video_440k",
  "persistentNotifyUrl": "http://fake.com/qiniu/notify"
}

```

3.2 签名算法



URL友好的Base64编码 (urlencode_base64)

SHA1摘要算法

HMAC签名算法

3.3 魔法变量 (Magic Variables)

魔法变量是七牛云与用户交换数据的一种占位符机制，在业务服务器生成的 `returnBody` 或 `callbackBody` 中占据某一位置形成模板，由七牛云在恰当时机求值并填入对应反馈信息。具体传递方法参看 [示例](#)。

■ 书写形式：类似变量引用与属性引用，区分大小写。

```
$(MagicVariable)
$(MagicVariable.Field)
$(MagicVariable.Field.SubField)
...
```

■ 通用变量一览：

| 名称 | 说明 |
|--------------|---|
| bucket | 上传目标资源空间的名字 |
| etag | 资源上传成功后的ETag。上传时不指定资源名，其值为缺省资源名（由七牛云自动推导） |
| fname | 原始文件名 |
| fsize | 资源大小，单位：字节（Byte） |
| mimeType | 资源类型，比如 <code>.jpg</code> 图片的资源类型为 <code>image/jpg</code> 。上传时不指定资源类型，其值为缺省资源类型（由七牛云自动推导） |
| endUser | <code>uploadToken</code> 中指定的 <code>endUser</code> 值，即唯一属主标识 |
| persistentId | |

■ 图片资源相关变量一览：

| 名称 | 说明 |
|------------------------|---|
| imageInfo | 图片基本信息 |
| imageInfo.format | 图片格式 |
| imageInfo.width | 图片宽度，单位：像素（Pixel） |
| imageInfo.height | 图片高度，单位：像素（Pixel） |
| imageInfo.colorModel | 色彩模型 |
| exif | EXIF信息 更多子项请参考 EXIF白皮书 |
| exif.ApertureValue | 光圈值 |
| exif.ColorSpace | 色彩模型 |
| exif.DateTime | 拍摄时间 |
| exif.ExifVersion | EXIF版本 |
| exif.ExposureBiasValue | 曝光偏差值 |
| exif.ExposureMode | 曝光模式 |
| exif.ExposureTime | 曝光时间 |
| exif.FNumber | 光圈号 |
| exif.Flash | 闪光灯状态 |
| exif.FocalLength | 焦距 |
| exif.ISOSpeedRatings | ISO值 |
| exif.Model | 设备型号 |
| exif.WhiteBalance | 白平衡模式 |
| exif.XResolution | 横向分辨率 |
| exif.YResolution | 纵向分辨率 |

3.4 自定义变量 (X-Variables)

自定义变量是七牛云提供的另一种占位符机制，用于协助业务客户端与业务服务器之间交换信息。业务服务器负责生成 `returnBody` 或 `callbackBody` 模板，由七牛云填入业务客户端传回的对应信息后返回给业务服务器。借由这一机制可以减少不必要的交互，降低失败风险，减少流量。

自定义变量的名称由七牛用户自行定义，并以 `x:` 打头（区分大小写），如 `x:location` 和 `x:price`。具体传递方法参看[示例](#)。

Chapter 4

API

七牛云的资源上传API基于HTTP协议，通过HTTP POST方法实现。

上传参数与数据采用multipart/form-data格式组织，便于使用HTML表单直接上传文件的同时也适用于其它场景。

4.1 直传

4.1.1 直传单一资源（文件）

■ 策略参数：

```
{
  "scope": "demoshow:test2.txt",
  "deadline": 1379918153
}
```

■ 请求：

```
POST / HTTP/1.1
Connection: close
Content-Length: 619
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: multipart/form-data; boundary=763C032728D21BC3D50ED93F1965413BC37B25B4
Host: up.qiniu.com

--763C032728D21BC3D50ED93F1965413BC37B25B4
Content-Disposition: form-data; name="token"

u8WqmQuijH21kxpIQmo2LqntzugM1VoHE9_pozCU:wj8BDLCrYgiKVfhUIxU01iF3k0g=:eyJkZWZkbG1...
--763C032728D21BC3D50ED93F1965413BC37B25B4
Content-Disposition: form-data; name="key"

test2.txt
--763C032728D21BC3D50ED93F1965413BC37B25B4
Content-Disposition: form-data; name="file"; filename="test2.txt"
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

This is a plain text file for testing.
Again.

