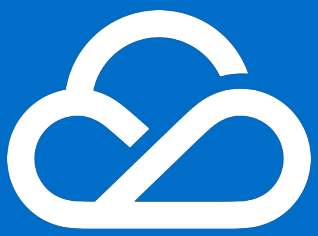
万象优图

用户指南

产品文档



腾讯云

【版权声明】

【商标声明】

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

如何接入

1 基本架构及业务流程

1.1基本架构

数据安全是云存储的重中之重，为了保护开发者的数据安全，腾讯云·万象优图的上传、复制、删除和下载（如果设置了防盗链）服务需要鉴权签名；鉴权签名的生成需要用到SecretKey，所以签名的生成需要在可信任的环境中进行，绝不能在客户端进行，否则会带来密钥泄漏的风险，危害开发者数据。腾讯云·万象优图推荐使用以下的服务器架构设计：

主要包括三个部分：

1 腾讯云·万象优图

提供高成功率、高可靠、高速的图片存储服务，以及强大的图像处理技术；

2. 开发者服务器

开发者服务器至少提供如下3个功能：

1)生成签名。签名的生成不能在终端上进行，否则会产生极大的安全隐患；
2)使用数据库管理用户信息、图片资源信息；
3)响应终端的业务请求；

另外，开发者服务器还可以直接与腾讯云·万象优图通信对图片进行管理操作。

3. 终端

通常既上传图片也下载图片。在展示图片前，通常先从开发者服务器获取要展示的图片信息，比如图片的url；在上传图片前，通常先从开发者服务器获取签名，然后带着签名请求腾讯云·万象优图。

1.2 业务流程

1.2.1 图片上传

终端上传图片前先请求开发者服务器获取上传签名，然后再请求腾讯云·万象优图，腾讯云·万象优图会对上传签名进行验证，无效的签名会返回签名认证错误等信息。图片上传流程根据开发者是否设置回调可分为两种方案。

方案一：无回调设置

上传流程主要包括三个步骤：

1. 终端访问开发者服务器，获取签名（注意：签名的创建不能在终端上进行，否则可能会导致用户信息泄露等安全隐患）；
2. 终端上传图片，腾讯云·万象优图验证签名、存储图片，生成文件ID，URL等信息返回给终端；
3. 终端将图片信息和用户信息反馈给开发者服务器。

方案二：有回调设置

如果有设置回调，则上传完成时腾讯云·万象优图会自动向开发者服务器发起设置的回调url请求。

上传流程主要包括四个步骤：

1. 终端访问开发者服务器，获取签名（注意：签名的创建不能在终端上进行，否则可能会导致用户信息泄露等安全隐患）；
2. 终端上传图片，腾讯云·万象优图验证签名、存储图片，生成文件ID，URL等信息通过回调url传给开发者服务器；
3. 业务服务存储相应的图片信息，并将处理结果返还给腾讯云·万象优图；
4. 腾讯云·万象优图根据开发者服务器返回的处理信息返还给终端结果。

1.2.2 图片下载

图片下载使用图片上传成功后返回的的下载url直接访问，即直接访问download_url。

如果设置了Token防盗链，则使用下载url加签名的方式直接访问，即：download_url?sign=[签名]

2 接入服务

根据开发者业务的现有情况可将服务接入分为两种类型：回源镜像和一般接入。其中回源镜像适用于开发者的历史图片没有存储在腾讯云·万象优图，而要使用腾讯云·万象优图的下载（和其他服务，如果开发者希望接入）服务；一般接入是指开发者没有或者不考虑历史图片，使用腾讯云·万象优图对图片进行上传、下载和其他操作。

> 注：回源镜像可以用作开发者图片数据的灾备；也可以让开发者一键试用服务，方便快捷。

2.1 回源镜像

2.1.1 注册账号

前往腾讯云·万象优图控制台注册账号。

> 注：以下称图片空间为bucket。

2.1.2 场景接入

腾讯云·万象优图根据开发者的源站类型和所需服务，将回源镜像划分为四个场景。

> 注意： 腾讯云·万象优图针对四个场景分别进行了优化，请根据自身的情况选择合适场景接入，获取最优的服务体验。

场景一：使用万象优图下载，回源到其他云存储厂商

场景二：使用万象优图下载，回源到自己站点

场景三：使用万象优图上传和下载，历史图片回源到其他云存储厂商

场景四：使用万象优图上传和下载，历史图片回源到自己站点

2.2 一般接入

由2.1.1 基本架构章节可知，开发者如需接入腾讯云·万象优图服务，为了保护数据的安全，开发者需要在自己开发者服务器上面部署鉴权服务；终端需要向开发者服务器请求的签名，然后进行相应的服务操作，例如进行图片的上传，复制，删除，下载等。

2.2.1 接入流程

接入腾讯云·万象优图只需以下三步：

1 注册腾讯云·万象优图；

2 终端部署示例；

3 鉴权服务部署示例。

下面分别介绍终端部署简单示例，鉴权服务部署简单示例，以及终端和开发者服务器交互签名的过程。

说明：

1. 文档中终端和开发者服务器的签名交互过程只是一个简单的例子，开发者需要根据自己业务需求开发交互过程。
2. 如果开发者想使用本文档提供的例子进行简单测试，请确保终端和服务端的项目信息一致，如项目ID，空间名称，SecretID和SecretKey等。

2.2.1.1 注册腾讯云·万象优图

1. 前往腾讯云·万象优图控制台注册账号；如果已经注册账号，请跳过此步骤；

2. 在腾讯云·万象优图控制台
   创建一个图片空间，获取空间名称（bucket）和项目ID；如果已创建过图片空间，请跳过此步骤；

3. 在腾讯云·万象优图控制台项目设置
   中添加密钥，获取SecretID和SecretKey；如果已经添加过密钥，请跳过此步骤。

2.2.1.2 终端部署示例

腾讯云·万象优图提供了丰富的Restful API接口，开发者可以参考Restful API部署和开发终端。
同时腾讯云·万象优图还为开发者提供了功能丰富的移动端SDK，以及web端部署示例。开发者也可是使用移动端SDK开发自己的服务，下面简单介绍使用终端SDK部署服务的简单示例，开发者需要根据自身业务情况开发相应的代码：

终端部署示例-Android

终端部署示例-iOS

web端部署示例

移动端SDK的详细文档说明请参见移动端SDK文档。

2.2.1.3 鉴权服务部署示例

以下为鉴权服务部署的简单示例(使用了服务端SDK)，开发者可以简单参考，开发自身业务鉴权服务，并集成到自身服务器中：

注意：下面的文档只是简单的示例，展示了服务端为终端提供签名的简单用法，开发者务必根据自身业务开发相应的鉴权服务逻辑，并集成到自身服务器中。

鉴权服务部署示例-Java

鉴权服务部署示例-PHP

鉴权服务部署示例-Python

鉴权服务部署示例-Nodejs

鉴权服务部署示例-GO

如需使用服务端更多功能，请参见服务端SDK文档。

3 历史数据迁移

在使用万象优图以前，您的图片数据可能存在服务器本地文件系统、分布式文件系统以及其他云存储等，我们提供迁移工具，方便您将这些历史图片迁移到万象，目前支持3种存储方式的迁移：

1. 图片在服务器本地存储，直接将某目录下的所有文件上传到万象优图。
2. 指定URL列表文件，文件中每一行是一张图片的URL。工具会下载列表文件中的每一张图片并上传到万象优图。
3. 指定七牛云存储的账号和空间名，迁移该指定空间中的所有文件或部分文件到万象优图。
   本工具目前支持类Unix操作系统。推荐在Linux或Mac OS X下使用Python 2.7运行。

3.1 迁移工具下载地址

https://github.com/tencentyun/Cloud-Image-Migration-Tool

3.2 迁移工具安装使用方法

从github上通过git获取，或者下载压缩包解压到linux服务器上，然后进入工具目录

```
cd Cloud-Image-Migration-Tool/bin
```

3.2.1 启动迁移

运行start.sh脚本，工具会按照配置文件中
的配置开始运行.启动前请修改配置，否则会运行失败。如果要迁移的文件较多，可后台运行，运行命令：

```
./start.sh &
```

上传完成后再次运行start.sh会重试失败的任务。

start.sh脚本首先扫描待上传的任务提交到任务列表，然后开始上传。如果第一个过程被打断，下次运行start.sh会重新扫描待上传的任务；如果第二个过程被打断，下次运行start.sh会继续上传。如果需要强制刷新任务列表请运行以下命令：

```
./start.sh -f &
```

3.2.2 查看迁移进度

启动后可查看迁移状态，运行stat.sh脚本，会在屏幕上持续输出迁移状态，按Ctrl+c退出查看。

```
./stat.sh
```

该脚本输出内容如下图：

数字有三列，第一列：失败的文件数；第二列：成功迁移的文件数；第三列：要迁移的文件总数

```
[root@VM_10_94_centos bin]# ./stat.sh
failed, successful / submitted: 1122, 0 / 1122
```

3.2.3 停止迁移

若启动后想做配置调整或任何异常发生，可停止迁移，运行stop.sh。

```
./stop.sh
```

在迁移过程中运行stop.sh脚本停止迁移，这一过程可能需要等待几秒钟的时间以保证已经开始的任务正常结束并写入日志。如果需要强制停止，运行stop.sh -f。

停止迁移之后如果没有清空日志并且没有修改配置信息，再次启动迁移会继续上次的迁移任务，重试所有失败记录但已经迁移成功的文件不会重传。

3.2.4 获取失败信息

获取迁移失败的文件列表及信息，运行view_failed.sh。

```
./view_failed.sh
```

3.2.5 清除日志

若想开始一次新迁移，可清空之前的日志信息，运行clean.sh。请慎用，因为日志文件会作为增量断点续传的参考信息，删除后再执行相同任务所有文件会重新上传。

3.3 修改配置

配置文件为config.ini位于conf目录, 配置文件中section name和option name不区分大小写。

```
vi Cloud-Image-Migration-Tool/conf/config.ini
```

3.3.1 迁移类型设置

设置migrate.type 为

Local 表示迁移linux本地文件系统的文件
URLList 表示迁移url列表文件中所有url指向的文件
Qiniu 表示迁移七牛某空间的文件

```
[MigrateInfo]
migrate.type = Local
```

3.3.1.1 本地文件系统

若迁移类型为Local，则需要配置上传的本地根目录位置，必须为绝对路径。上传后的file id不包含根目录。例如：

```
[Local]
local.image_root_path = /root/data/images/
```

迁移后图片的访问路径：

http: //上传的万象空间域名/图片相对于所配置的根目录的相对路径

比如，设置的本地目录：/data/web/images
有一张图片路径为：/data/web/images/2015/07/17/abc.jpg，上传后访问路径：http: //上传的万象空间域名/2015/07/17/abc.jpg

3.3.2 URL列表文件

若迁移类型为URLList，则需要配置URL列表文件的位置，必须为绝对路径。例如：

```
[URLList]
```

```
url.url_list_file_path = /data/url_list
```

迁移后图片的访问路径：

http: //上传的万象空间域名/原url的url路径

比如，原url为http: //www .xxx.yyy .com/2015/07/17/abc.jpg，上传后访问路径：http: //上传的万象空间域名/2015/07/17/abc.jpg

> 注意：若原url中有参数，上传后的文件名是去掉参数后的，比如，原url为http:// www .xxx. yyy. com/2015/07/17/abc.jpg?width=1024&height=1024，上传后访问路径：http: //上传的万象空间域名/2015/07/17/abc.jpg

3.3.3 七牛

若迁移类型为Qiniu，则需要配置七牛云存储账号相关信息。

qiniu.bucket填写被迁移的空间名；qiniu.domain为七牛域名，需要包含协议类型（如http:// ）； 若只迁移一部分则需要提供qiniu.start_offset和qiniu.total_num，qiniu.start_offset从0计；若空间开启了防盗链，需要在qiniu.referer中指定访问来源域名，同样需要包含协议类型（如http:// ）；
如果是私有空间，将qiniu.isprivate设置为True，否则设置为False。

```
[Qiniu]
[Qiniu]
qiniu.bucket = my_bucket_name
qiniu.AccessKey = _17terLxP-ZK7tma9jXgm7MuEOk72yP9OZBIP35G
qiniu.SecretKey = PFw6JivhTAdNKRojaguUkC6tlFHAI9SBrjVYdfya
qiniu.domain = http://abcde.com1.fg.glb.clouddn.com/
qiniu.referer =
qiniu.isprivate = False
```

迁移后图片的访问路径：

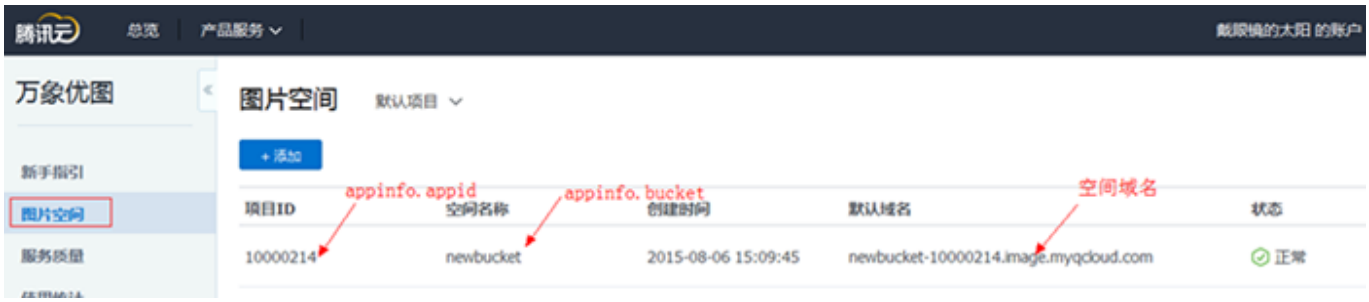
http: //上传的万象空间域名/原七牛key

比如，原图片下载路径为http:// 7x3m4e .com 1.z0.glb.clouddn.com /2015/07/17/abc.jpg，
上传后访问路径：http: //上传的万象空间域名/2015/07/17/abc.jpg

3.3.4 万象优图账号信息

从万象优图图片空间中查看项目ID和空间名称，分别填写到appinfo.appid和appinfo.bucket；

从万象优图项目设置中查看Secret ID和Secret Key分别填写到appinfo.secretID和appinfo.secretKey。




```
[AppInfo]
appinfo.appid =
appinfo.secretID =
appinfo.secretKey =
appinfo.bucket =
```

3.3.5 上传设置

Concurrency控制同时运行的上传进程数目，请根据上行带宽和机器配置适当调整该数值。必须提供一个大于0的整数。

db.commit.interval控制上传结果写回日志的时间间隔，单位为秒，应该提供一个合法的正整数或浮点数或正

无穷（inf）。如果进程意外结束，未写回的日志将丢失。此值设置过小将严重影响性能，一般情况下使用默认值即可。

jobqueue.capacity、jobqueue.reload.threshold、buffer.size是控制任务队列和内存缓冲区的选项，一般情况下使用默认值即可。

```
[ToolConfig]
concurrency = 10
jobqueue.capacity = 2000
jobqueue.reload.threshold = 0.4
buffer.size = 100000
db.commit.interval = 3
```

fileid.ignore.if使工具在提交任务时忽略file id符合指定条件的任务。

fileid.ignore.unless使工具在提交任务时忽略file id不符合指定条件的任务。

error.ignore.if使工具在重试出错任务时忽略错误日志符合指定条件的任务。

以上三个选项若启用则应提供一个正则表达式，否则留空，语法参考Python re Module Reference。如果同时提供了fileid.ignore.if和fileid.ignore.unless，并且某个file id同时符合两个正则表达式，那么它会被忽略。

如忽略OS X操作系统下的.DS_Store文件，不重试错误码为-1886的任务：

```
[Advanced]
fileid.ignore.if = .*.DS_Store
fileid.ignore.unless =
error.ignore.if = .*code: -1886.*
```

控制台使用说明

1 新手指引

开发者可以按照
管理控制台指定的步骤熟悉相关
的SDK接口和代码，并验证相应的服务，完成接入指引，具体可参见接入指引。

2 图片空间

空间（Bucket）对应于旧版的应用，是图片资源的组织管理单元。之前，开发者为了将项目中不同类型的图片分类存储需要为每类图片建立一个应用，而开发者需要为每个应用部署一套鉴权服务来生成签名来验证请求的合法性，这样开发者的一个项目如果在管理控制台建立了多个应用，开发者就需要部署多套鉴权服务，而不能一个项目的多个应用共用一套鉴权服务。为了解决这个问题，新版的图片服务使用了空间的概念，开发者可以为一个项目创建多个空间，用以存储不同类型的图片，同时一个项目下的多个空间共用一套鉴权服务，从而极大简化了开发者的鉴权任务。

