

专线接入 API文档 产品文档



腾讯云

【版权声明】

©2013-2018 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

API文档

简介

API 概览

调用方式

请求结构

公共参数

接口鉴权

返回结果

更新历史

专线接入相关接口

接受专用通道申请

创建专用通道

删除专用通道

查询专用通道列表

修改专用通道属性

拒绝专用通道申请

数据结构

专线接入 API 2017

简介

API概览

物理专线

查询专线列表

专线通道

创建专线通道

修改专线通道

删除专线通道

查询专线通道列表

接受专线通道申请

拒绝专线通道申请

调用方式

请求结构

接口鉴权

公共参数

返回值

返回值结构

错误码

异步任务接口返回格式

示例代码

API文档

简介

最近更新时间：2018-09-14 15:12:02

专线接入 API 升级到 3.0 版本，全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验；支持全地域就近接入让您更快连接腾讯云产品。

欢迎使用腾讯云专线接入（DC）。专线接入提供了一种快速安全连接腾讯云与本地数据中心的方法，用户可以通过一条物理专线一次性打通位于多地域的腾讯云计算资源，实现灵活可靠的混合云部署。本文档提供的 API 供您使用请求调用的方式来操作专线接入。请确保在使用这些接口前，已充分了解 DC 产品、及其使用和收费方式。

在本文档的接口说明部分，凡出现任何参数可选范围等方面与腾讯云官网上给出的数值发生矛盾时，均以官网上给出的值为准。

API 概览

最近更新时间：2018-09-14 15:12:02

专线接入相关接口

接口名称	接口功能
AcceptDirectConnectTunnel	接受专用通道申请
CreateDirectConnectTunnel	创建专用通道
DeleteDirectConnectTunnel	删除专用通道
DescribeDirectConnectTunnels	查询专用通道列表
ModifyDirectConnectTunnelAttribute	修改专用通道属性
RejectDirectConnectTunnel	拒绝专用通道申请

调用方式

请求结构

最近更新时间：2018-09-18 16:59:39

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 `dc.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `dc.ap-guangzhou.tencentcloudapi.com`。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到**最近的**某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 `dc.ap-guangzhou.tencentcloudapi.com` 是一致的。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	<code>dc.tencentcloudapi.com</code>
华南地区(广州)	<code>dc.ap-guangzhou.tencentcloudapi.com</code>
华东地区(上海)	<code>dc.ap-shanghai.tencentcloudapi.com</code>
华北地区(北京)	<code>dc.ap-beijing.tencentcloudapi.com</code>
西南地区(成都)	<code>dc.ap-chengdu.tencentcloudapi.com</code>
西南地区(重庆)	<code>dc.ap-chongqing.tencentcloudapi.com</code>
东南亚地区(首尔)	<code>dc.ap-seoul.tencentcloudapi.com</code>
东南亚地区(新加坡)	<code>dc.ap-singapore.tencentcloudapi.com</code>
亚太地区(孟买)	<code>dc.ap-mumbai.tencentcloudapi.com</code>
美国西部(硅谷)	<code>dc.na-siliconvalley.tencentcloudapi.com</code>
美国东部(弗吉尼亚)	<code>dc.na-ashburn.tencentcloudapi.com</code>

注意：由于**金融区**和**非金融区**是隔离不互通的，因此当访问金融区服务时（公共参数 `Region` 为金融区地域），需要同时指定带金融区地域的域名，最好和 `Region` 的地域保持一致。

金融区接入地域	金融区域名
---------	-------

金融区接入地域	金融区域名
华东地区(上海金融)	dc.ap-shanghai-fsi.tencentcloudapi.com
华南地区(深圳金融)	dc.ap-shenzhen-fsi.tencentcloudapi.com

2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持 POST 和 GET 请求。POST 请求目前仅支持 Content-Type 类型为 application/x-www-form-urlencoded。

4. 字符编码

均使用UTF-8编码。

公共参数

最近更新时间：2018-09-18 16:59:39

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

参数名称	类型	必选	描述
Action	String	是	具体操作的指令接口名称，例如想要调用云服务器的查询实例列表接口，则 Action 参数即为 DescribeInstances。
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	API 的版本。例如 2017-03-12
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

