

智营网优 开发者手册 产品文档



腾讯云

【版权声明】

©2013-2019 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【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

开发者手册

签名与鉴权

开发者手册

签名与鉴权

最近更新时间：2019-03-06 19:49:39

腾讯云 API 会对每个访问的请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证用户身份。签名信息由用户所执有的安全凭证生成，安全凭证包括密钥 ID（SecretId）和密钥 KEY（SecretKey）。

获得游戏 ID（GameId）、SecretId 和 SecretKey

进入腾讯云控制台产品页面，新建加速服务后，在列表页面可以得到游戏 ID（GameId）、密钥 ID（SecretId）和密钥 KEY（SecretKey），每开通一项加速服务，会产生一个“游戏 ID”，和“游戏 KEY”，游戏 KEY 默认只有创建者、管理员（全局协作者）才有查看游戏 KEY 的权限，如果您看不到相关的信息，请联系管理员查看。

生成签名字符串

有了安全凭证 GameId，SecretId 和 SecretKey 后，就可以生成签名串了。下面给出了一个生成签名串的详细过程：

假设用户的 GameId，SecretId 和 SecretKey 分别是：

GameId: 1794235
SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
SecretKey: Gu5t9xGARNpq86cd98joQYCN3Cozk1qA

⚠ 注意：

这里只是示例，请用户根据自己实际的 SecretId 和 SecretKey 进行后续操作。

以开通 4G 加速服务请求为例，当用户调用这一接口时，其请求参数如下：

参数名称	中文	参数值
Action	方法名	open
GameId	游戏 ID	1794235

参数名称	中文	参数值
SecretId	密钥 ID	AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
PhoneNO	用户手机标识（按运营商接口返回的字符串）	13788282828
Timestamp	当前时间戳	1496203804
Nonce	随机正整数	1038417
DeviceCode	手机设备号（可不填）	xxx-yyy
VersionId	游戏版本信息（可不填）	1794235

对参数排序

首先对所有请求参数按参数名做字典序升序排列，所谓字典序升序排列，直观上就如同在字典中排列单词一样排序，按照字母表或数字表里递增顺序的排列次序，即先考虑第一个“字母”，在相同的情况下考虑第二个“字母”，依此类推。您可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 `ksort` 函数。上述示例参数的排序结果如下：

```
{
  "Action": "open",
  "DeviceCode": "xxx-yyy",
  "GameId": "2001902634",
  "Nonce": 1038417,
  "PhoneNO": 13788282828,
  "SecretId": "AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA",
  "Timestamp": 1465185768,
  "VersionId": "1.2.3.0"
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

拼接请求字符串

此步骤生成请求字符串。

把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对 `Action` 参数，其参数名称为“Action”，参数值为“open”，因此格式化后即为 `Action=open`。

⚠ 注意：

“参数值”为原始值而非 URL 编码后的值。

将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=open&DeviceCode=xxx-yyy&Gameld=1794235&Nonce=1038417&PhoneNO=13788282828&ProjectId=1006972&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1496203804&VersionId=1794235
```

拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

- 请求方法：支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
- 请求主机：qos.api.cloud.tencent.com。
- 请求路径：请求路径固定为 /qos。
- 请求字符串：即生成的 [请求字符串](#)。

签名原文串的拼接规则为：请求方法 + 请求主机 + 请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETqos.api.qcloud.com/qos?Action=open&DeviceCode=xxx-yyy&Gameld=1794235&Nonce=1038417&PhoneNO=13788282828&ProjectId=1006972&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1496203804&VersionId=1794235
```

生成签名串

首先使用签名算法（HmacSHA256）对上一步中获得的签名原文字符串进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例，由于本例中所用的签名算法为 HmacSHA256，因此生成签名串的代码如下：

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3Cozk1qA'; //密钥KEY
$srcStr =
    'GETqos.qcloud.com/qos?Action=open&DeviceCode=xxx-yyy&Gameld=1794235&Nonce=1038417&PhoneNO=13788282828&ProjectId=1006972&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&Timestamp=1496203804&VersionId=1794235';
$signStr = base64_encode(hash_hmac('sha256', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为：

```
ORFGm9wSTil++b/NAIG63NRuEhA0x1AjXvrg72yls5Y=
```

使用其它程序设计语言开发时, 可用上面示例中的原文进行签名验证, 得到的签名串与例子中的一致即可。

签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 ORFGm9wSTil++b/NAIG63NRuEhA0x1AjXvrg72yls5Y=，则其编码后为 ORFGm9wSTil++b/NAIG63NRuEhA0x1AjXvrg72yls5Y=。因此，最终得到的签名串请求参数(Signature)为：ORFGm9wSTil++b/NAIG63NRuEhA0x1AjXvrg72yls5Y=，它将用于生成最终的请求 URL。

⚠ 注意：

如果用户的请求方法是 GET，则对所有请求参数值均需要做 URL 编码；部分语言库会自动对 URL 进行编码，重复编码会导致签名校验失败。

鉴权失败

当鉴权不通过时，可能出现的错误如下表：

错误代码	错误类型	错误描述
1	鉴权失败	签名验证失败，请确保您请求参数中的 Signature 计算正确；也可能是密钥状态有误，请确保 API 密钥有效且未被禁用。
2	查询IP数据库失败	请确认 IP 是否正确
3	运营商不支持加速	用户所处地区的运营商加速服务未开通
4	请求 URL 异常	请求 URL 异常
5	不满足开启条件	不满足开启条件
6	系统错误	系统出现异常错误
7	获取手机号 URL 失败	获取手机号 URL 失败
8	开启失败	开启失败

错误代码	错误类型	错误描述
9	关闭失败	关闭失败