--763C032728D21BC3D50ED93F1965413BC37B25B4--
```

■ 参数说明:

| 协议层次 | 名称 | 说明 | 备注 |
|------|--------------|---|--------------|
| HTTP | Content-Type | 指定参数与数据组织格式 固定值: multipart/form-data | |
| HTTP | User-Agent | 发起端身份标识, 可用于防盗链、日志分析 | SDK 可设定特殊标识符 |
| HTTP | Host | 上传服务器主机域名 固定值: up.qiniu.com | |
| API | token | 上传授权凭证 | |
| API | key | 资源名 | |
| API | file | 文件内容 | |

■ 响应:

```
HTTP/1.1 200 OK
Date: Sun, 29 Sep 2013 05:09:05 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Content-Length: 57
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: BDT;BDT;LBD:1;rs.put:15;rs-upload.putFile:59;UP:60
X-Reqid: KTOAAMALhqX-RigT
```

```
{"hash": "Fre2JqI866GUYQQUpfy6lvE792Ka", "key": "test2.txt"}
```

■ 返回值说明:

| 协议层次 | 名称 | 说明 | 备注 |
|------|---------|-----------------------------------|----|
| API | X-Log | 七牛云用于跟踪错误的微型日志 ★可作为查错线索提交给客服人员 | |
| API | X-Reqid | 七牛云用于跟踪错误的流水号 ★可作为查错线索提交给客服人员 | |
| API | hash | ETag值, 用于追踪资源版本 | |
| API | key | 最终资源名 | |

4.1.2 直传单一资源（图片+魔法变量+自定义变量）

■ 策略参数:

```
{
  "scope": "demoshow:gogopher.jpg",
  "deadline": 1379918153,
  "returnUrl": "http://fake.com/path/to/return",
  "returnBody": "w=$(imageInfo.width)&h=$(imageInfo.height)&t=$(x:tag)"
}
```

■ 请求:

```
POST / HTTP/1.1
Connection: close
Content-Length: 215348
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: multipart/form-data; boundary=59DD4152B8BEADAAA35AAAA0D8C5A90F75AAFEF8
Host: up.qiniu.com
```

```
--59DD4152B8BEADAAA35AAAA0D8C5A90F75AAFEF8
Content-Disposition: form-data; name="token"

u8WqmQuijH21kxpIQmo2LqntzugM1VoHE9_pozCU:UZM2jTH7J97jzIN7fVS2BJx_bdQ=:eyJyZXR1cm5...
--59DD4152B8BEADAAA35AAAA0D8C5A90F75AAFEF8
Content-Disposition: form-data; name="key"
```

```
gogopher.jpg
--59DD4152B8BEADAAA35AAAA0D8C5A90F75AAFEF8
Content-Disposition: form-data; name="x:tag"
```

```
gopher
--59DD4152B8BEADAAA35AAAA0D8C5A90F75AAFEF8
Content-Disposition: form-data; name="file"; filename="gogopher.jpg"
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

```
JFIFHHXICC_PROFILEHLinomntrRGB XYZ ...
```

■ 响应:

```
HTTP/1.1 301 Moved Permanently
Date: Sun, 29 Sep 2013 09:00:20 GMT
Content-Type: text/plain; charset=utf-8
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Location: http://fake.com/path/to/return?upload_ret=dz02NDAmAD00MjcmdD0iZ29waGVyIg==
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: BDT:4;BDT:5;LBD:6;rs.put:14;rs-upload.putFile:22;UP:23/301
X-Reqid: KTOAAHjM7hidUygT
Content-Length: 0
```

■ responseBody:

```
dz02NDAmAD00MjcmdD0iZ29waGVyIg==
Base64 decode =>
w=640&h=427&t="gopher"
```

4.2 断点续上传

4.2.1 微型资源上传

对于小于4MB的微型资源，只需要上传一个数据块并合成文件即可。

由于一次完整的断点续上传至少涉及2次HTTP交互，共计4个HTTP报文，因此建议根据应用场景灵活决定某个大小以下的资源改用直传，减少交互次数。

本例不涉及多个数据片上传。

■ 策略参数：

```
{
  "scope": "demoshow:test2.txt",
  "deadline": 1379918153,
}
```

■ 请求样例1，创建数据块上传会话，传递首个数据块大小与其第一个数据片：

```
POST /mkblk/46 HTTP/1.1
Connection: close
Content-Length: 46
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: application/octet-stream
Host: up.qiniu.com
Authorization: UpToken u8WqmQuijH21kxpIQmo2LqntzugM1VoHE9_pozCU:Hh2Fi_7KmGEtiqXFF...
```

This is a plain text file for testing.
Again.