> 注意： 为了兼容旧版本（V1版本），万象优图将开发者之前创建的应用添加到了同一个默认的项目中，用户可以在该默认项目下创建新的空间。新创建的空间共用一套鉴权服务，但是开发者在旧版本控制台创建的应用和新创建的空间的鉴权不通用。
> 如下面的图片空间图所示，在图片空间下开发者可以添加空间，管理空间，为空间设置样式。开发者在V1版本控制台上面创建的应用转化为一个空间，使用其创建时的AppID作为其空间名称。

不同版本空间区分：

 ：V2版本图标

：V1版本图标

V2加强版本：V2加强版本空间名称是一串字符串，无特殊标记，如下图中的“newbucket”；

V2版本：V2版本空间名称前面会加一个V2版本图标予以标示，如下图中的“yuntest”；

V1版本：V1版本空间名称前面会加一个V1版本图标予以标示，如下图中的“200636”。

注：版本的详细说明参见[版本说明]。

2.1 添加空间

点击“添加按钮”，来添加新的空间。如下面添加空间图所示，创建一个名称为“tencentyun”的空间，点击“完成”按钮完成空间的添加。

创建完成后，图片空间下就会多出刚才创建的“tencentyun”空间，如下图所示。

2.2 空间管理

在图片空间下，开发者可以对添加的每个空间进行管理，这里选择刚才创建的“tencentyun”空间，点击该空间操作下的“管理”操作，可以进入图片空间的空间管理页面。在空间管理页面，开发者可以查看空间的基本信息，进行域名设置，404页面设置，防盗链设置，回调设置等。

2.2.1 域名设置

为了满足业务自身域名的统一性，腾讯云•万象优图支持自定义域名设置。

将鼠标移动至“未设置”文字上，旁边会出现铅笔形状按钮，点击铅笔形状按钮进行域名设置。如下图，在文本框中输入自定义的域名，点击“保存”按钮进行保存，如下图所示。万象优图会审核开发者自定义的域名，如果审核成功，自定义的域名预计30分钟生效；如果审核失败，开发者需要重新进行域名设置。

2.2.2 镜像源设置

镜像源即源站，就是开发者存储业务图片的地方。

镜像源设置是针对开发者希望使用万象优图进行图片下载的回源镜像功能。设置镜像源后，开发者可以使用新域名通过万象优图加速图片下载，同时万象优图会根据开发者设置的镜像源类型将相应的图片转存到万象优图

。设置生效时间平均为30分钟。

注：开发者可以参考场景接入进行相应的设置。

2.2.2.1 镜像源映射

镜像源映射是设置镜像源域名和万象优图域名或者开发者自定义域名对应关系的功能。

如果开发者没有设置自定义域名，开发者只需将图片空间默认的域名映射到源站域名，假定开发者源站域名为source.example.com，设置如下图所示：

如果开发者设置了自定义域名（参见域名设置），镜像源映射会出现两行，如下图所示：

万象优图根据域名进行回源，如下图所示，如果开发者只设置了，tencent.liudon.org到source.example.com的域名映射，则用户访问域名tencent.liudon.org才会回源到源站，而访问域名tencentyun-10000812.image.myqcloud.com则不会回源到源站。

开发者也可以将自定义域名和图片空间默认域名映射为同一个域名，同样也可以映射为不同的域名。不同的设置，根据域名回源的原则，会产生不同的效果。如下图所示：

2.2.2.2 镜像源类型

万象优图现只支持存储原图的镜像源类型

当用户请求的为原图（是否带有签名都可以）时，万象优图会将图片存储下来，之后针对该图片，可以使用空间样式，样式下载别名和实时处理功能；

当用户请求的为带有样式别名或压缩参数的图片时，万象优图会回源到源站，将图片返还给用户，并在CDN上面做缓存；同时会启动另外一个任务，去源站拉取原图，并将图片进行储存。

> 注意：万象优图只识别实时处理
> 文档中的压缩参数，以及在万象优图控制台图片空间设置的样式别名，对于不理解压缩参数，万象优图按原图处理。

例如：

用户请求http://domain.com/url?imageView2/1/w/100/h/100时，
万象优图会回源两次，将缩略图http://domain.com/url?imageView2/1/w/100/h/100缓存到CDN上面，
并将原图http://domain.com/url存储到万象优图；
用户请求http://domain.com/url，万象优图识别为原图，回源拉取图片，并将图片存储到万象优图；
用户请求http://domain.com/url?a=1，万象优图不识别压缩参数a=1，会将其当做一张原图来处理，存储时会舍弃"?a=1"参数（http标准实现），url作为fileid(图片标识)存储到万象优图；
用户请求http://domain.com/url!a=1，万象优图不识别压缩参数a=1，会将其当做一张原图来处理，但存储

时不会舍弃"!a=1"参数，url!a=1作为fileid(图片标识)存储到万象优图；

注：开发者可以参考场景接入进行相应的设置。

2.2.3 防盗链设置

为了防止不良网站盗用开发者的图片链接，盗取图片流量，给开发者带来经济损失，腾讯云•万象优图支持防盗链功能，通过HTTP协议支持的Referer机制（参见HTTP Referer）来进行相应的来源识别和管理。

2.2.3.1 Token防盗链

TOKEN防盗链相当于私密下载，启用token防盗链之后，需要稍等10-30分钟方可生效，下载图片时需要在url中传入签名，签名参数名为sign，签名可以是多次有效，也可以是单次有效，具体签名方法见签名方法。

如需使用请到图片空间--空间管理中配置，如下图所示，打开Token防盗链开关，点击保存即可。

2.2.3.2 域名防盗链

您可以通过设置域名防盗链白名单，来决定哪些域名能够访问该空间下的图片；也可以通过设置域名防盗链黑名单，来决定哪些域名不能够访问该空间下的图片。

如需使用请到图片空间--空间管理中配置，如下图，设置完成后，点击“保存”进行保存。

2.2.4 样式分隔符

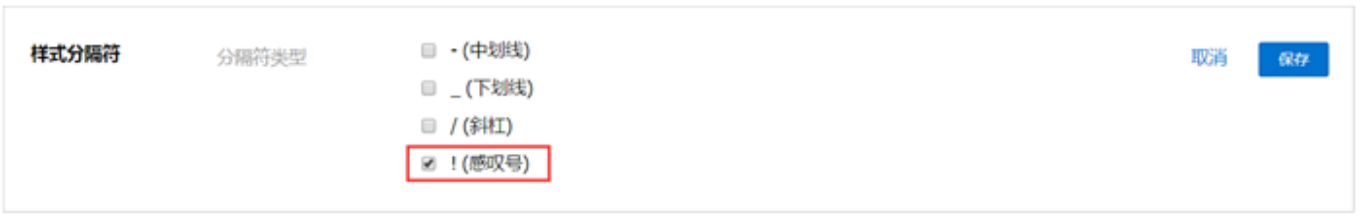
样式分隔符是分割文件名称和处理样式的符号，包含中划线(-)、下划线(_)、斜杠(/)和感叹号(!)。设置生效时间平均为30分钟。

> 注意：处理样式包括控制台设置的样式和样式下载别名。控制台设置的样式是图片上传后会针对样式生成一个样式图；而样式下载别名是图片下载时为实时处理参数组起的一个别名。

使用方法：http: //绑定域名/文件名称 + 分隔符 + 处理样式名。

如下图所示，假定设定感叹号(!)为样式分隔符，同时样式名为yunstyle，原图fileid为sample.jpg，则原图的经过样式yunstyle处理后的图片url为：

http://space.image.com/sample.jpg!yunstyle



> 注意：

1、为避免歧义，处理样式名中不可出现当前所启用的间隔标志符
2、更改分隔符需清除缓存，全网生效至少需要24小时
3、取消已使用的分隔符，可能导致产品功能异常

2.2.5 样式下载别名

样式下载别名是使用图片实时处理功能时，为开发者预设的特定的处理参数组合起的一个别名。设置生效时间平均为30分钟。
开发者可以参考实时处理文档设计好处理参数，按照（样式下载别名=处理字符串）的格式进行设置，每行一个，回车分隔，如下图所示：

2.2.6 回调设置

2.2.6.1 回调URL

为了减少图片上传的处理环节，腾讯云•万象优图支持图片上传回调URL配置，能够满足客户端上传完图片由图片服务直接反馈给业务服务端，节省客户端的中转逻辑。

将鼠标移动至回到URL“未设置”文字旁边，出现铅笔形状按钮，点击进行回调URL设置。如下图所示，输入回调URL，点击“保存”。回调URL预计30分钟生效。

回调URL设置生效后，图片上传完毕后腾讯云•万象优图会默认回调该URL，向其发送一个标准的HTTP POST通知消息。

HTTP包信息如下表：

参数名称	类型	必选	描述
appid	String	是	接入万象优图时,生成的唯一id, 用于唯一标识接入业务
token	String	是	签名
userid	String	否	签名时填入的账号信息
url	String	是	上传后的资源url，包括域名
fileid	String	是	资源存储的唯一标识
sha	String	是	资源sha1
size	Int	是	资源大小，字节
magic_context	String	否	上传图片，SDK携带的透传参数。如果业务有很复杂的参数，建议整理成json格式。

接口样例如下：

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded

appid=1234
&b
ucke
t=bucket
Name&fileid=e9e5
70bfb54ebc708179208983ce8b95&mag
ic_context=&sha=77
cc9a9cb437a8d82c7545c3dc158a1dd09796
```

d9&size=51200&token=7

d33a816d302b6&url=http://xx.yy.com/10000/zz/e9e570bfb54ebc708179208983ce8b95/0

2.2.6.2 审核回调URL

为保障图片上传的合法性，腾讯云•万象优图将多年积累的非法图片识别的技术开放出来，能够支持图片上传的自动识别，如果发现该图片为非法图片默认会将结果通知到服务端。

如下图所示，将鼠标移动至回到URL“未设置”文字旁边，出现铅笔形状按钮，点击该铅笔形状按钮，进行回调URL设置。如下图所示，输入回调URL，点击保存。回调URL预计30分钟生效。

当发现上传的图片为非法图片时万象优图会默认回调该URL，向其发送一个标准的HTTP POST通知消息。HTTP包信息如下表：

参数名称	类型	必选	描述
url	String	是	上传后的资源url，包括域名
fileid	String	是	资源存储的唯一标识
result	Int	是	审核结果,一般是1删除
type	Int	是	100//正常;10001//广告;20001//政治;20004//社会;20002//色情;20006//违法犯罪;20008//欺诈;20013//版权;21000//其他

接口样例如下：

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded

appid=1234
&fileid=e9e570bfb54ebc708179208983ce8b95&result=1&type=20002&url=http://xx.yy.com/10000
/zz/e9e570bfb54ebc708179208983ce8b95/0&userid=123456
```

2.3 空间样式

在图片空间下，开发者可以为添加的每个空间设置图片的样式，方便管理不同需求的图片。样式创建成功后，预计30分钟会生效，之后上传的每张图片都会针对每个样式生成一个处理后的图片。

注意：

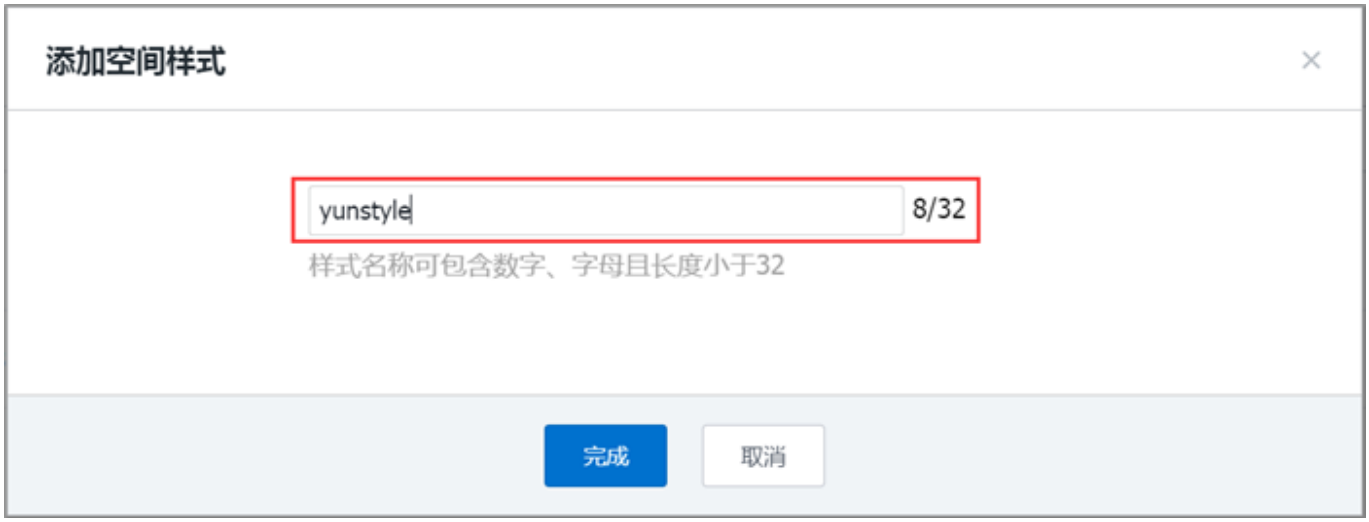
(1) 样式名需要区分大小写，需要完全按照样式名来获取带有样式的图片，设置样式，每新上传一张图片，除了其原图占用存储空间之外，生成处理后图片也会占用开发者存储空间。

(2) 目前，开发者自定义的样式，只对样式生效后新上传的图片有效，对样式生效之前上传的图片无效。建议开发者先设置所需样式，再上传图片。

这里选择“tencentyun”空间，点击该空间操作下的“样式”操作，进入图片空间的空间样式页面，如下图所示。

在空间样式页面下，开发者可以管理空间的图片样式，如添加样式，设置样式，删除样式等。

点击“添加”按钮来创建样式，如下图所示，输入样式名称，这里创建名称为“yunstyle”的样式，点击“完成”。



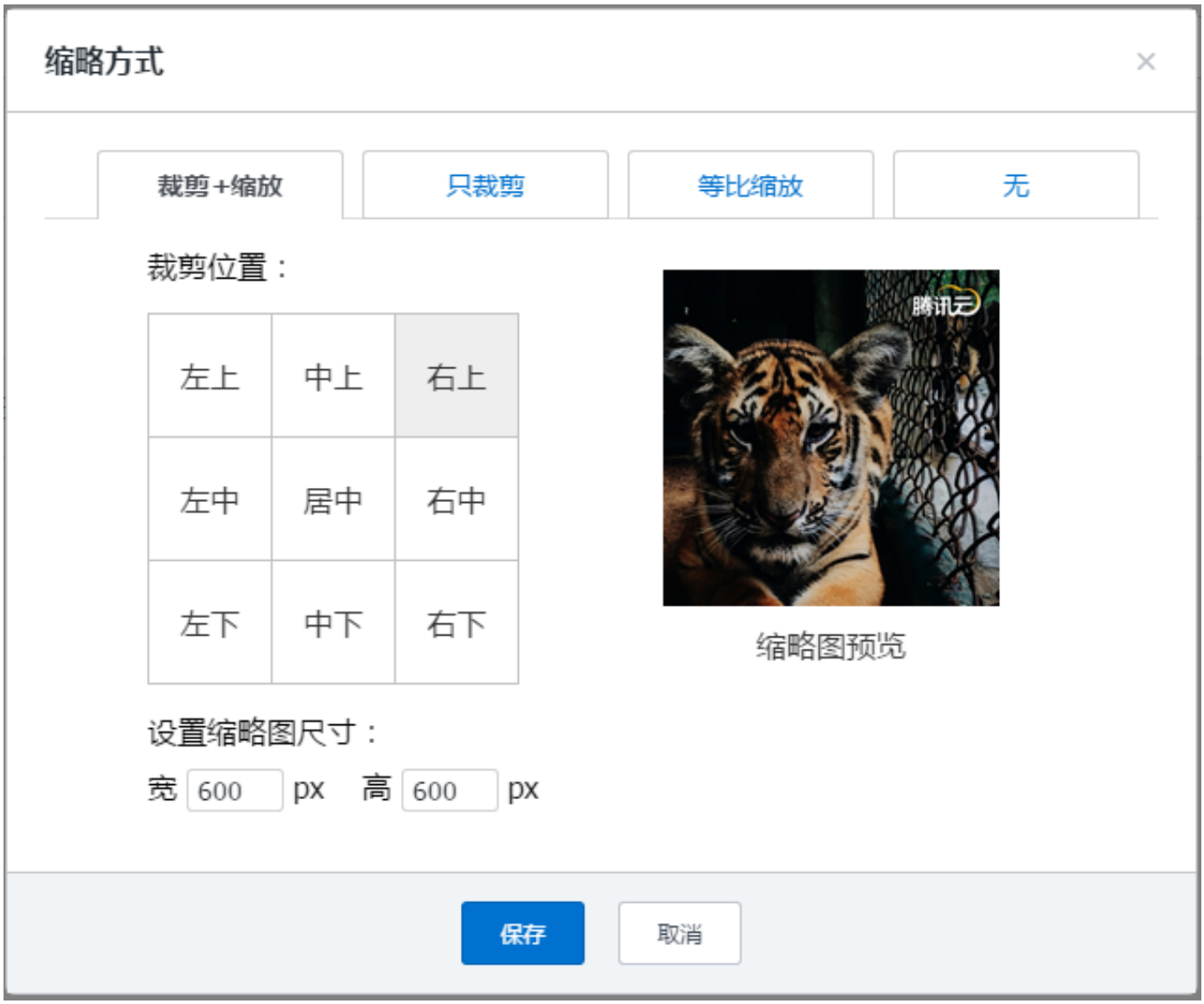
样式创建成功后，可以为样式设置缩放方式，文字和图片水印，锐化效果和输出效果等。

2.3.1 缩放方式

万象优图支持裁剪+缩放、只裁剪、等比缩放三种缩略处理方式。默认为不处理，即不对图片进行缩放，如需缩放，可以编辑缩略方式。
注意： 万象优图的缩放处理都是将图片缩小，而不会将图片拉伸，若上传图片的尺寸小于设置的尺寸，则不会对图片进行处理。