假设用户想要查询广州地域的云服务器实例列表，则其请求链接的形式可能如下：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances
&SecretId=xxxxxxx
&Region=ap-guangzhou
&Timestamp=1402992826
&Nonce=345122
&Signature=xxxxxxx
&Version=2017-03-12
```

接口鉴权

最近更新时间：2018-09-18 16:59:40

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往[云API密钥页面](#)申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往[云API密钥页面](#)申请安全凭证。安全凭证包括 SecretId 和 SecretKey：

SecretId 用于标识 API 调用者身份 SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

- 1) 登录[腾讯云管理中心控制台](#)。
- 2) 前往[云API密钥](#)的控制台页面
- 3) 在[云API密钥](#)页面，点击【新建】即可以创建一对SecretId/SecretKey

开发商帐号最多可以拥有两对SecretId / SecretKey。

2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

```
SecretId: AKIDz8krbsJ5yKBZQpn74WfkmLPx3EXAMPLE SecretKey:  
Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
```

注意：这里只是示例，请根据用户实际申请的SecretId和SecretKey进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户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': 'DescribeInstances',
  'InstanceIds.0': 'ins-09dx96dg',
  'Limit': 20,
  'Nonce': 11886,
  'Offset': 0,
  'Region': 'ap-guangzhou',
  'SecretId': 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
  'Timestamp': 1465185768,
  'Version': '2017-03-12',
}
```

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。

2.2. 拼接请求字符串

此步骤生成请求字符串。 将把上一步排序好的请求参数格式化“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 "DescribeInstances"，因此格式化后就为 Action=DescribeInstances。 **注意：“参数值”为原始值而非url编码后的值。**

然后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=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. 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：

1) 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。 2) 请求主机: 查看实例列表 (DescribeInstances) 的请求域名为：cvm.tencentcloudapi.com。 实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。 3) 请求路径: 当前版本云API的请求路径固定为 /。 4) 请求字符串: 即上一步生成的请求字符串。

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：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。 首先使用 HMAC-SHA1 算法对上一步中获得的**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：

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
EliP9YW3pW28FpsEdkXt/+WcGel=
```

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。

3. 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。**注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则对所有请求参数值均需要做 URL 编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。**如上一步生成的签名串为

EliP9YW3pW28FpsEdkXt/+WcGel=，最终得到的签名串请求参数(Signature)为：

EliP9YW3pW28FpsEdkXt/+WcGel=，它将用于生成最终的请求URL。

注意：有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

4. 鉴权失败

根据实际情况，存在以下鉴权失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.UnauthorizedOperation	请求未 CAM 授权
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.MFAFailure	MFA 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [JavaScript](#)
- [.NET](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt/+WcGel=&Timestamp=1465185768&Version=2017-03-12`

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
```

```
mac.init(secretKeySpec);
byte[] hash = mac.doFinal(s.getBytes(CHARSET));
return DatatypeConverter.printBase64Binary(hash);
}

public static String getStringToSign(TreeMap<String, Object> params) {
    StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
    // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
    for (String k : params.keySet()) {
        s2s.append(k).append("=").append(params.get(k).toString()).append("&");
    }
    return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException
{
    StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
    // 实际请求的url中对参数顺序没有要求
    for (String k : params.keySet()) {
        // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
        url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append
("&");
    }
    return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MA
    X_VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间，例如：params.put("Timestamp", System.currentTimeMillis() / 100
    0);
    params.put("Timestamp", 1465185768); // 公共参数
    params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceId.0", "ins-09dx96dg"); // 业务参数
    params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE",
    "HmacSHA1")); // 公共参数
    System.out.println(getUrl(params));
}
```

```
}  
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包：`pip install requests`。

```
# -*- coding: utf8 -*-  
import base64  
import hashlib  
import hmac  
import time  
  
import requests  
  
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"  
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"  
  
def get_string_to_sign(method, endpoint, params):  
    s = method + endpoint + "?"  
    query_str = "&".join("%s=%s" % (k, data[k]) for k in sorted(data))  
    return s + query_str  
  
def sign_str(key, s, method):  
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()  
    return base64.b64encode(hmac_str)  
  
if __name__ == '__main__':  
    endpoint = "cvm.tencentcloudapi.com"  
    data = {  
        'Action': 'DescribeInstances',  
        'InstanceId.0': 'ins-09dx96dg',  
        'Limit': 20,  
        'Nonce': 11886,  
        'Offset': 0,  
        'Region': 'ap-guangzhou',  
        'SecretId': secret_id,  
        'Timestamp': 1465185768, # int(time.time())  
        'Version': '2017-03-12'  
    }  
    s = get_string_to_sign("GET", endpoint, data)  
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)  
    print(data["Signature"])  
    # 此处会实际调用，成功后可能产生计费
```