● 接口URL格式：

http://up.qiniu.com/mkblk/<blockSize>

● 参数说明：

| 协议层次 | 名称 | 说明 | 备注 |
|------|----------------|--|-------|
| HTTP | Content-Length | (第一个) 数据片大小 | |
| HTTP | Content-Type | 指定上传数据块的内容类型 固定值：application/octet-stream | |
| HTTP | Host | 上传服务器主机域名 固定值：up.qiniu.com | |
| HTTP | Authorization | 上传授权凭证 格式：UpToken <授权凭证> | |
| API | /mkblk | 上传指令：创建一个数据块上传会话 | |
| API | /46 | 上传数据块大小 | 最大4MB |
| API | This is a... | 数据片内容 | |

■ 响应样例1，收存第一个数据片，返回上传会话信息：

```
HTTP/1.1 200 OK
Server: nginx/1.0.8
Date: Tue, 08 Oct 2013 08:05:16 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Content-Length: 323
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: BDT;BDT;LBD:1;UP:4
X-Reqid: KTOAEC4BhvVEysT

{"ctx":"o0Fl-Y7dIZ...", "checksum":"t7YmojzroZ...", "crc32":2035613763,
"offset":46, "host":"http://up-nb-5.qbox.me"}
```

● 返回值说明：

| 协议层次 | 名称 | 说明 | 备注 |
|------|----------|----------------------------------|---------|
| API | ctx | 已传部分的上传上下文信息 | 仅供服务器使用 |
| API | checksum | 已传部分的校验信息 | 仅供服务器使用 |
| API | crc32 | 已传部分的CRC32校验值 可与本地计算的CRC32值对比 | |
| API | offset | 已传部分的偏移值，单位：Byte（字节） | |
| API | host | 下一请求报文使用的主机域名 | |

■ 请求样例2，合成资源，传递资源大小、资源名和上传上下文信息：

```
POST /mkfile/46/key/dGVzdDIudHh0 HTTP/1.1
Connection: close
Content-Length: 208
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: text/plain
Host: up-nb-5.qbox.me
Authorization: UpToken u8WqmQu1jH21kxpIQmo2LqntzugM1VoHE9_pozCU:Hh2Fi_7KmGEtiqXFF...

oOFI-Y7dIZH6cPkXfuPWYP9jZCZxpFONALRhFDlUpuQAASNFZ4mrze_-3LqYdlQyEPDh0sNUaG1zIG1zI...
```

● 接口URL格式：

```
http://<selectedUpHost>/mkfile/<blockSize>
                                   /key/<urlSafeBase64EncodedKey>
                                   /x:userVariable1/<urlSafeBase64EncodedVariable1>
                                   /x:userVariable2/<urlSafeBase64EncodedVariable2>
                                   ...
                                   /x:userVariableN/<urlSafeBase64EncodedVariableN>
```

● 参数说明：

| 协议层次 | 名称 | 说明 | 备注 |
|------|----------------------|---|---------------------|
| HTTP | Content-Type | 指定上传上下文的内容类型 固定值：text/plain | |
| HTTP | Host | 上传服务器主机域名 可变值：mkblk响应报文中的host字段值 | |
| HTTP | Authorization | 上传授权凭证 格式：UpToken <授权凭证> | 即mkblk请求报告中的授权凭证 |
| API | /mkfile | 上传指令：根据传递的上传上下文序列，合成资源并指定资源名 | |
| API | /46 | 资源大小 | |
| API | / key/ dGVzd-DIudHh0 | 指定资源名 | 需要编码为URL友好的Base64格式 |
| API | oOFI-Y7dIZ... | 各个数据块的上传上下文列表，以逗号(,)分隔 按数据块在资源中的位置排序 | |