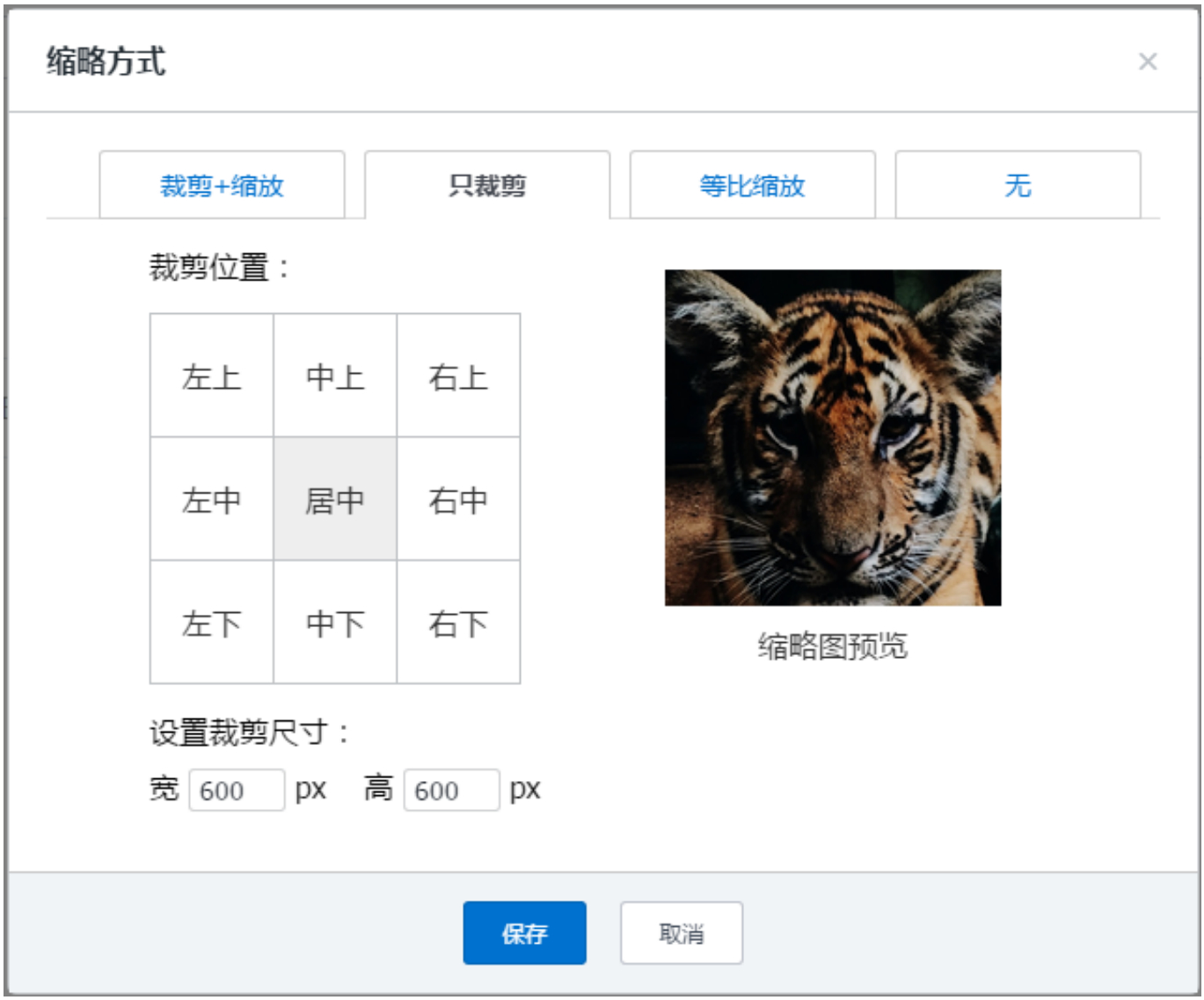
2.3.1.1 裁剪+缩放

裁剪+缩放功能是为了能够在原图较大，目标图较小且比例和原图不一致时使用。例如原图为1200*900，缩略图为尺寸为600*600，则会先将原图按宽高比（600:600即1:1）做裁剪，裁剪原图到900*900后，再进行缩放，缩小到600*600的目标尺寸。可以通过九宫格的选择，来决定裁剪中心的位置。



2.3.1.2 只裁剪

只裁剪样式根据您设置的裁剪位置和缩略图尺寸，直接对原图进行裁剪，其中九宫格确定裁剪的中心位置。例如设置裁剪位置为居中，裁剪后的缩略图尺寸为600*600
，则沿宽中线左右各取300，高沿中线各取300，形成最终的600*600缩略图进行裁剪。如下图所示。



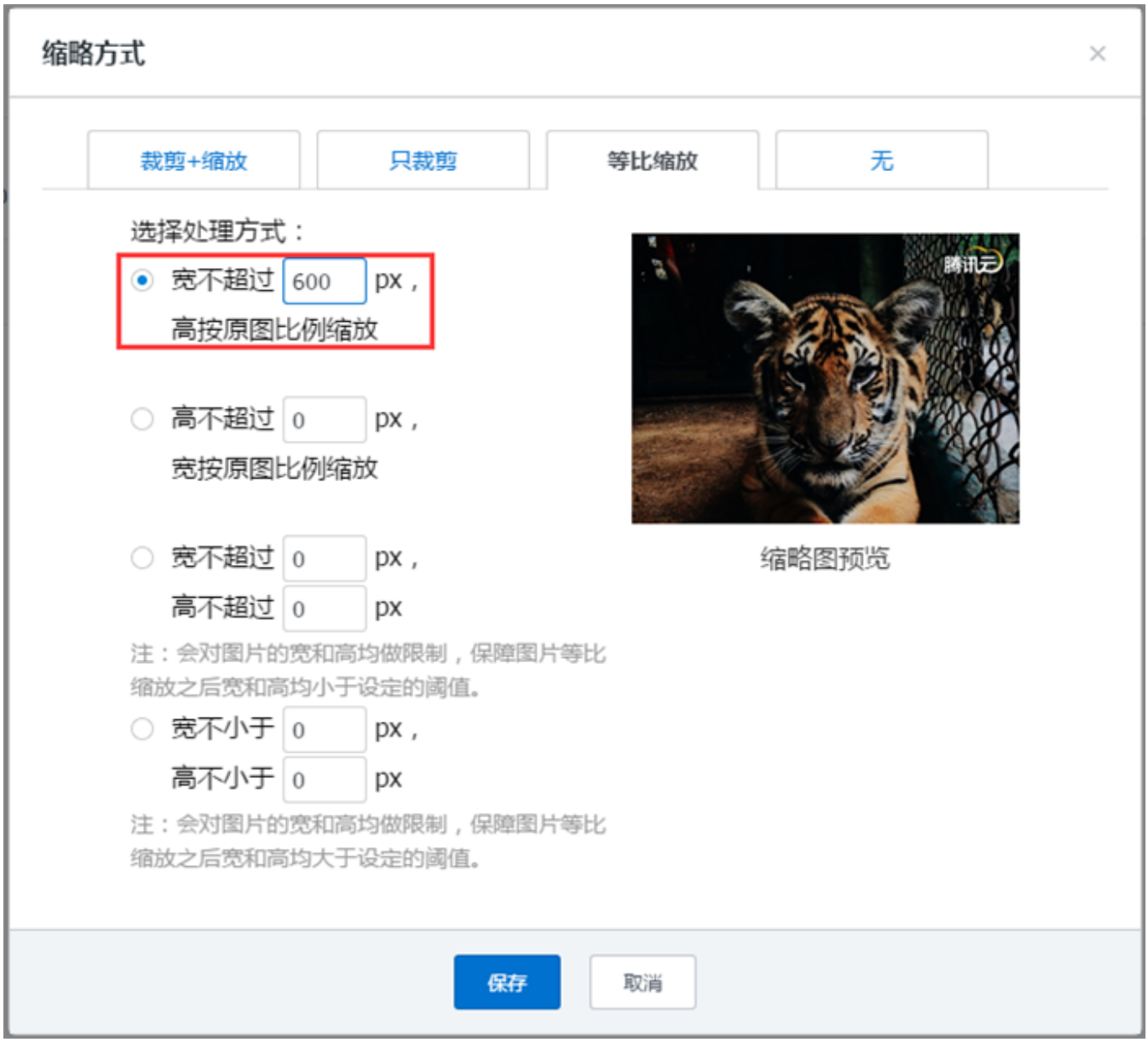
2.3.1.2 等比缩放

等比缩放是将上传的图片按照原图的宽高比例进行缩放，缩放到设定的尺寸。

目前支持三种尺寸设定：

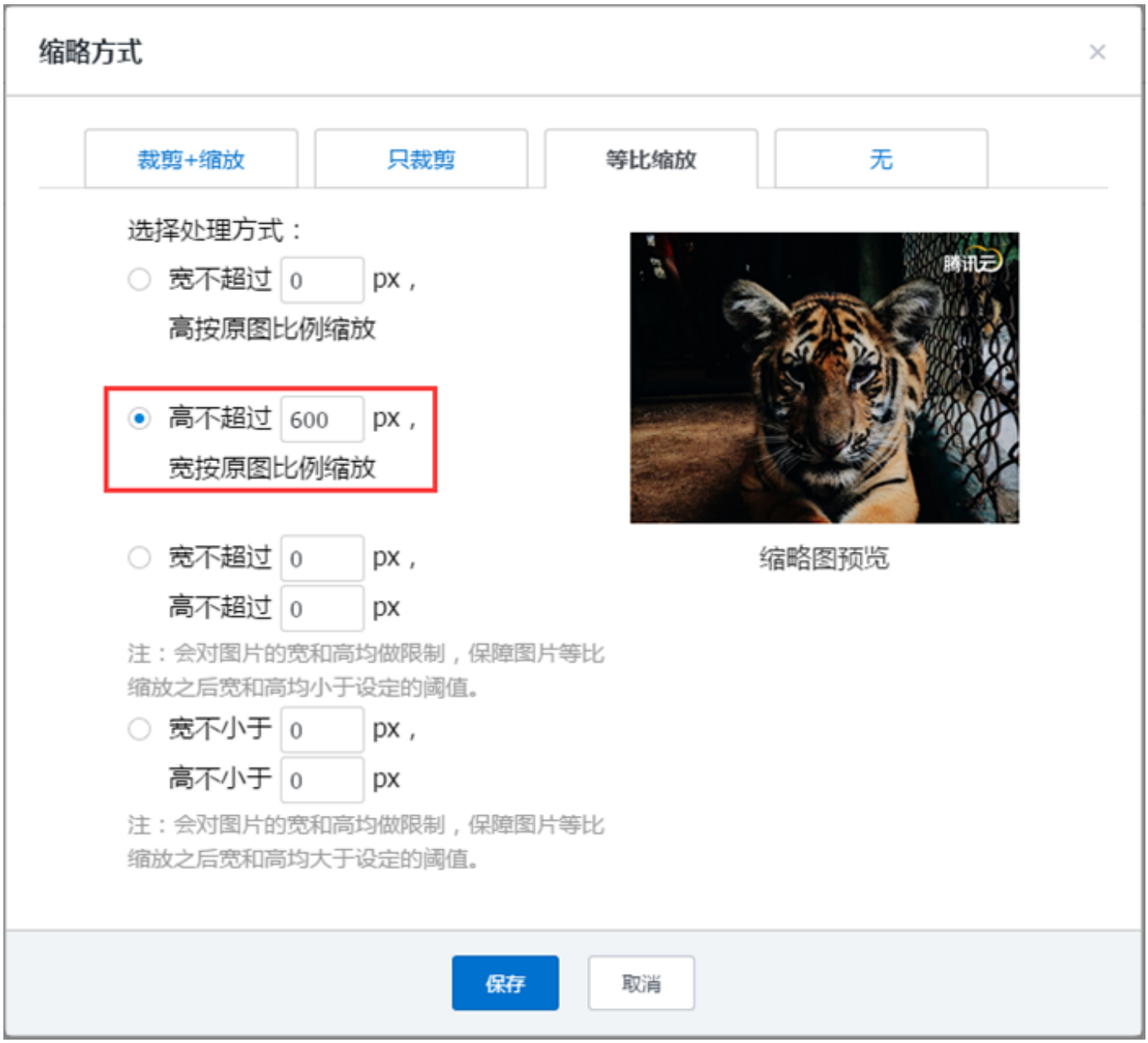
1. 限定宽，高自适应

如下图，限定宽为600，高按原图比例缩放。这样如果上传的图片宽*高为 1200*900则处理后的图片为600:450；但是如果上传的图片宽<600,则不会进行处理，比如上传的图片宽*高为50*50，则保存的缩略图还是50*50。