```
# resp = requests.get("https://" + endpoint, params=data)
# print(resp.url)
```

返回结果

最近更新时间：2018-09-18 16:59:40

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。

- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误代码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云API密钥类型）
AuthFailure.MFAFailure	MFA错误
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureExpire	签名过期
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token错误
AuthFailure.UnauthorizedOperation	请求未CAM授权
DryRunOperation	DryRun操作，代表请求将会是成功的，只是多传了DryRun参数
FailedOperation	操作失败
InternalError	内部错误
InvalidAction	接口不存在
InvalidParameter	参数错误
InvalidParameterValue	参数取值错误
LimitExceeded	超过配额限制
MissingParameter	缺少参数错误
NoSuchVersion	接口版本不存在

错误代码	错误描述
RequestLimitExceeded	请求的次数超过了频率限制
ResourceInUse	资源被占用
ResourceInsufficient	资源不足
ResourceNotFound	资源不存在
ResourceUnavailable	资源不可用
UnauthorizedOperation	未授权操作
UnknownParameter	未知参数错误
UnsupportedOperation	操作不支持
UnsupportedProtocol	http(s)请求协议错误，只支持GET和POST请求
UnsupportedRegion	接口不支持所传地域

更新历史

最近更新时间：2018-09-14 15:12:05

第 2 次发布

发布时间：2018-08-17 13:00:45

本次发布包含了以下内容：

改善已有的文档。

第 1 次发布

发布时间：2018-07-19 16:17:35

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AcceptDirectConnectTunnel](#)
- [CreateDirectConnectTunnel](#)
- [DeleteDirectConnectTunnel](#)
- [DescribeDirectConnectTunnels](#)
- [ModifyDirectConnectTunnelAttribute](#)
- [RejectDirectConnectTunnel](#)

新增数据结构：

- [BgpPeer](#)
- [DirectConnectTunnel](#)
- [Filter](#)
- [RouteFilterPrefix](#)

专线接入相关接口

接受专用通道申请

最近更新时间：2018-09-18 16:59:44

1. 接口描述

接口请求域名：dc.tencentcloudapi.com。

接受专用通道申请

默认接口请求频率限制：20次/每秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：AcceptDirectConnectTunnel
Version	是	String	公共参数，本接口取值：2018-04-10
Region	否	String	公共参数，本接口不需要传递此参数。
DirectConnectTunnelId	是	String	物理专线拥有者接受共享专用通道申请

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 接受专用通道申请

输入示例

```
https://dc.tencentcloudapi.com/?Action=AcceptDirectConnectTunnel
&DirectConnectTunnelId=dcx-abcdefgh
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03"
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（ SDK ），支持多种编程语言，能更方便的调用 API 。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InvalidParameter	参数错误。
MissingParameter	缺少参数错误
ResourceNotFound	资源不存在
ResourceNotFound.DirectConnectTunnelIdsNotExist	专用通道不存在。
UnsupportedOperation.StateConflict	状态冲突。

创建专用通道

最近更新时间：2018-09-18 16:59:43

1. 接口描述

接口请求域名：dc.tencentcloudapi.com。

用于创建专用通道的接口

默认接口请求频率限制：20次/每秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值： CreateDirectConnectTunnel
Version	是	String	公共参数，本接口取值：2018-04-10
Region	否	String	公共参数，本接口不需要传递此参数。
DirectConnectId	是	String	专线 ID，例如：dc-kd7d06of
DirectConnectTunnelName	是	String	专用通道名称
DirectConnectOwnerAccount	否	String	物理专线 owner，缺省为当前客户（物理 专线 owner） 共享专线时这里需要填写共享专线的开发 商账号 ID
NetworkType	否	String	网络类型，分别为VPC、BMVPC， CCN，默认是VPC VPC：私有网络 BMVPC：黑石网络 CCN：云联网
NetworkRegion	否	String	网络地域
VpcId	否	String	私有网络统一 ID 或者黑石网络统一 ID