■ 响应样例2, 合成资源:

```
HTTP/1.1 200 OK
Server: nginx/1.0.8
Date: Tue, 08 Oct 2013 08:05:16 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Content-Length: 57
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: rs.put:7;mkfilev2.onFilePutted:7;UP:7
X-Reqid: KTOAANDC-iLVEysT
```

```
{"hash": "Fre2JqI866GUYQQUpfy6lvE792Ka", "key": "test2.txt"}
```

● 返回值说明:

| 协议层次 | 名称 | 说明 | 备注 |
|------|------|-----------------|----|
| API | hash | ETag值, 用于追踪资源版本 | |
| API | key | 最终资源名 | |

4.2.2 大型资源上传 (6MB)

对于超过4MB的资源，断点续上传要求按4MB为单位将资源分割成数据块（Block），并进一步切分成数据片（Chunk）上传。

数据块可以并发上传，而同一个数据块的数据片只能串行上传。

数据片的大小可根据实际情况动态调整。

本例涉及到两个数据块串行上传。

■ 策略参数:

```
{
  "scope": "demoshow:zeros",
  "deadline": 1379918153,
}
```

■ 请求样例1，创建1号数据块上传会话，传递首个数据块大小与其第一个数据片:

```
POST /mkblk/4194304 HTTP/1.1
Connection: close
Content-Length: 262144
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: application/octet-stream
Host: up.qiniu.com
Authorization: UpToken u8WqmQu1jH21kxpIQmo2LqntzugM1VoHE9_pozCU:iUhDkt1IhUazVkJLP7...

...
```

● 参数说明:

| 协议层次 | 名称 | 说明 | 备注 |
|------|----------------|---------|-------|
| HTTP | Content-Length | 上传数据片大小 | |
| API | /4194304 | 上传数据块大小 | 最大4MB |

■ 响应样例1，收存第一个数据片，返回1号上传会话信息:

```
HTTP/1.1 200 OK
Server: nginx/1.0.8
Date: Fri, 11 Oct 2013 07:25:53 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Content-Length: 327
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: UP:4
X-Reqid: IxwAAPh6wJFs_SsT

{"ctx":"uIBWEEiv8J...", "checksum":"LgAPp-hXWc...", "crc32":3792628258,
"offset":262144, "host":"http://up-nb-5.qbox.me"}
```

■ 请求样例2，传递1号上传上下文、已传大小与第N个数据片:

```
POST /bput/uIBWEEiv8J.../262144 HTTP/1.1
Connection: close
Content-Length: 262144
```

```
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: application/octet-stream
Host: up-nb-5.qbox.me
Authorization: UpToken u8WqmQuijH21kxpIQmo2LqntzugM1VoHE9_pozCU:iUhDkt1IhUazVkJLP7...
```

...

● 参数说明:

| 协议层次 | 名称 | 说明 | 备注 |
|------|---------------|------------------------------------|-------|
| HTTP | Host | 服务器返回的下一上传服务器主机域名 保证会话维持在相同服务器上 | |
| API | /bput | 上传指令: 上传数据片 | 最大4MB |
| API | /uIBWEEiv8J.. | 已传部分的上传上下文 | |
| API | /262144 | 上传偏移位置 | |

■ 响应样例2, 收存第N个数据片, 返回1号上传会话信息:

```
HTTP/1.1 200 OK
Server: nginx/1.0.8
Date: Fri, 11 Oct 2013 07:25:53 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Content-Length: 327
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: UP:7
X-Reqid: IxwAAEgEC6ps_SsT

{"ctx": "JpQn4L-Kz3...", "checksum": "a1IeHSpjLC...", "crc32": 3792628258,
"offset": 524288, "host": "http://up-nb-5.qbox.me"}
```

● 返回值说明:

| 协议层次 | 名称 | 说明 | 备注 |
|------|-----|--------------------------------|----|
| API | ctx | 已传部分的上传上下文, 随着成功上传 内容的增多而变化 | |

■ 请求样例3, 创建2号数据块上传会话, 传递第二个数据块大小与其第一个数据片:

```
POST /mkblk/2097152 HTTP/1.1
Connection: close
Content-Length: 262144
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: application/octet-stream
Host: up-nb-5.qbox.me
Authorization: UpToken u8WqmQuijH21kxpIQmo2LqntzugM1VoHE9_pozCU:iUhDkt1IhUazVkJLP7...
```

...