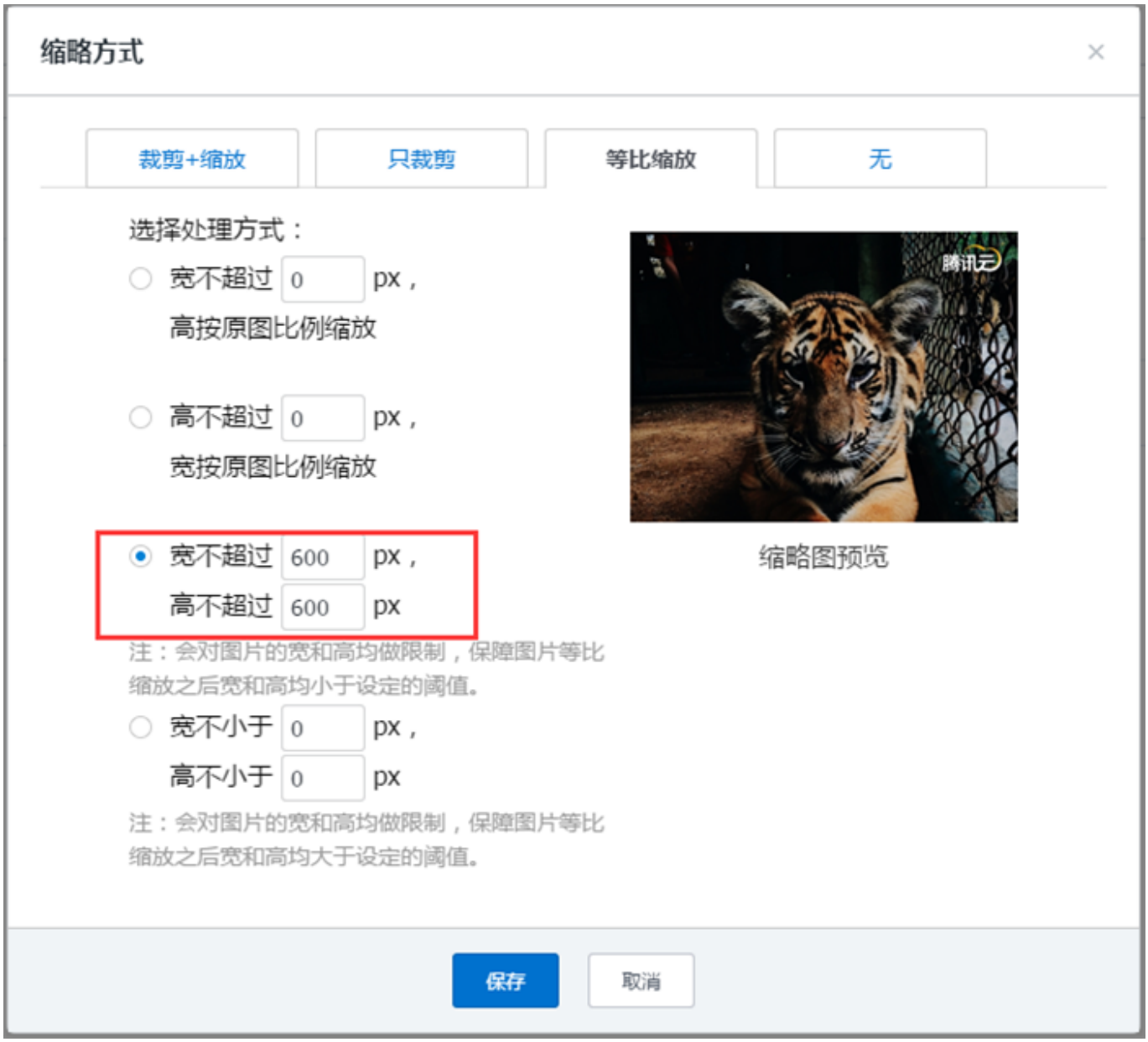
1. 限定高，宽自适应

如下图，限定高为600，高按原图比例缩放。这样如果上传的图片宽*高为 1200*900，则处理后的图片为800*600；但是如果上传的图片高<600,则不会进行处理，比如上传的图片宽*高为50*50，则保存的缩略图还是50*50。



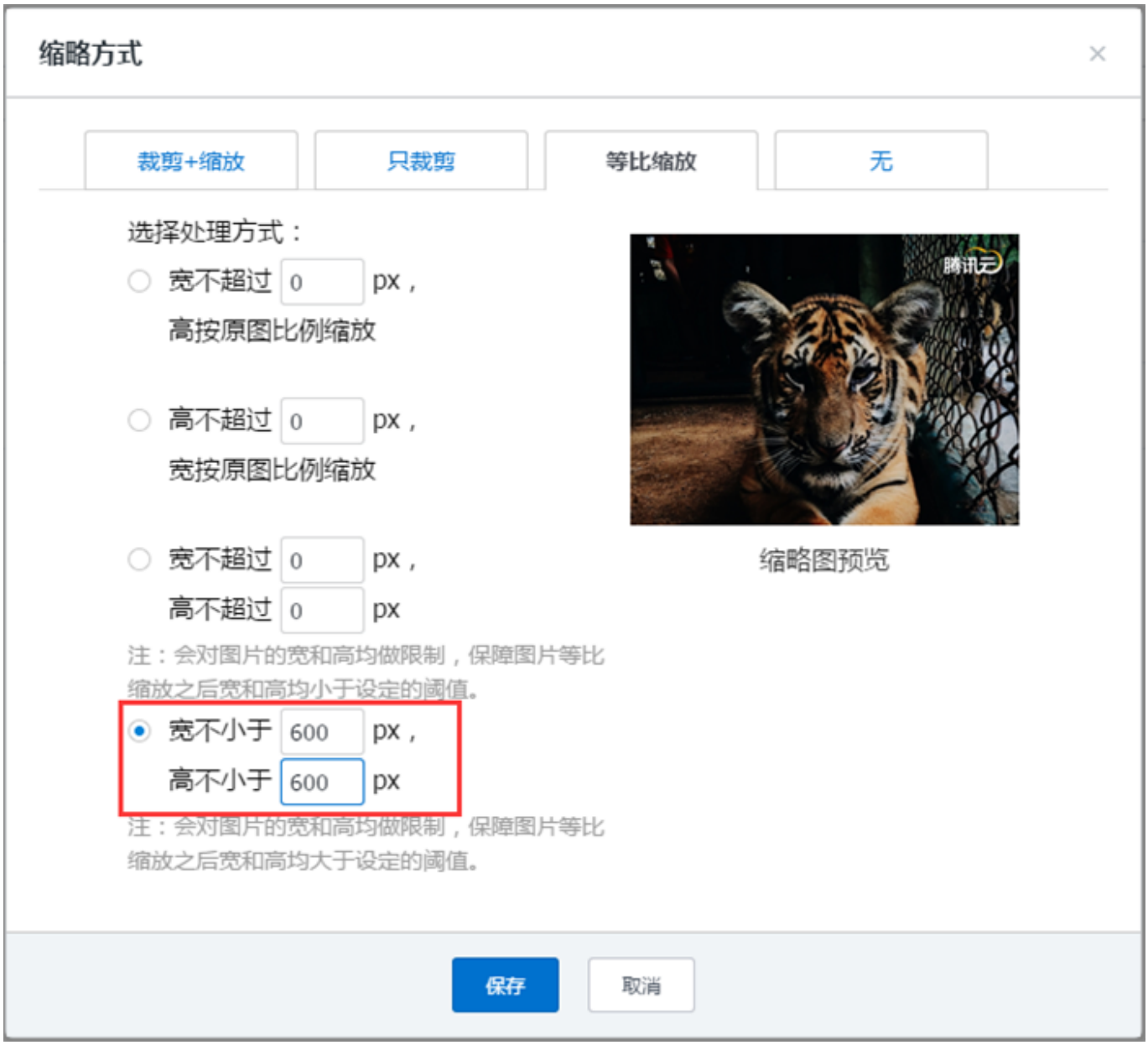
1. 限定宽高的最大值

如下图，限定宽为600，高为600。这样如果上传的图片宽*高为 1200*900，则处理后的图片为600*450；如果上传的图片宽和高都<600,则不会进行处理，比如上传的图片宽*高为50*50，则保存的缩略图还是50*50。



1. 限定宽高的最小值

如下图，限定宽为600，高为600。这样如果上传的图片宽*高为 1200*900，则处理后的图片为800*600；如果上传的图片宽和高都<600,则不会进行处理，比如上传的图片宽*高为50*50，则保存的缩略图还是50*50。



注：所有处理方式都不会对图片进行拉伸。

2.3.2 水印处理设置

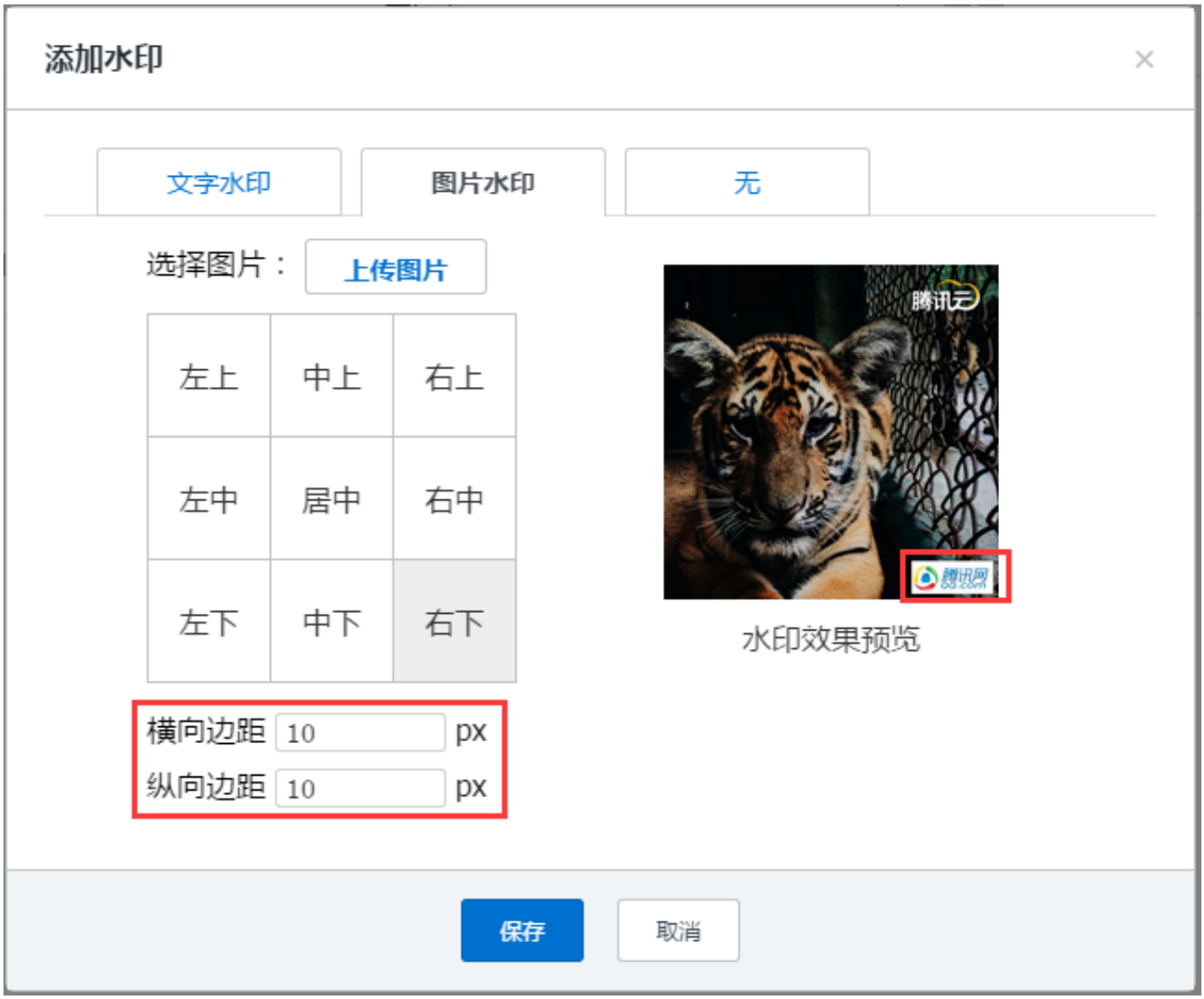
2.3.2.1 文字水印

文字水印能够按照您设置的文字，由九宫格确定水印位置，在目标图片上设置水印。如选择右下，横向边距和纵向边距为10，文字水印为“万象优图”，得到的缩略图如下。


添加水印
文字水印
图片水印
无
万象优图
左上
中上
右上
左中
居中
右中
左下
中下
右下
横向边距 10 px
纵向边距 10 px
水印效果预览
保存
取消

2.3.2.2 图片水印

图片水印能够按照您设置的图片，由九宫格确定水印位置，在目标图片上设置水印。如选择右下，横向边距和纵向边距为10，上传图片水印，得到的缩略图如下



2.3.3 输出效果

开发者可设置样式图的输出格式和输出质量，如下图所示。

2.4 图片识别

打开需要开通鉴黄服务的图片空间，进入空间识别页面，开启图片鉴黄

2.4.1 回调阈值设置

1. 回调阈值为分值区间
2. 设置后，万象优图将会把图片分值在阈值内的图片返回给您
3. 对于腾讯云封禁的图片，如果勾选了回调，此类图片也会返回给您，但是无法进行下载
4. 回调URL需以http开头且默认返回200正确码方可以进行使用，请在保存设置前进行检查

如下图所示，输入回调URL，点击“保存”。回调URL预计30分钟后生效。

回调URL设置生效后，当发现上传的图片分值在阈值内，万象优图会默认回调该URL，向其发送一个标准的HTTP POST通知消息。

HTTP包信息如下表：

参数名称	类型	必选	描述

参数名称	类型	必选	描述
url	String	是	上传后的资源url，包括域名
result	Int	是	供参考的识别结果，0正常，1黄图，2疑似图片
forbid_status	Int	是	封禁状态，0表示正常，1表示图片已被封禁（只有存储在万象优图的图片才会被封禁）
confidence	Double	是	识别为黄图的置信度，范围0-100；是normal, hot, porn的综合评分
normal	Double	是	图片为正常图片的评分
hot	Double	是	图片为性感图片的评分
porn	Double	是	图片为色情图片的评分

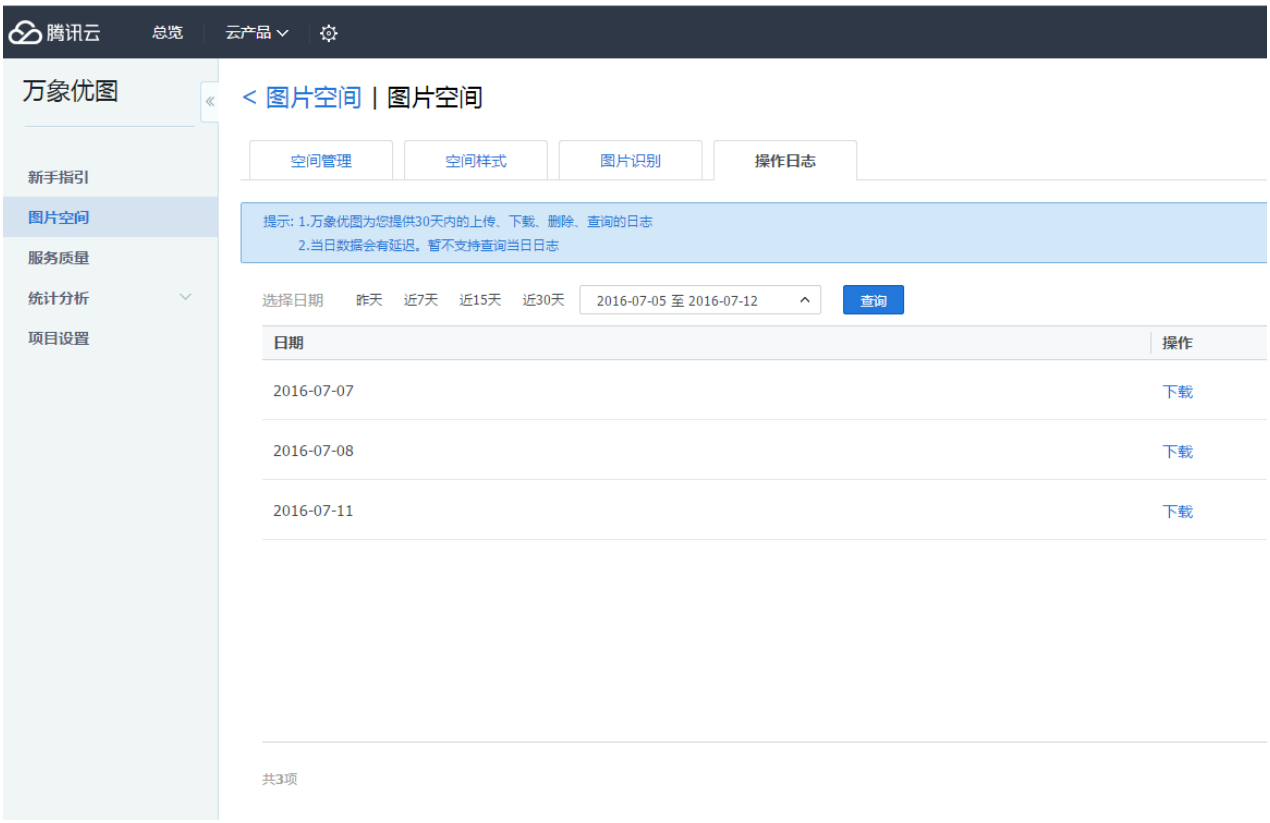
接口样例如下：

```
POST / HTTP/1.1
Content-Type: application/x-www-form-urlencoded

confidence=100&forbid_status=1&hot=3.971e-06&normal=1.988e-10&porn=100
&result=1&url=http://xx.yy.com/10000/zz/e9e570bfb54ebc708179208983ce8b95/0
```