参数名称	必选	类型	描述
DirectConnectGatewayId	否	String	专线网关 ID，例如 dcg-d545ddf
Bandwidth	否	Integer	专线带宽，单位：Mbps 默认是物理专线带宽值
RouteType	否	String	BGP：BGP路由 STATIC：静态 默认为 BGP 路由
BgpPeer	否	BgpPeer	BgpPeer，用户侧bgp信息，包括Asn和AuthKey
RouteFilterPrefixes.N	否	Array of RouteFilterPrefix	静态路由，用户IDC的网段地址
Vlan	否	Integer	vlan，范围：0 ~ 3000 0：不开启子接口 默认值是非0
TencentAddress	否	String	TencentAddress，腾讯侧互联 IP
CustomerAddress	否	String	CustomerAddress，用户侧互联 IP

3. 输出参数

参数名称	类型	描述
DirectConnectTunnelIdSet	Array of String	专用通道ID
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 BGP路由模式和私有网络VPC的专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=CreateDirectConnectTunnel
&DirectConnectId=dc-abcdefgh
&DirectConnectTunnelName=Test
&NetworkType=VPC
&NetworkRegion=ap-guangzhou
&VpcId=vpc-abcdefgh
&DirectConnectGatewayId=dcg-abcdefgh
&Bandwidth=100
&RouteType=BGP
&Vlan=100
&TencentAddress=192.168.1.2/30
&CustomerAddress=192.168.1.1/30
&BgpPeer.Asn=65128
&BgpPeer.AuthKey=abcdefg
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "DirectConnectTunnelIdSet": [
      "dcx-abcdefgh"
    ],
    "RequestId": "24a0d7e5-4c13-49be-aa16-94f698fbef3e"
  }
}
```

示例2 STATIC路由模式和黑石VPC的专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=CreateDirectConnectTunnel
&DirectConnectId=dc-abcdefgh
&DirectConnectTunnelName=Test
&NetworkType=BMVPC
&NetworkRegion=ap-guangzhou
&VpcId=vpc-abcdefgh
&Bandwidth=100
&RouteType=STATIC
&Vlan=100
&TencentAddress=192.168.1.2/30
&CustomerAddress=192.168.1.1/30
&RouteFilterPrefixes.0.Cidr=192.168.0.0/24
```

```
&RouteFilterPrefixes.1.Cidr=192.168.0.0/24
&RouteFilterPrefixes.2.Cidr=192.168.0.0/24
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "DirectConnectTunnelIdSet": [
      "dcx-abcdefgh"
    ],
    "RequestId": "24a0d7e5-4c13-49be-aa16-94f698fbef3e"
  }
}
```

示例3 BGP路由模式和共享专线的专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=CreateDirectConnectTunnel
&DirectConnectId=dc-abcdefgh
&DirectConnectTunnelName=Test
&DirectConnectOwnerAccount=240791248
&NetworkType=VPC
&NetworkRegion=ap-guangzhou
&VpcId=vpc-abcdefgh
&DirectConnectGatewayId=dcg-abcdefgh
&Bandwidth=100
&RouteType=BGP
&Vlan=100
&TencentAddress=192.168.1.2/30
&CustomerAddress=192.168.1.1/30
&BgpPeer.Asn=65128
&BgpPeer.AuthKey=abcdefg
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "DirectConnectTunnelIdSet": [
      "dcx-abcdefgh"
    ],

```

```
"RequestId": "24a0d7e5-4c13-49be-aa16-94f698fbef3e"  
}  
}
```

示例4 BGP路由模式和云联网专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=CreateDirectConnectTunnel  
&DirectConnectId=dc-abcdefgh  
&DirectConnectTunnelName=Test  
&NetworkType=CCN  
&NetworkRegion=ap-guangzhou  
&DirectConnectGatewayId=dcg-abcdefgh  
&Bandwidth=100  
&RouteType=BGP  
&Vlan=100  
&TencentAddress=192.168.1.2/30  
&CustomerAddress=192.168.1.1/30  
&BgpPeer.Asn=65128  
&BgpPeer.AuthKey=abcdefg  
&<公共请求参数>
```