● 参数说明:

| 协议层次 | 名称 | 说明 | 备注 |
|------|--------------------------------|---|----|
| HTTP | Content-Length | 上传数据片大小 | |
| HTTP | Host | 服务器返回的下一上传服务器主机域名 保证所有数据块会话维持在相同服务器上 | |
| API | /262144 | 上传数据块大小 | |

■ 响应样例3，收存第1个数据片，返回2号上传会话信息：

```
HTTP/1.1 200 OK
Server: nginx/1.0.8
Date: Fri, 11 Oct 2013 07:26:06 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Content-Length: 327
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: UP:7
X-Reqid: IxwAANjV1ohv_SsT

{"ctx":"cXa9tl8idv...", "checksum":"LgAPp-hXWc...", "crc32":3792628258,
"offset":262144, "host":"http://up-nb-5.qbox.me"}
```

■ 请求样例4，合成资源，传递资源大小、资源名和上传上下文信息列表：

```
POST /mkfile/6291456/key/emVy3M= HTTP/1.1
Connection: close
Content-Length: 417
User-Agent: Easy-Qiniu-Perl-SDK/0.1
Content-Type: text/plain
Host: up-nb-5.qbox.me
Authorization: UpToken u8WqmQu1jH21kxpIQmo2LqntzugM1VoHE9_pozCU:iUhDkt1IhUazVkJLP7...

TonL6i-rCDe02rRnbqZmYXnGiJNc1QxEUrPuZU-gEpcAmGfdaoqgkYoSr7qnyLQMFHu428MAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAQAAAAAA
lAkAAPfWV1IAAEAAAABAAAAAAAVAP____8=,pqQAr1UMKPQQLCBHnt0oGNE9oPVhy1uEXUsaMiHI_Y4AD1VfP
YZPpHlufFxJnab8i8wECL8AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAVAP____8=
```

■ 响应样例4，合成资源：

```
HTTP/1.1 200 OK
Server: nginx/1.0.8
Date: Fri, 11 Oct 2013 07:26:10 GMT
Content-Type: application/json
Connection: close
Cache-Control: no-store, no-cache, must-revalidate
Content-Length: 53
Pragma: no-cache
X-Content-Type-Options: nosniff
X-Log: rs.put:11;mkfilev2.onFilePutted:12;UP:17
X-Reqid: IxwAADCc4YZw_SsT

{"hash":"lvxwSaB2VXJaY8dXRiat4RlrTPTZ", "key":"zeros"}
```

4.3 追加上传

待续……

Chapter 5

FAQ

- 资源名可以是中文吗？

资源名可以是中文，上传前需要转换成UTF-8编码。

- 为什么上传授权凭证会经常失效？

上传授权凭证使用的过期时间是绝对时间，需要将本地机器的时间校正为标准UTC时间（并设置为东八区）。

Chapter 6

术语

- **资源**

存入七牛云的数据体的统称，在有限前提下可与“文件”互换。

- **七牛云**

所有对外放开的七牛服务及相关系统的统称，含云存储、云处理和云加速三大子系统。
本文通篇使用“七牛云”指代“七牛云存储子系统”这一实体。

- **业务服务器**

七牛用户自建的、用于运行业务逻辑的服务器实体。

- **业务客户端**

七牛用户的用户（即最终用户）、用于运行交互逻辑的客户端实体。

- **高速线路**

机房（或IDC）之间使用的通信线路，具有低延迟（20~50毫秒）、极低丢包率、中间路由少或转发效率高、上下行带宽对等等特点。

注意：高速通信线路可能跨越不同运营商的异构网络。

- **低速线路**

机房（或IDC）与业务客户端之间使用的通信线路，具有中等延迟（50~100毫秒）、低丢包率、偶发故障、偶发线路切换、上下行带宽不对等等特点。

- **客户端接入环境**

业务客户端接入互联网的环境，常见的有LAN（局域网）、WLAN/Wi-Fi（无线局域网）、2G/3G（移动网）等。

注意：接入环境会随着客户端物理位置随时发生改变，如离开房屋导致从Wi-Fi环境切换到3G环境，跨越3G信号塔覆盖区域导致在不同的接入基站之间切换。