2.5 操作日志

1. 登录万象优图控制台，进入目标图片空间>操作日志
2. 万象优图提供30天之内图片上传、下载、删除、查询的操作日志。选择您需要查询的时间，在列表里下载即可。由于数据会有延迟，故暂不支持当日操作日志的下载



3 服务质量

腾讯云•万象优图针对单个应用，汇总其全国各省、各运营商、不同网络类型移动终端用户上传下载图片的速度、成功率、平均文件大小等信息，通过海量数据计算，给出全方位服务质量数据。数据会有一定延迟，延迟在2个小时左右。

如需使用请到万象优图--服务质量中查看。

4 使用统计

腾讯云•万象优图提供针对单个应用其存储量、下载流量、上传下载删除的次数，wep及水印请求的次数等信息的使用量统计数据，数据会有一定延迟，延迟时间在20分钟以内。

可以在万象优图--使用统计中查看使用统计数据。

5 项目设置

在项目设置下，开发者可以查看项目的密钥，添加密钥，启动、禁用和删除密钥。一个项目最多有两套密钥，但是不可以同时禁用。

添加密钥：

禁用密钥：


项目设置
默认项目
+ 添加密钥
项目名称
密钥管理
创建时间
状态
操作
默认项目
Secret ID:
Secret Key:
2015-07-03 16:08:07
已停用
启用
删除
默认项目
Secret ID:
Secret Key:
2015-07-03 18:36:25
使用中
禁用
1、为了增强整体图片服务的延展能力，扩大图片服务支持的的url范围，万象优图服务就整体url及鉴权方式做了次整体的升级。
2、为了不影响老客户正常的体验，之前老用户创建的appid均会保留，且appid的规则会延续，一个appid对应一组维纳斯、图片服务、微视频服务。
3、老客户图片数据均按照空间的方式保留到图片空间中，样式功能、自定义域名、回调配置、404配置、防盗链配置均会在空间中保留。
4、如需要快速兼容新版本的自定义url建议直接重新建立空间来做调整，如需要做数据迁移，我们会直接为您提供迁移服务。

部署示例

Android部署示例

1 环境准备

本文档以eclipse+Android插件为例，开发者需准备好相应的开发环境。

2 SDK集成

1. 下载Android SDK

Android SDK的下载地址为：Android SDK。

1. 导入项目

将SDK包中的libs目录合并到本地工程的libs目录，然后配置工程导入所有jar包。

上传SDK的libs目录如下：

下载SDK的libs目录如下：

1. 配置manifest

SDK需要网络访问相关的一些权限，需要在manifest中进行权限声明如下所示:

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

3 测试

以下程序全部来自于Demo，这里做部分说明。Demo的下载地址为：Android Demo（Android SDK中的Demo和此Demo相同）。

> 注意：如果开发者想根据demo做简单测试，请将demo中的项目信息修改为自己的项目信息。具体修改方法如下：
> (1) 进入package com.tencent.cloud.demo.common中的class BizService；
> (2) 修改BizService中的APPID(项目ID)，BUCKET（空间名称）为自己的项目信息；
> (3) 确保开发者自己的测试鉴权服务器中的项目信息和(2)中的项目信息一致。

程序在使用万象SDK服务之前必须注册，进行用户信息初始化，注册所需要的签名因安全因素需要放在开发者服务器上面，由
终端在需要使用的时候向开发者服务器请求，开发者服务器鉴权服务部署参考鉴权服务部署示例。

3.1 向业务服务器请求sign

在体验的程序中的请求sign代码在UpdateSignTask 中，代码如下：

```
HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();
InputStream in = urlConnection.getInputStream();
```

```
byte[] mSocketBuf = new byte[4* 1024];
ByteArrayOutputStream baos = new ByteArrayOutputStream();
int count = 0;
while ((count = in.read(mSocketBuf, 0, mSocketBuf.length)) > 0) {
   baos.write(mSocketBuf, 0, count);
}

String config = new String(baos.toByteArray());
JSONObject jsonData = new JSONObject(config);
String configContent = jsonData.getString("sign");
```

3.2 注册

代码在BizService中。

```
UploadManager.authorize(APPID, USERID, FileType.Photo, PHOTO_SIGN);
```

3.3 使用万象服务，下面以上传为例说明

```
UploadTask task = new PhotoUploadTask(filePath,new IUploadTaskListener() {
    @Override
    public void onUploadSucceed(final FileInfo result) {
        Log.i("Demo", "upload succeed: " + result.url);
    }

    @Override
    public void onUploadStateChange(TaskState state) {
    }

    @Override
    public void onUploadProgress(long totalSize, long sendSize){
```

```
            long p = (long) ((sendSize * 100) / (totalSize * 1.0f));
            Log.i("Demo", "上传进度: " + p + "%");
        }

        @Override
        public void onUploadFailed(final int errorCode, final String errorMsg){
            Log.i("Demo", "上传结果:失败! ret:" + errorCode + " msg:"
+ errorMsg);
        }
    });
    task.setBucket(BizService.BUCKET);  // 设置Bucket(可选)
    task.setFileId("test_fileId_" + UUID.randomUUID()); // 为图片自定义FileID(可选)
    cuploadManager.upload(task);  // 开始上传
```

其他功能同理，具体代码详情见demo。

iOS部署示例

1 环境准备

这里以Xcode为例，开发者需安装了Xcode开发环境。

2 SDK集成

iOS SDK下载地址为：iOS SDK。

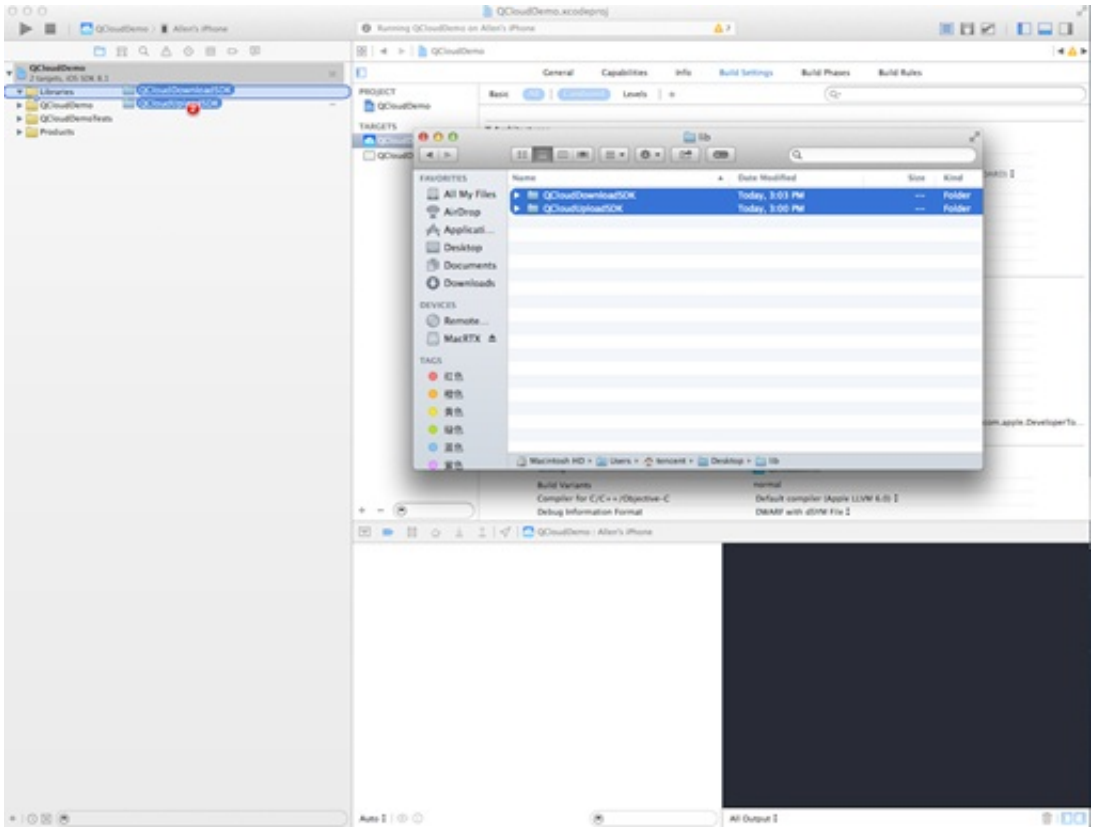
万象优图iOS SDK其中包括上传SDK和下载SDK，上传SDK压缩包QCloudUploadSDK.zip,下载SDK压缩包QCloudDownloadSDK.zip.上传和下载SDK压缩包中分别包含了一个.a静态库和一个包含头文件的文件夹Headers，解压后的内容如下：

上传SDK:

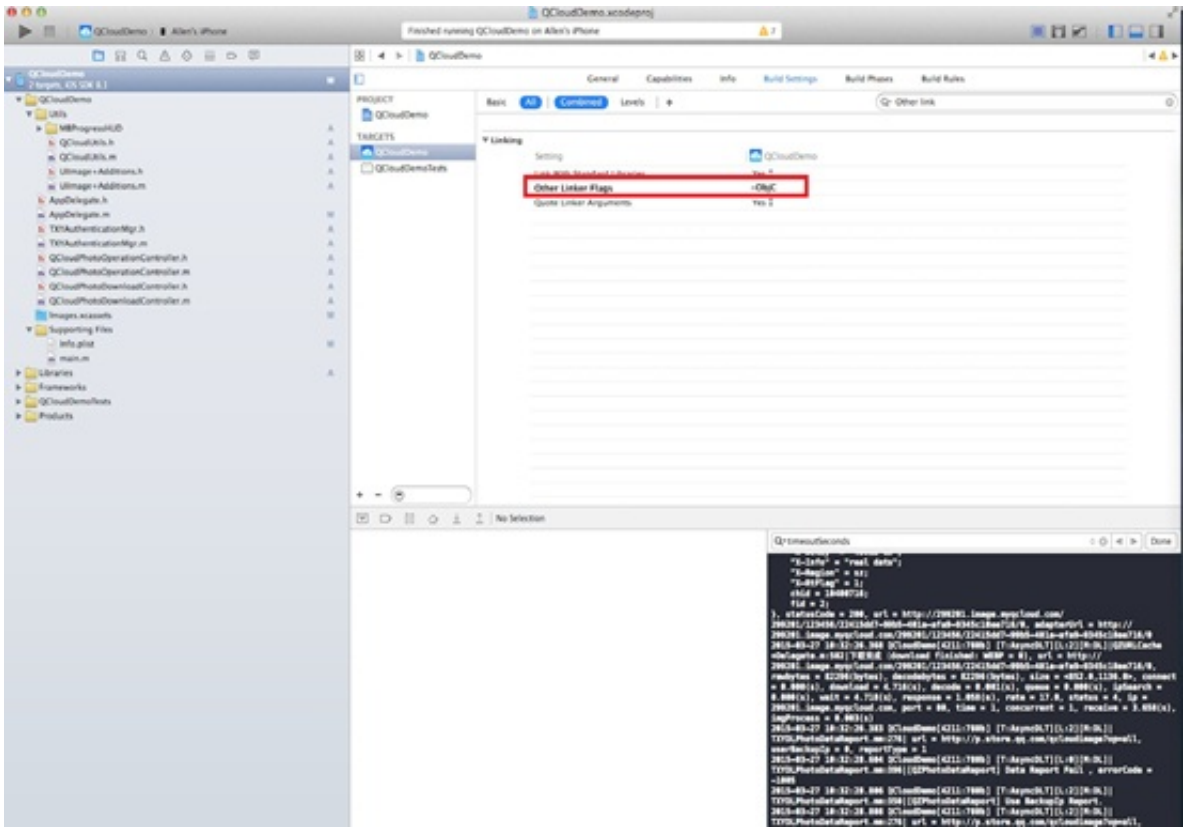
下载SDK:

将解压后的QCloudUPloadSDK和QCloudDownloadSDK拖入工程目录，Xcode会自将其加入链接库列表中。

注：如果只需要上传或下载功能，则只拖入对应的SDK即可。



在build Settings中设置Other Linker Flags，加入参数-ObjC



在build Phases -> Link Binary With Libraries中加入以下几个依赖库

1) SystemConfiguration.framework

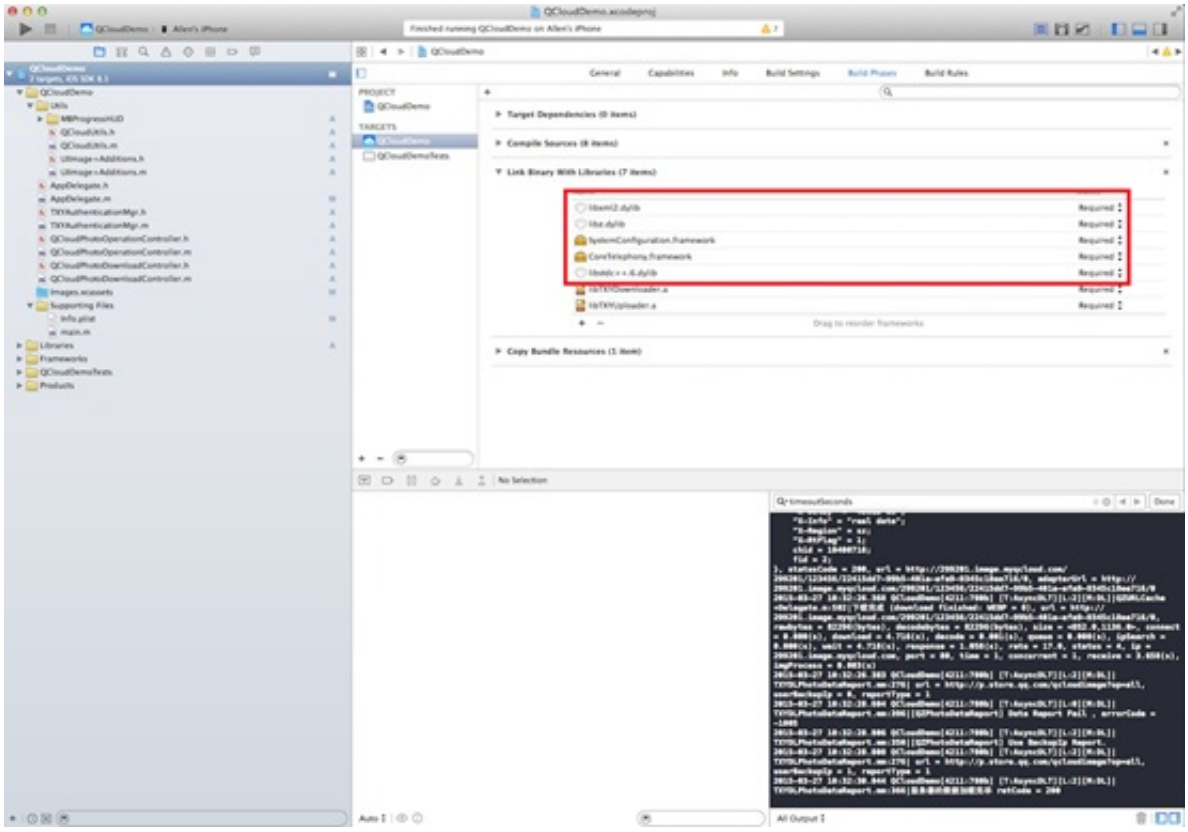
2) CoreTelephony.framework

3) MobileCoreServices.framework

4) libxml2.dylib

5) libz.dylib

6) libstdc++.6.dylib



注：如果只需要上传或下载功能，则只需要引入对应的动态库：

上传SDK依赖的系统动态库有：

1) SystemConfiguration.framework

2) CoreTelephony.framework

3) libstdc++.6.dylib

4) libz.dylib

下载SDK依赖的系统动态库有：

1) SystemConfiguration.framework

2) CoreTelephony.framework

3) MobileCoreServices.framework

4) libxml2.dylib

5) libz.dylib

3 测试

以下所有代码都来源于体验Demo，这里讲述iOS客户端SDK的基本使用方法。Demo的下载地址为：iOS Demo。

> 注意：如果开发者想根据demo做简单测试，请将demo中的项目信息修改为自己的项目信息。具体修改方法如下：
> (1) 打开文件QCloudAuthenticationMgr.m文件；
> (2) 修改init 函数中的appid （项目id） 和bucket（空间名称）字符串为自己的项目信息；
> (3) 确保开发者自己的测试鉴权服务器中的项目信息和(2)中的项目信息一致。