输出示例

```
{  
  "Response": {  
    "DirectConnectTunnelIdSet": [  
      "dcx-abcdefgh"  
    ],  
    "RequestId": "24a0d7e5-4c13-49be-aa16-94f698fbef3e"  
  }  
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（ SDK ），支持多种编程语言，能更方便的调用 API 。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InvalidParameter	参数错误。
InvalidParameter.AddressError	互联IP错误。
InvalidParameter.DirectConnectIdIsNotUin	物理专线不属于该账号。
InvalidParameter.UinIsNotExist	该账号ID不存在。
InvalidParameterValue	参数取值错误
InvalidParameterValue.VlanConflict	vlan冲突。
LimitExceeded	超过配额限制
LimitExceeded.DirectConnectLimitExceeded	物理专线数已达上限。
LimitExceeded.DirectConnectTunnelLimitExceeded	物理专线的专用通道数已达上限。
MissingParameter	缺少参数错误
ResourceInUse.DcVpclsExist	物理专线的vpc已经存在。

错误码	描述
ResourceUnavailable.InsufficientBalance	对不起您的帐号已欠费，欠费状态下无法开通产品，请您先行充值。
UnsupportedOperation	操作不支持。

删除专用通道

最近更新时间：2018-09-18 16:59:43

1. 接口描述

接口请求域名：dc.tencentcloudapi.com。

删除专用通道

默认接口请求频率限制：20次/每秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DeleteDirectConnectTunnel
Version	是	String	公共参数，本接口取值：2018-04-10
Region	否	String	公共参数，本接口不需要传递此参数。
DirectConnectTunnelId	是	String	专用通道ID

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 删除专用通道

输入示例


```
https://dc.tencentcloudapi.com/?Action=DeleteDirectConnectTunnel
&DirectConnectTunnelId=dcx-abcdefgh
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03"
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（ SDK ），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InvalidParameter	参数错误。
ResourceNotFound	资源不存在
ResourceNotFound.DirectConnectTunnelIdsNotExist	专用通道不存在。
UnsupportedOperation	操作不支持。
UnsupportedOperation.StateConflict	状态冲突。

查询专用通道列表

最近更新时间：2018-09-18 16:59:42

1. 接口描述

接口请求域名：dc.tencentcloudapi.com。

用于查询专用通道列表。

默认接口请求频率限制：20次/每秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值： DescribeDirectConnectTunnels
Version	是	String	公共参数，本接口取值：2018-04-10
Region	否	String	公共参数，本接口不需要传递此参数。
Filters.N	否	Array of Filter	过滤条件： 参数不支持同时指定DirectConnectTunnelIds和Filters。 <ul style="list-style-type: none">direct-connect-tunnel-name, 专用通道名称。direct-connect-tunnel-id, 专用通道实例ID，如dcx-abcdefgh。direct-connect-id, 物理专线实例ID，如，dc-abcdefgh。
DirectConnectTunnelIds.N	否	Array of String	专用通道 ID数组
Offset	否	Integer	偏移量，默认为0
Limit	否	Integer	返回数量，默认为20，最大值为100

3. 输出参数

参数名称	类型	描述
DirectConnectTunnelSet	Array of DirectConnectTunnel	专用通道列表
TotalCount	Integer	符合专用通道数量。
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 查询BGP路由的专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=DescribeDirectConnectTunnels
&Filters.0.Name=direct-connect-tunnel-id
&Filters.0.Values.0=dcx-r3sml04o
&Limit=20
&Offset=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "DirectConnectTunnelSet": [
      {
        "BgpPeer": {
          "Asn": 65139,
          "AuthKey": "tencent"
        },
        "CreatedTime": "2018-06-01 14:59:16",
        "CustomerAddress": "169.254.64.2/30",
        "DirectConnectGatewayId": "d cg-r70hz833",
        "DirectConnectId": "dc-9s5kpgyp",
        "DirectConnectOwnerAccount": "2407912486",
        "DirectConnectTunnelId": "dcx-r3sml04o",
        "DirectConnectTunnelName": "测试专用通道",
        "NetworkRegion": "ap-guangzhou",
        "NetworkType": "VPC",
        "OwnerAccount": "2407912486",
```