3.1 APP注册

在使用其他接口之前必须先调用此方法进行用户信息初始化，即APP注册。

```
[[QCloudAuthenticationMgr shareInstance] registerAuthentication];
```

下面给出函数的一种实现方法，分为两步：

1 从开发者业务服务器获取签名，开发者服务器鉴权服务部署参考鉴权服务部署示例

2 上传和下载注册(上传和下载需分别注册)

开发者需要根据业务情况修改相应的代码。

```
- - (void)registerAuthentication
{
    NSString *requestSign = nil;
    //此URL是我们demo的服务器的地址这个地方需要替换成使用者的服务器地址
    requestSign = [self signatureWithURL:
@"http://203.195.194.28/php/getsign.php"];
    //请求失败，使用备份签名注册
    self.signature = requestSign;

    //上传需要注册appId，应用级sign，userId根据用户需要选填
    [TXYUploadManager authorize:self.appId userId:self.userId sign:self
.signature];
    //下载需要注册appId，sign信息由用户填入url参数中
    [TXYDownloader authorize:self.appId userId:self.userId];
}


#pragma mark -- 向业务后台拉取签名
- (NSString *)signatureWithURL:(NSString *)urlString
{
    NSURL *url = [NSURL URLWithString:urlString];

    NSURLRequest *request = [NSURLRequest requestWithURL:url];

    NSHTTPURLResponse *httpResponse = nil;
    NSError *connectionError = nil;
    NSData *signData = [NSURLConnection
 sendSynchronousRequest:request returningResponse:&httpResponse error:&connectio
nError];

    NSString *signature = nil;
    NSDictionary *responseDic = [NSJSONSerialization
 JSONObjectWithData:signData options:kNilOptions error:nil];
    signature = [responseDic objectForKey:@"sign"] ;
```

```
    return signature;
}
```

3.2 上传图片

上传图片分为一下两步：

1 初始化上传任务

```
    //1.构造TXYPhotoUploadTask上传任务,
    self
.uploadPhotoTask = [[TXYPhotoUploadTask alloc] initWithPath:self.uploadPhotoPath
                                                 expiredDate:0

                                                msgContext:@"上传成功后，传给用户业务后台的信息"
                                                     bucket:QCLOUD_UPLOA
D_PHOTO_BUCKET    //默认上传到"test1"这个bucket

                           fileId:nil];                          //自定义fileId
```

2 用uploadmanager 来执行上传

```
 //2.调用TXYUploadManager的upload接口
 DECLARE_WEAK_SELF;
 [self.uploadManager upload:self.uploadPhotoTask
                       sign:nil
                   complete:^(TXYTaskRsp *resp, NSDictionary *context) {
                       DECLARE_STRONG_SELF;
                       if (!strongSelf) return;

                       strongSelf.uploadPhotoTask = nil;
                       //retCode大于等于0，表示上传成功
```

```
                        if (resp.retCode >= 0) {
                            //得到图片上传成功后的回包信息
                            TXYPhotoUploadTaskRsp *photoResp = (TXYPhotoUploadTas
kRsp *)resp;
                            strongSelf.urlDisplayLabel.text = photoResp.photoURL;
                            strongSelf.uploadStateLabel.text = @"上传成功!";
                            NSLog(
@"上传成功!,code:%d desc:%@", resp.retCode, resp.descMsg);
                        }
                        else
                        {

          strongSelf.uploadStateLa
bel.text = [NSString stringWithFormat:
@"上传失败,code:%d desc:%@", resp.retCode, resp.descMsg];
                            NSLog(
@"上传图片失败,code:%d desc:%@", resp.retCode, resp.descMsg);
                        }


      } progress:^(int64_t totalSize, int64_t sendSize, NSDictionary *context) {
                         DECLARE_STRONG_SELF;
                         if (!strongSelf) return;

                         long progress = (long)(sendSize * 100.0f/totalSize);

    strongSelf.uploadStateLabel.text = [NSString
 stringWithFormat:@"上传进度: %ld%%",progress];

                         NSLog(
@"上传进度,sendSize %lld,totolSize %lld, progress %ld",sendSize,totalSize,progress);
                }stateC
hange:^(TXYUploadTaskState state, NSDictionary *context) {
                    DECLARE_STRONG_SELF;
```

```
                    if (!strongSelf) return;
                    if(state == TTXYUploadTaskStatePause) {
                        [self.pauseButton setTitle:@"□□" forState:
UIControlStateNormal];
                    }
                    else
                    {
                        [self.pauseButton setTitle:@"□□" forState:
UIControlStateNormal];
                    }

                    NSLog(@"Upload photo task state change %zd",state);
                }];
```

3.3 下载图片

下载和其他功能可参考体验demo。

Web端部署示例

本文档介绍web端如何请求服务端，通过js调用腾讯云万象优图REST API进行图片的上传、下载、查询、复制和删除。其中签名需要向开发者服务器请求，开发者服务器鉴权服务部署参考鉴权服务部署示例。

1 web前端部署与代码示例

将web前端部署在与服务端同域下，web前端代码可以参考以下示例，示例页面参见web示例页面。

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
<title>腾讯云万象优图 - 示例程序</title>
<script type="text/javascript" src="../jquery-2.1.4.min.js"></script>
<script type="text/javascript" src="../jquery.form.min.js"></script>
</head>

<div>
    <h2>腾讯云万象优图 - 示例程序</h2>
    <form id="uploadForm">
        <input type="file" name="FileContent"></input>
        <input id="subbtn" type="submit">
    </form>

     <div id="downloadRet" style="display:none">
         <h3>下载链接</h3>
         <span id="downloadUrl"></span><input id="downloadBtn" type=
"button" value="下载"><br/>
         <img id="downloadImg" src=""></img>
         <h3>文件ID</h3>
         <div id="fileid"></div>
         <h3>管理URL</h3>
         <span id="url"></span>
                <input id="queryBtn" type="button" value="查询">
                <input id="deleteBtn" type="button" value="删除">
```

```
                <input id="copyBtn" type="button" value="复制"><br/>
          <span id="imgInfo"></span>
      </div>
</div>


<script type="text/javascript">

$(document).ready(function() {
    initUploadForm();
});

$('input[name=FileContent]').change(function () {
    initUploadForm();
});

$('body').on('click', '#downloadBtn', function(){
    $('#downloadImg').attr('src', $('#downloadUrl').text());
});

$('body').on('click', '#deleteBtn', function(){
    var manageUrl = $('#url').text();
    if (!manageUrl) {
        alert('尚未获取管理url');
        return false;
    }
    manageUrl = manageUrl + '/del';
    // 请将以下获取签名的链接换成您部署好的服务端http url
    // 建议通过业务登陆态检查来增强安全性，避免签名被非法获取
    $.getJSON(
'http://203.195.194.28/p
hp/getsign.php?type=delete&url='+encodeURIComponent(manageUrl),
                function(data) {
        var sign = data.sign,
```

```
            url = manageUrl + '?sign=' + encodeURIComponent(sign);
        $.ajax({
            type: "POST",
            url: url,
            data: {},
            success: function() {
                  alert('删除成功');
            },
            contentType:"multipart/form-data",
            dataType: 'json'
        });
    });
});

$('body').on('click', '#copyBtn', function(){
    var manageUrl = $('#url').text();
    if (!manageUrl) {
        alert('尚未获取管理url');
        return false;
    }
    manageUrl = manageUrl + '/copy';
    // 请将以下获取签名的链接换成您部署好的服务端http url
    // 建议通过业务登陆态检查来增强安全性，避免签名被非法获取
    $.getJSON(
'http://203.195.194.28/php
/getsign.php?type=copy&url='+encodeURIComponent(manageUrl),
             function(data) {
        var sign = data.sign,
            url = manageUrl + '?sign=' + encodeURIComponent(sign);
        $.ajax({
            type: "POST",
            url: url,
            data: {},
            success: function(ret) {
```

```
                alert('复制成功');
            },
            contentType:"multipart/form-data",
            dataType: 'json'
        });
    });
});

$('body').on('click', '#queryBtn', function(){
    var manageUrl = $('#url').text();
    if (!manageUrl) {
        alert('尚未获取管理url');
        return false;
    }
    $.ajax({
        type: "GET",
        url: manageUrl,
        data: {},
        success: function(data) {
              $('#imgInfo').text(JSON.stringify(data));
        },
        error:function(ret) {
            $('#imgInfo').text(ret.responseText);
        },
        dataType: 'json'
    });
});

function initUploadForm () {
    // 请将以下获取签名的链接换成您部署好的服务端http url
    // 建议通过业务登陆态检查来增强安全性，避免签名被非法获取
    $.getJSON('http://203.195.194.28/php/getsign.php', function(data) {
        var sign = data.sign,
            url = data.url + '?sign=' + encodeURIComponent(sign);
```

```
        var options = {
                    type: 'post',
                    url: url,
                    dataType: 'json',
                    contentType:"multipart/form-data",
            success:function(ret) {
                $('#downloadUrl').html(ret.data.download_url);
                $('#fileid').text(ret.data.fileid);
                $('#url').text(ret.data.url);
                $('#downloadRet').show();
            },
            error:function (ret) {
                alert(ret.responseText);
            }
        };

        // pass options to ajaxForm
        $('#uploadForm').ajaxForm(options);
    });
}

</script>
</body>
```

2 测试

访问web前端入口，进行测试调用，验证各种操作能够正常执行。

Java鉴权服务部署示例

本文档介绍了腾讯云万象优图服务端java的部署和集成，搭建一个java+tomcat为基础，对web端或者移动端提供http签名接口服务的例子程序。

注意：本文档只是简单的示例，展示了服务端为终端提供签名的基本示例，开发者务必根据自身业务开发相应的鉴权服务逻辑，并集成到自身服务器中。

1 环境准备

下面以在腾讯云云服务器CentOS 6.2 64位上安装nginx为例，简单介绍如何将腾讯云万象优图集成，对web端或者移动端提供http签名接口服务所需要的基础环境搭建。开发者可以根据自己业务的需要，构建http或者非http服务，为自身业务的web端、移动端提供签名。

1.1 安装nginx

```
yum install nginx -y
service nginx restart
```

1.2 验证nginx

直接访问云服务器ip地址，验证nginx是否已经运行起来。

2 环境安装配置

2.1 java安装

有网络的情况下，可通过以下命令行安装。

```
yum install [java□□]
//java□□□□□□□□□□
yum -y search java
```

无网络的条件下可以去官网下载安装。

http://www.oracle.com/technetwork/java/javase/downloads/index.html

2.2 配置java环境变量

编辑/etc/profile（或者其他可配置环境变量的文件)

```
export JAVA_HOME=/usr/java/jdk1.7.0_51
export JAVA_BIN=/usr/java/jdk1.7.0_51/bin
export PATH=$PATH:$JAVA_HOME/bin
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export TOMCAT_HOME=/usr/local/tomcat
export CATALINA_HOME=$TOMCAT_HOME
export PATH=$PATH:$TOMCAT_HOME/bin
export JAVA_HOME JAVA_BIN PATH CLASSPATH TOMCAT_HOME CATALINA_HOME
```

配置完成后，需要source文件来载入配置。

2.3 配置web container

修改/etc/nginx/conf.d/default.conf如下：

```
#
# The default server
#
server {
    listen       80 default_server;
    server_name  _;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location ^~ /java/ {
```

```
        proxy_pass http://localhost:8080;
    }
}
```

修改配置完成后，需要执行以下命令重新加载配置。

```
nginx -s reload
```

2.4 tomcat安装

tomcat可以直接下载安装，下载地址：http://tomcat.apache.org/ 。

2.5 启动tomcat

执行tomcat安装目录下bin内的startup.sh

3 下载java sdk

在网络可用的情况下可以直接用git命令行下载。
git clone https://github.com/tencentyun/java-sdk.git
也可以直接在git网站下载，网址如下:
https://github.com/tencentyun/java-sdk/archive/master.zip

4 开发鉴权服务器逻辑

本节介绍一个简单的鉴权服务器代码示例，开发者需要根据自身业务逻辑开发相应的代码。
注意：如果开发者想根据此示例进行简单的测试，请将代码中相应的项目信息替换为自己项目的信息，具体见注释。
鉴权服务器代码的示例如下：(getsign.jsp)

```
<%@ page language="java" import="com.qcloud.*,org.json.*" pageEncoding="UTF-8"%>
```

```
<%
        //请将下面的APP_ID_V2，SECRET_ID_V2，SECRET_KEY_V2，BUCKET替换为开发者自己的项目信息
        int APP_ID_V2 = 10000001;  //项目ID
        String SECRET_ID_V2 =
"AKIDNZwDVhbRtdGkMZQfWgl2Gnn1dhXs95C0"; //项目SecretID
        String SECRET_KEY_V2 =
"ZDdyyRLCLv1TkeYOl5OCMLbyH4sJ40wp";  //项目SecretKey
        String BUCKET = "testa"; //空间名称bucket

        String type = request.getParameter("type");
        String fileid = request.getParameter("fileid");

        PicCloud pc = new
 PicCloud(APP_ID_V2, SECRET_ID_V2, SECRET_KEY_V2, BUCKET);
        String sign = "";
        long expired = System.currentTimeMillis() / 1000 + 3600;
        if(null == type || "".equals(type)){
                sign = pc.GetSign(expired);
        }else if("upload".equals(type)){
                sign = pc.GetSign(expired);
        }else if("copy".equals(type) ||
                        "del".equals(type) ||
                        "download".equals(type) ){
                sign = pc.GetSignOnce(fileid);
        }else{
                sign = "";
        }

        JSONObject jsonObject = new JSONObject();
        jsonObject.put("ver", "V2");
        jsonObject.put("sign", sign.toString());
        out.print(jsonObject.toString());
```

```
%>
```

开发完成后编译，注意，需要将依赖库也打包。

5 开发部署到tomcat

Tomcat的部署目录在安装目录/webapps下面。

部署包的目录结构例如：

```
drwxr-xr-x 2 root root 4096 Jul 24 13:40 META-INF
drwxr-xr-x 4 root root 4096 Jul 24 13:40 WEB-INF
-rw-r--r-- 1 root root  836 Jul 24 13:40 getsign.jsp
-rw-r--r-- 1 root root  462 Jul 24 13:40 index.html
```

6 测试

1. 终端通过CGI：
   http://203.195.194.28/java/getsign.jsp?type=[opType]&fileid=[fileid]来获取相应的签名。
   opType：可取值：upload(上传), stat（查询），copy（复制），
   del（删除）和download（下载，如果开启token防盗链）；
   fileid：是图片资源的唯一标识；当opType为upload时，如果开发者没有指定fileid，fileid置空，否则指定为相应的fileid；下载签名，fileid可以空，也可以为开发者查看的图片fileid。
   注意： 下载签名只有开发者在控制台上面设置了token防盗链时才使用，如果没有token防盗链，不需要下载签名，直接使用下载url下载图片。
   示例：

```
http://203.195.194.28/java/getsign.jsp?type=download&fileid=sample123
http://203.195.194.28/java/getsign.jsp?type=upload&fileid=sample123
http://203.195.194.28/java/getsign.jsp?type=copy&fileid=sample123
http://203.195.194.28/java/getsign.jsp?type=del&fileid=sample123
http://203.195.194.28/java/getsign.jsp?type=stat&fileid=sample123
```

2

通过web端js或者移

动端程序请求以上http接口获取签名，

上传图片。Web端js示例请参考web端部署与SDK集成

；移动端程序示例请分别参考移动端部署与SDK集成-Android和移动端部署与SDK集成-iOS。

PHP鉴权服务部署示例

本文档介绍了腾讯云万象优图服务端php的部署和集成，搭建一个php+nginx为基础，对web端或者移动端提供http签名接口服务的例子程序。
注意：本文档只是简单的示例，展示了服务端为终端提供签名的基本示例，开发者务必根据自身业务开发相应的鉴权服务逻辑，并集成到自身服务器中。

1 环境准备

下面以在腾讯云云服务器CentOS 6.2 64位上安装nginx为例，简单介绍如何将腾讯云万象优图集成，对web端或者移动端提供http签名接口服务所需要的基础环境搭建。开发者可以根据自己业务的需要，构建http或者非http服务，为自身业务的web端、移动端提供签名。

1.1 安装nginx

```
yum install nginx -y
service nginx restart
```

1.2 验证nginx

直接访问云服务器ip地址，验证nginx是否已经运行起来。

2 安装配置PHP环境

下面介绍安装php-fpm和配置web container的详细步骤。
1 安装php-fpm

```
yum install -y php php-fpm
service php-fpm restart
```

2 配置web container
修改/etc/nginx/conf.d/default.conf如下：

```
#
# The default server
#
server {
    listen       80 default_server;
    server_name  _;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    location ~ \.php$ {
        root           /data/www/tencentyun/;
        fastcgi_pass   127.0.0.1:9000;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME  $document_root$fastcgi_script_name;
        include        fastcgi_params;
    }
}
```

3 重新加载nginx配置

修改配置完成后，需要执行以下命令重新加载配置。

```
nginx -s reload
```

3 部署集成PHP SDK及鉴权服务器逻辑开发

将sdk集成到开发者代码，开发鉴权服务逻辑，这里给一个简单的示例：

注意: 如果开发者想按照本示例做简单地测试，需要修改sdk中Tencentyun文件夹下Conf.php中APPID、SECRET_ID、SECRET_KEY为您在万象优图对应的鉴权信息；并将下列鉴权服务逻辑代码中的相应字段信息替换为自身的项目信息。

3.1 使用composer方式

1 安装composer到您的项目，这里是tencentyun目录下的php

```
cd /data/www/tencentyun/php
  curl -sS https://getcomposer.org/installer | php
```

2 为php项目指定依赖关系

```
{
    "require":{
        "tencentyun/php-sdk":"2.0.*"
    }
  }
```

3 安装依赖项

```
php composer.phar install
```

4 将sdk集成到开发者代码，开发鉴权服务逻辑，这里以php目录下getsign.php为例：
注意：如果开发者想按照本示例做简单地测试，需要将下面代码中的相应字段替换为自己的项目信息，具体见注释。

```
<?php

require 'vendor/autoload.php';

if (isset($_GET['type'])) {
    $type = $_GET['type'];
} else {
    $type = 'upload';
```

```
}

//以下bucket，projectId信息请到http://console.qcloud.com/image/bucket获取,并替换为自己的项目信息
$bucket = 'test0706';      // 空间名称
$projectId = '10000037';   // 项目ID
$userid = 0;               // 用户ID 可以自定义 默认为0

switch ($type) {
    case 'upload':
        $fileid = '/u/can/use/slash/sample'
.time();                                  // 自定义文件名
        //生成新的上传签名
        $url = Tencentyun\ImageV2::generateResUrl($bucket, $userid, $fileid);
        $expired = time() + 999;
        $sign = Tencentyun\Auth::getAppSignV2($bucket, $fileid, $expired);
        $ret = array('url' => $url,'sign' => $sign);
        exit(json_encode($ret));
    case 'stat':
        $fileid = rawurldecode($_GET['fileid']);
        $url = Tencentyun\ImageV2::generateResUrl($bucket, $userid, $fileid);
        $ret = array('url' => $url);
        exit(json_encode($ret));
    case 'del':
    case 'copy':
        $fileid = rawurldecode($_GET['fileid']);
        $url = Tencentyun\ImageV2::generateResUrl($bucket, $userid, $fileid, $ty
pe);
        $sign = Tencentyun\Auth::getAppSignV2($bucket, $fileid, 0);
        $ret = array('url' => $url,'sign' => $sign);
        exit(json_encode($ret));
    case 'download':
        $fileid = rawurldecode($_GET['fileid']);
        $expired = time() + 999;
        $sign = Tencentyun\Auth::getAppSignV2($bucket, $fileid, $expired);
```

```
        $ret = array('sign' => $sign);
        exit(json_encode($ret));
    default:
        exit;
}

//end of script
```

3.2 直接源码集成

1 从github下载源码到php目录下的tencentyun目录

```
git clone https://github.com/tencentyun/php-sdk.git tencentyun
```

2 将sdk集成到开发者代码，开发鉴权服务逻辑，这里以php目录下getsigninclude.php为例：

注意：如果开发者想按照本示例做简单地测试，需要将下面代码中的相应字段替换为自己的项目信息，具体见注释。

```
<?php

require 'tencentyun/include.php';

//以下bucket，projectId信息请到http://console.qcloud.com/image/bucket获取,并替换为自己的项目信息
$bucket = 'test0706';      // 空间名称
$projectId = '10000037';   // 项目ID
$userid = 0;               // 用户ID 可以自定义 默认为0

switch ($type) {
    case 'upload':
        $fileid = '/u/can/use/slash/sample'
.time();                                  // 自定义文件名
        //生成新的上传签名
```

```
        $url = Tencentyun\ImageV2::generateResUrl($bucket, $userid, $fileid);
        $expired = time() + 999;
        $sign = Tencentyun\Auth::getAppSignV2($bucket, $fileid, $expired);
        $ret = array('url' => $url,'sign' => $sign);
        exit(json_encode($ret));
    case 'stat':
        $fileid = rawurldecode($_GET['fileid']);
        $url = Tencentyun\ImageV2::generateResUrl($bucket, $userid, $fileid);
        $ret = array('url' => $url);
        exit(json_encode($ret));
    case 'del':
    case 'copy':
        $fileid = rawurldecode($_GET['fileid']);
        $url = Tencentyun\ImageV2::generateResUrl($bucket, $userid, $fileid, $ty
pe);
        $sign = Tencentyun\Auth::getAppSignV2($bucket, $fileid, 0);
        $ret = array('url' => $url,'sign' => $sign);
        exit(json_encode($ret));
    case 'download':
        $fileid = rawurldecode($_GET['fileid']);
        $expired = time() + 999;
        $sign = Tencentyun\Auth::getAppSignV2($bucket, $fileid, $expired);
        $ret = array('sign' => $sign);
        exit(json_encode($ret));
    default:
        exit;
}

//end of script
```

4 测试

1. 终端通过CGI：

http://203.195.194.28/php/getsignv2.php?type=[opType]&fileid=[fileid]来获取相应的签名。

opType：可取值：upload(上传), stat（查询），copy（复制），

del（删除）和download（下载，如果开启token防盗链）；

fileid：是图片资源的唯一标识；当opType为upload时，如果开发者没有指定fileid，fileid置空，否则指定为相应的fileid；下载签名，fileid可以空，也可以为开发者查看的图片fileid。

注意： 下载签名只有开发者在控制台上面设置了token防盗链时才使用，如果没有token防盗链，不需要下载签名，直接使用下载url下载图片。

示例：

```
http://203.195.194.28/php/getsignv2.php?type=upload&fileid=sample123
 http://203.195.194.28/php/getsignv2.php?type=copy&fileid=sample123
 http://203.195.194.28/php/getsignv2.php?type=stat&fileid=sample123
 http://203.195.194.28/php/getsignv2.php?type=del&fileid=sample123
 http://203.195.194.28/php/getsignv2.php?type=download&fileid=sample123
```

2

通过web端js或者移

动端程序请求以上http接口获取签名，

上传图片。Web端js示例请参考web端部署与SDK集成

；移动端程序示例请分别参考移动端部署与SDK集成-Android和移动端部署与SDK集成-iOS。

Python鉴权服务部署示例

本文档介绍了腾讯云万象优图服务端python的部署和集成，搭建一个python+uWSGI+nginx为基础，对web端或者移动端提供http签名接口服务的例子程序。

注意：本文档只是简单的示例，展示了服务端为终端提供签名的基本示例，开发者务必根据自身业务开发相应的鉴权服务逻辑，并集成到自身服务器中。

1 环境准备

下面以在腾讯云云服务器CentOS 6.2 64位上安装nginx为例，简单介绍如何将腾讯云万象优图集成，对web端或者移动端提供http签名接口服务所需要的基础环境搭建。开发者可以根据自己业务的需要，构建http或者非http服务，为自身业务的web端、移动端提供签名。

1.1 安装nginx

```
yum install nginx -y
service nginx restart
```

1.2 验证nginx

直接访问云服务器ip地址，验证nginx是否已经运行起来。

2 安装配置环境

2.1 安装uWSGI

执行以下命令安装uWSGI。

```
yum install -y uwsgi uwsgi-plugin-python
```

2.2 配置web container

修改/etc/nginx/conf.d/default.conf如下：

```
#
# The default server
#
server {
    listen       80 default_server;
    server_name  _;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location ^~ /python/ {
       uwsgi_pass localhost:9003;
       include /etc/nginx/uwsgi_params
    }
}
```

修改配置完成后，需要执行以下命令重新加载配置。

```
nginx -s reload
```

3 部署Python SDK

执行以下命令部署python SDK。

```
cd /data/www/tencentyun/python
pip install tencentyun
```

4 开发鉴权服务逻辑

1 集成SDK开发代码

将SDK集成到开发者代码，开发鉴权服务逻辑，这里以python目录下getsignv2.py为例。开发者需要根据自身业务逻辑开发相应的鉴权服务。

注意：如果开发者想根据此示例进行简单的测试，请将代码中相应的项目信息替换为自己项目的信息，具体见注释。

```
# -*- coding: utf-8 -*-

import time
from cgi import parse_qs, escape
from urllib import unquote
import json
import tencentyun

def application(env, start_response):
    start_response('200 OK', [('Content-Type','application/json')])
    d = parse_qs(env['QUERY_STRING'])
    print d
    //请将下面的projectid′bucket′secret_id ′secret_key 等信息替换为开发者自己的项目信息
    projectid = '10000037'
    bucket = 'test0706'
    userid = '0'
    secret_id = 'AKIDpoKBfMK7aYcYNlqxnEtYA1ajAqji2P7T'
    secret_key = 'P4FewbltIpGeAbwgdrG6eghMUVlpmjIe'

    image = tencentyun.ImageV2(projectid,secret_id,secret_key)

    type = 'upload'
    fileid = ''
    if d.has_key('type') :
        type = d.get('type', [''])[0]
    if d.has_key('fileid') :
        fileid = unquote(d.get('fileid', [''])[0])
```

```
    if 'upload' == type:
        fileid = '/u/can/use/slash/sample'+str(int(time.time()))
        expired = int(time.time()) + 999
        url = image.generate_res_url_v2(bucket,userid,fileid)
        auth = tencentyun.Auth(secret_id,secret_key)
        sign = auth.get_app_sign_v2(bucket, fileid, expired)
        ret = {'url':url, 'sign':sign}
    elif 'download' == type:
        expired = int(time.time()) + 999
        auth = tencentyun.Auth(secret_id,secret_key)
        url = image.generate_res_url_v2(bucket,userid,fileid)
        sign = auth.get_app_sign_v2(bucket, fileid, expired)
        ret = {'sign':sign}
    else:
        if not fileid:
            ret = {'error':'params error'}
        else :
            if 'stat' == type:
                type = ''
            expired = 0
            auth = tencentyun.Auth(secret_id,secret_key)
            url = image.generate_res_url_v2(bucket,userid,fileid,type)
            sign = auth.get_app_sign_v2(bucket, fileid, expired)
            ret = {'url':url, 'sign':sign}

    return json.dumps(ret)
```

2 配置uWSGI配置文件

```
[uwsgi]
vhost = false
uid = nginx
gid = www
```

```
plugins = python
socket = 127.0.0.1:9003
master = true
enable-threads = true
workers = 1
wsgi-file = /data/www/tencentyun/python/getsignv2.py
chdir = /data/www/tencentyun/python/
```

3 运行程序

```
cd /data/www/tencentyun/python
nohup uwsgi --ini /data/www/tencentyun/python/uwsgi.ini &
```

5 测试

1. 终端通过CGI：http://203.195.194.28/python?type=[opType]&fileid=[fileid]来获取相应的签名。
   opType：可取值：upload(上传), stat（查询），copy（复制），
   del（删除）和download（下载，如果开启token防盗链）；
   fileid：是图片资源的唯一标识；当opType为upload时，如果开发者没有指定fileid，fileid置空，否则指定为相应的fileid；下载签名，fileid可以空，也可以为开发者查看的图片fileid。
   注意： 下载签名只有开发者在控制台上面设置了token防盗链时才使用，如果没有token防盗链，不需要下载签名，直接使用下载url下载图片。
   示例：

```
http://203.195.194.28/python/?type=download&fileid=sample123
 http://203.195.194.28/python/?type=upload&fileid=sample123
 http://203.195.194.28/python/?type=copy&fileid=sample123
 http://203.195.194.28/python/?type=del&fileid=sample123
 http://203.195.194.28/python/?type=stat&fileid=sample123
```

2

通过web端js或者移

动端程序请求以上http接口获取签名，

上传图片。Web端js示例请参考web端部署与SDK集成

；移动端程序示例请分别参考移动端部署与SDK集成-Android和移动端部署与SDK集成-iOS。

Nodejs鉴权服务部署示例

本文档介绍腾讯云·万象优图服务端nodejs的部署和集成，搭建一个nodejs+nginx为基础，对web端或者移动端提供http签名接口服务的例子程序。

注意：本文档只是简单的示例，展示了服务端为终端提供签名的基本示例，开发者务必根据自身业务开发相应的鉴权服务逻辑，并集成到自身服务器中。

1 环境准备

下面以在腾讯云云服务器CentOS 6.2 64位上安装nginx为例，简单介绍如何将腾讯云万象优图集成，对web端或者移动端提供http签名接口服务所需要的基础环境搭建。开发者可以根据自己业务的需要，构建http或者非http服务，为自身业务的web端、移动端提供签名。

1.1 安装nginx

```
yum install nginx -y
service nginx restart
```

1.2 验证nginx

直接访问云服务器ip地址，验证nginx是否已经运行起来。

2 安装配置Nodejs环境

下面介绍安装Nodejs和配置web container的详细步骤。

1 安装Nodejs

```
yum install -y nodejs npm
```

2 配置web container

修改/etc/nginx/conf.d/default.conf如下：

```
#
# The default server
#
server {
    listen       80 default_server;
    server_name  _;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location ^~ /node/ {
        proxy_pass http://localhost:9002;
    }
}
```

3 重新加载nginx配置

修改配置完成后，需要执行以下命令重新加载配置。

```
nginx -s reload
```

3 安装Nodejs SDK

执行以下命令安装Nodejs SDK。

```
cd /data/www/tencentyun/node
npm install tencentyun
```

4 开发鉴权服务逻辑

将sdk集成到开发者代码，开发鉴权服务逻辑，这里以node目录下getsignv2.js为例（开发者务必根据自身业务开发相应的鉴权服务逻辑）：

注意：如果开发者想按照本示例做简单地测试，需要将下面代码中的相应字段替换为自己的项目信息，具体见代码注释。

```
var http=require('http');
var url = require('url');
var util = require('util');
var tencentyun = require('tencentyun');

var server=new http.Server();
server.on('request',function(req,res){

    var urlinfo = url.parse(req.url,true),
        type = 'upload';

    if (urlinfo.query && urlinfo.query.type) {
        type = urlinfo.query.type;
    }

    //□□□□□bucket, projectId, secretId□secretKey□□□□□□□□□□□□□
    var bucket = 'test0706',
        projectId = '10000037',
        userid = 0,
        secretId = 'AKIDpoKBfMK7aYcYNlqxnEtYA1ajAqji2P7T',
        secretKey = 'P4FewbltIpGeAbwgdrG6eghMUVlpmjIe';

    tencentyun.conf.setAppInfo(projectId, secretId, secretKey);

    var error = false;

    switch(type) {
        case 'upload':
            var fileid = '/u/can/use/slash/sample' + Math.round(+new Date()/
1000),
                expired = Math.round(+new Date()/1000) + 999,
```

```
                uploadurl = tencentyun.imagev2.generateResUrlV2(bucket, userid,
fileid),
                sign = tencentyun.auth.getAppSignV2(bucket, fileid, expired);
                ret = {'sign':sign,'url':uploadurl};
            break;
        case 'stat':
            if (!urlinfo.query || !urlinfo.query.fileid) {
                error = true;
            } else {
                var fileid = decodeURIComponent(urlinfo.query.fileid),
                    otherurl = tencentyun.imagev2.generateResUrlV2(bucket, useri
d, fileid),
                    ret = {'url':otherurl};
            }
            break;
        case 'del':
        case 'copy':
            if (!urlinfo.query || !urlinfo.query.fileid) {
                error = true;
            } else {
                var fileid = decodeURIComponent(urlinfo.query.fileid),
                    otherurl = tencentyun.imagev2.generateResUrlV2(bucket, useri
d, fileid, type),
                    sign = tencentyun.auth.getAppSignV2(bucket, fileid, 0);
                    ret = {'sign':sign,'url':otherurl};
            }
            break;
        case 'download':
            if (!urlinfo.query || !urlinfo.query.fileid) {
                error = true;
            } else {
                var fileid = decodeURIComponent(urlinfo.query.fileid),
                    expired = Math.round(+new Date()/1000) + 999,
                    sign = tencentyun.auth.getAppSignV2(bucket, fileid, expired)
```

```
;
                    ret = {'sign':sign};
            }
            break;
    }

    res.writeHead(200,{'Content-Type':'application/json'});
    if (error) {
        res.end({'error':'params error'});
    } else {
        res.end(JSON.stringify(ret));
    }
});

server.listen(9002);
console.log('HTTP SERVER is LISTENING AT PORT 9002.');
```

5 运行程序

```
cd /data/www/tencentyun/node
nohup node getsignv2.js &
```

6 测试

1. 终端通过CGI：http://203.195.194.28/node/?type=[opType]&fileid=[fileid]来获取相应的签名。
   opType：可取值：upload(上传), stat（查询），copy（复制），
   del（删除）和download（下载，如果开启token防盗链）；
   fileid：是图片资源的唯一标识；当opType为upload时，如果开发者没有指定fileid，fileid置空，否则指定为相应的fileid；下载签名，fileid可以空，也可以为开发者查看的图片fileid。
   注意： 下载签名只有开发者在控制台上面设置了token防盗链时才使用，如果没有token防盗链，不需要下载签名，直接使用下载url下载图片。
   示例：

```
http://203.195.194.28/node/?type=del&fileid=sample123
http://203.195.194.28/node/?type=copy&fileid=sample123
http://203.195.194.28/node/?type=stat&fileid=sample123
http://203.195.194.28/node/?type=download&fileid=sample123
http://203.195.194.28/node/?type=upload&fileid=sample123
```

2

通过web端j

s或者移动端程序请求以

上http接口获取签名，上传图片。Web端js示例

请参考web端部署与SDK集成；移动端程序示例请分别参考移动端部署与SDK集成-Android和移动端部署与SDK集成-iOS。

GO鉴权服务部署示例

本文档介绍腾讯云万象优图服务端go的部署和集成，搭建一个go+nginx为基础，对web端或者移动端提供http签名接口服务的例子程序。
注意：本文档只是简单的示例，展示了服务端为终端提供签名的基本示例，开发者务必根据自身业务开发相应的鉴权服务逻辑，并集成到自身服务器中。

1 环境准备

下面以在腾讯云云服务器CentOS 6.2 64位上安装nginx为例，简单介绍如何将腾讯云万象优图集成，对web端或者移动端提供http签名接口服务所需要的基础环境搭建。开发者可以根据自己业务的需要，构建http或者非http服务，为自身业务的web端、移动端提供签名。

1.1 安装nginx

yum install nginx –y
service nginx restart

1.2 验证nginx

直接访问云服务器ip地址，验证nginx是否已经运行起来。

2 安装配置Go环境

go的web开发框架可以自由选择，这里以beego为例，beego详见Homepage-beego。

2.1 beego安装

安装beego

```
go get github.com/astaxie/beego
```

安装bee

```
go get github.com/beego/bee
```

2.2 配置bee环境

bee安装完成后配置环境变量PATH。

```
export PATH=$PATH:$GOPATH/bin
```

2.3 配置web container

修改/etc/nginx/conf.d/default.conf如下：

```
#
# The default server
#
server {
    listen       80 default_server;
    server_name  _;

    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location ^~ /go/ {
        proxy_pass http://localhost:9001;
    }
}
```

修改配置完成后，需要执行以下命令重新加载配置。

```
nginx -s reload
```

2.4 创建项目

在任意目录下创建新应用服务器项目，这里项目名采用signServer。

```
bee new signServer
```

3 集成Go SDK

在开发环境下直接执行如下命令来集成Go SDK。

```
go get github.com/tencentyun/go-sdk
```

4 开发鉴权服务逻辑

以下代码是一个简单的示例，开发者需要根据自己的业务需求来开发相应代码。
注意：如果开发者想根据此示例进行简单的测试，请将conf目录下的app.conf配置文件中相应的项目信息替换为自己项目的信息，具体见下面“配置的读取”注释。

1. 代码开发
   在之前建立的signServer项目controllers目录下编写sign.go

```
package controllers

import (
        "encoding/json"
        "fmt"

        "github.com/astaxie/beego"
        "github.com/tencentyun/go-sdk"
)
```

```
type signData struct {
        Version string `json:"ver"`
        Sign    string `json:"sign"`
}

type SignController struct {
        beego.Controller
}

func (c *SignController) Get() {
        appid, _ := beego.AppConfig.Int("appid")
        sid := beego.AppConfig.String("sid")
        skey := beego.AppConfig.String("skey")
        bucket := beego.AppConfig.String("bucket")

        types := c.Input().Get("type")
        fileid := c.Input().Get("fileid")

        var data signData
        cloud := qcloud.PicCloud{uint(appid), sid, skey, bucket}
        data.Version = "V2"
        if types == "" || types == "upload" {
                data.Sign, _ = cloud.Sign(3600)
        } else if types == "copy" ||
                types == "del" ||
                types == "download" {
                data.Sign, _ = cloud.SignOnce(fileid)
        } else {
                data.Sign = ""
        }

        rsp, err := json.Marshal(data)
        if err != nil {
                fmt.Println("error: ", err.Error())
```

```
        }
        c.Ctx.WriteString(string(rsp))
}
```

1. 配置的读取

   Beego的配置文件在conf目录下的app.conf。

   开发者需要将文件中的appid(项目ID)，sid（项目SecretID），skey（项目SecretKey），bucket（空间名称）分别替换为开发者自己的项目信息。

```
appname = signServer
httpport = 9000
runmode = dev

appid = 10000001
sid = AKIDNZwDVhbRtdGkMZQfWgl2Gnn1dhXs95C0
skey = ZDdyyRLCLv1TkeYOl5OCMLbyH4sJ40wp
bucket = testa
```

3.route逻辑(routers/router.go)

```
package routers

import (
    "github.com/tencentyun/go-sdk/sample/signServer/controllers"
    "github.com/astaxie/beego"
)

func init() {
    beego.Router("/", &controllers.MainController{})
    beego.Router("/go/getsign", &controllers.SignController{})
}
```

5 编译运行beego

在signServer项目目录下执行如下命令即可。

```
bee run
```

go将编译应用服务器到一个binary文件，例如上述例子里为signServer。可以随便复制到任何地方执行（连带conf，statics和views，这里包含一些配置、静态资源和模板）。
以后只需要直接执行binary文件即可。不需要多次执行bee run。

6 测试

1. 终端通过CGI：
   http://203.195.194.28/go/getsign?type=[opType]&fileid=[fileid]来获取相应的签名。
   opType：可取值：upload(上传), stat（查询），copy（复制），
   del（删除）和download（下载，如果开启token防盗链）；
   fileid：是图片资源的唯一标识；当opType为upload时，如果开发者没有指定fileid，fileid置空，否则指定为相应的fileid；下载签名，fileid可以空，也可以为开发者查看的图片fileid。
   注意： 下载签名只有开发者在控制台上面设置了token防盗链时才使用，如果没有token防盗链，不需要下载签名，直接使用下载url下载图片。
   示例：

```
http://203.195.194.28/go/getsign?type=download&fileid=sample123
 http://203.195.194.28/go/getsign?type=upload&fileid=sample123
 http://203.195.194.28/go/getsign?type=copy&fileid=sample123
 http://203.195.194.28/go/getsign?type=del&fileid=sample123
 http://203.195.194.28/go/getsign?type=stat&fileid=sample123
```

2
通过web端js或者移
动端程序请求以上http接口获取签名，
上传图片。Web端js示例请参考web端部署与SDK集成

；移动端程序示例请分别参考移动端部署与SDK集成-Android和移动端部署与SDK集成-iOS。

万象优图支持字体列表

万象优图支持的字体列表如下：

Batang.TTF

CALIBRI.TTF

CANDARA.TTF

CANDARAI.TTF

CONSOLA.TTF

CORBEL.TTF

Candarab.ttf

Candaraz.ttf

Comic Sans MS.ttf

Davida Bold BT.ttf

Franklin Gothic Medium.ttf

Gabriola.ttf

LaoUI.ttf

LaoUIb.ttf

REFSAN.TTF

Shonar.ttf

Shonarb.ttf

Vani.ttf

Vanib.ttf

andlso.ttf

angsa.ttf

angsab.ttf

angsai.ttf

angsau.ttf

angsaub.ttf

angsaui.ttf

angsauz.ttf

angsaz.ttf

aparaj.ttf

aparajb.ttf

aparajbi.ttf

aparaji.ttf

arabtype.ttf

arial.ttf

browa.ttf

browab.ttf

browai.ttf

browaz.ttf

calibrib.ttf

calibrii.ttf

calibriz.ttf

cordia.ttf

cordiab.ttf

cordiai.ttf

cordiau.ttf

cordiaub.ttf

cordiaui.ttf

cordiauz.ttf

cordiaz.ttf

cour.ttf

courbd.ttf

courbi.ttf

coure.fon

couri.ttf

courier.ttf

daunpenh.ttf

dokchamp.ttf

ebrima.ttf

ebrimabd.ttf

estre.ttf

euphemia.ttf

frank.ttf

georgia.ttf

georgiab.ttf

georgiai.ttf

georgiaz.ttf

gisha.ttf

gishabd.ttf

gulim.ttc

himalaya.ttf

impact.ttf

iskpota.ttf

iskpotab.ttf

kaiu.ttf

kalinga.ttf

kalingab.ttf

kartika.ttf

kartikab.ttf

kokila.ttf

kokilab.ttf

kokilabi.ttf

kokilai.ttf

l_10646.ttf

latha.ttf

lathab.ttf

leelawad.ttf

leelawdb.ttf

lucon.ttf

lvnm.ttf

lvnmbd.ttf

majalla.ttf

majallab.ttf

malgun.ttf

malgunbd.ttf

mangal.ttf

mangalb.ttf

meiryo.ttc

meiryob.ttc

micross.ttf

mingliu.ttc

monbaiti.ttf

moolbor.ttf

mriam.ttf

mriamc.ttf

msgothic.ttc

msjh.ttf

msjhbd.ttf

msmincho.ttc

msuighur.ttf

msyi.ttf

mvboli.ttf

nrkis.ttf

ntailu.ttf

ntailub.ttf

nyala.ttf

pala.ttf

palab.ttf

palabi.ttf

palai.ttf

phagspa.ttf

phagspab.ttf

plantc.ttf

raavi.ttf

raavib.ttf

rod.ttf

segoepr.ttf

segoeprb.ttf

segoesc.ttf

segoescb.ttf

segoeui.ttf

segoeuib.ttf

segoeuii.ttf

segoeuil.ttf

segoeuiz.ttf

seguisb.ttf

seguisym.ttf

serife.fon

shruti.ttf

shrutib.ttf

simpbdo.ttf

simpfxo.ttf

simpo.ttf

smalle.fon

sserife.fon

svgafix.fon

svgasys.fon

sylfaen.ttf

symbol.ttf

tahoma.ttf

tahomabd.ttf

taile.ttf

taileb.ttf

times.ttf

timesbd.ttf

timesbi.ttf

timesi.ttf

tradbdo.ttf

trado.ttf

trebuc.ttf

trebucbd.ttf

trebucbi.ttf

trebucit.ttf

tunga.ttf

tungab.ttf

upcdb.ttf

upcdbi.ttf

upcdi.ttf

upcdl.ttf

upceb.ttf

upcebi.ttf

upcei.ttf

upcel.ttf

upcfb.ttf

upcfbi.ttf

upcfi.ttf

upcfl.ttf

upcib.ttf

upcibi.ttf

upcii.ttf

upcil.ttf

upcjb.ttf

upcjbi.ttf

upcji.ttf

upcjl.ttf

upckb.ttf

upckbi.ttf

upcki.ttf

upckl.ttf

upclb.ttf

upclbi.ttf

upcli.ttf

upcll.ttf

utsaah.ttf

utsaahb.ttf

utsaahbi.ttf

utsaahi.ttf

verdana.ttf

verdanab.ttf

verdanai.ttf

verdanaz.ttf

vga936.fon

vijaya.ttf

vijayab.ttf

vrinda.ttf

vrindab.ttf

webdings.ttf

wingding.ttf

运单识别指南

1. 背景

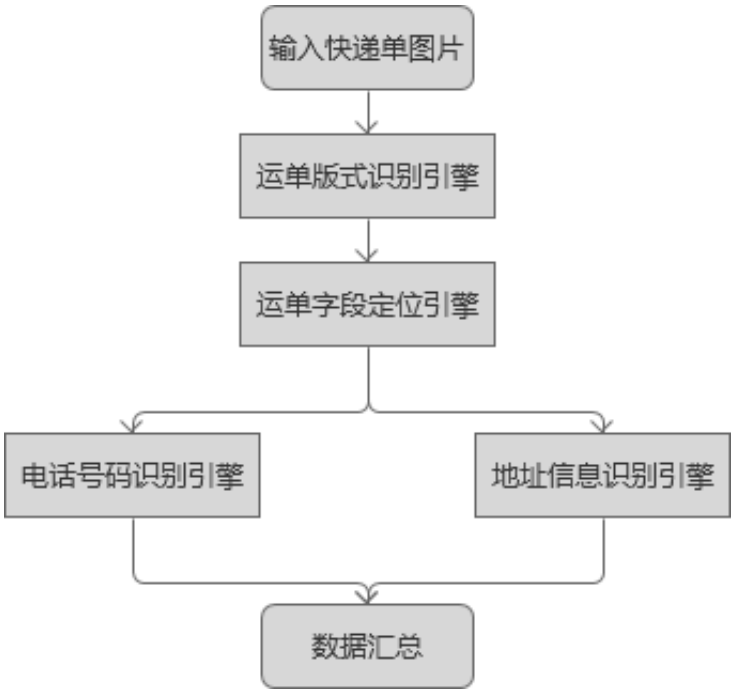
随着电子商务的不断发展，线上线下商业模式已经密不可分，其中物流起着至关重要的作用。从商家到买家、从厂家到商家，都需要通过物流来完成。通过了解发现：一件商品从揽收到送达中间要经过多个环节，其中打包中转这一环节最为繁忙，也最为重要。为了提高该环节效率，需要更加快速的识别运单上的物流信息。万象优图的运单内容自动识别，将极大的促进了这一环节的稳定高效运作。

2. 万象优图运单识别

下面是一张常见的某快递单，其中收件人地址和收件人电话是非常关键的信息，物流中转完全依赖这两个信息的完整性。

传统的运单扫描都是靠人工肉眼识别的，成本高且效率低。因此探索人工智能是否可以自动扫描运单，具有非常大的意义。

下面将为您详细介绍万象优图运单识别的流程。



- 运单版式识别

每个快递公司的运单都有不同的版式。通过大数据分析和深度学习训练，使用模式匹配方法，找出不同版式运单的特征。如下图所示，在已知版式的情况下，可以根据版式解析出图片中固定区域的内容。

- 字段定位

字段定位是针对收件人或寄件人的具体字段的区域检测，如电话号码、地址信息等。通过大量样本的训练，不断学习和适应各种运单中的版面变化、字体多样性变化、容忍各种噪声干扰等，最终训练得到准确性和稳定性非常强的字段定位的算法。字段定位效果示例如下图所示：

- 电话号码识别

运单上的电话号码通常是指11位的手机号或者8位的固定电话，基本上都是以整行的方式出现。通过长期的积累和对该场景的深入分析研究，万象优图提供了一套业内先进的整行识别技术，无需切分单字，直接识别整行手写字符。

- 地址信息识别

针对地址场景，结合运单的具体需求，我们提供独创的地址信息识别方案。借鉴数字整行识别的方法，并针对省市区地址特征和手写文本特征，改进了识别网络，使其能够适应整行手写汉字的图像特点，并直接给出对应

的省市区分类结果。整个地址识别流程简化成可以直接识别省市区信息，避免的中间其他环节的误差影响。如下图所示：

3. 接入指南

现运单识别已火热内测中，您可以提交工单申请内测资格。