```
"RouteFilterPrefixes": [],
"RouteType": "BGP",
"State": "PENDING",
"TencentAddress": "169.254.64.1/30",
"Vlan": 1321,
"Vpclid": "vpc-aipqhdez"
}
]
}
}
```

示例2 查询STATIC路由的专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=DescribeDirectConnectTunnels
&Filters.0.Name=direct-connect-tunnel-id
&Filters.0.Values.0=dcx-r3sml04o
&Limit=20
&Offset=2
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "DirectConnectTunnelSet": [
      {
        "BgpPeer": {
          "Asn": -1,
          "AuthKey": ""
        },
        "CreatedTime": "2018-06-01 14:59:16",
        "CustomerAddress": "169.254.64.2/30",
        "DirectConnectGatewayId": "dcg-r70hz833",
        "DirectConnectId": "dc-9s5kpgyp",
        "DirectConnectOwnerAccount": "2407912486",
        "DirectConnectTunnelId": "dcx-r3sml04o",
        "DirectConnectTunnelName": "测试专用通道",
        "NetworkRegion": "ap-guangzhou",
        "NetworkType": "VPC",
        "OwnerAccount": "2407912486",
        "RouteFilterPrefixes": [
          {
```

```
"Cidr": "172.18.27.6/32"
},
{
  "Cidr": "172.18.27.0/24"
}
],
"RouteType": "STATIC",
"State": "PENDING",
"TencentAddress": "169.254.64.1/30",
"Vlan": 1321,
"VpcId": "vpc-aipqhdez"
}
]
}
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
ResourceNotFound	资源不存在
ResourceNotFound.DirectConnectTunnelIdsNotExist	专用通道不存在。

修改专用通道属性

最近更新时间：2018-09-18 16:59:42

1. 接口描述

接口请求域名：dc.tencentcloudapi.com。

修改专用通道属性

默认接口请求频率限制：20次/每秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值： ModifyDirectConnectTunnelAttribute
Version	是	String	公共参数，本接口取值：2018-04-10
Region	否	String	公共参数，本接口不需要传递此参数。
DirectConnectTunnelId	是	String	专用通道ID
DirectConnectTunnelName	否	String	专用通道名称
BgpPeer	否	BgpPeer	用户侧BGP，包括Asn，AuthKey
RouteFilterPrefixes.N	否	Array of RouteFilterPrefix	用户侧网段地址
TencentAddress	否	String	腾讯侧互联IP
CustomerAddress	否	String	用户侧互联IP
Bandwidth	否	Integer	专用通道带宽值，单位为M。

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 修改路由模式是BGP的专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=ModifyDirectConnectTunnelAttribute
&DirectConnectTunnelId=dcx-abcdefgh
&DirectConnectTunnelName=Test
&Bandwidth=100
&TencentAddress=192.168.1.1/30
&CustomerAddress=192.168.1.2/30
&BgpPeer.Asn=65128
&BgpPeer.AuthKey=abcdefg
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03"
  }
}
```

示例2 修改路由模式是STATIC的专用通道

输入示例

```
https://dc.tencentcloudapi.com/?Action=ModifyDirectConnectTunnelAttribute
&DirectConnectTunnelId=dcx-abcdefgh
&DirectConnectTunnelName=Test
&Bandwidth=100
&TencentAddress=192.168.1.1/30
&CustomerAddress=192.168.1.2/30
&RouteFilterPrefixes.0.Cidr=192.168.0.0/24
&RouteFilterPrefixes.1.Cidr=192.168.1.0/24
&RouteFilterPrefixes.2.Cidr=192.168.2.0/24
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03"
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
FailedOperation	操作失败

错误码	描述
InternalServerError	内部错误
InvalidParameter	参数错误。
MissingParameter	缺少参数错误
ResourceNotFound.DirectConnectTunnelIdsNotExist	专用通道不存在。
UnsupportedOperation.StateConflict	状态冲突。

拒绝专用通道申请

最近更新时间：2018-09-18 16:59:41

1. 接口描述

接口请求域名：dc.tencentcloudapi.com。

拒绝专用通道申请

默认接口请求频率限制：20次/每秒。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见[公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：RejectDirectConnectTunnel
Version	是	String	公共参数，本接口取值：2018-04-10
Region	否	String	公共参数，本接口不需要传递此参数。
DirectConnectTunnelId	是	String	无

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求ID，每次请求都会返回。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 拒绝专用通道申请

输入示例

```
https://dc.tencentcloudapi.com/?Action=RejectDirectConnectTunnel
&DirectConnectTunnelId=dcx-abcdefgh
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03"
  }
}
```

5. 开发者资源

API Explorer

该工具提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力，能显著降低使用云 API 的难度，推荐使用。

- [API 3.0 Explorer](#)

SDK

云 API 3.0 提供了配套的开发工具集（ SDK ），支持多种编程语言，能更方便的调用 API 。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见[公共错误码](#)。

错误码	描述
InternalServerError	内部错误
InvalidParameter	参数错误。
ResourceNotFound.DirectConnectTunnelIdsNotExist	专用通道不存在。
UnauthorizedOperation	未授权操作
UnsupportedOperation.StateConflict	状态冲突。

数据结构

最近更新时间：2018-09-14 15:12:09

BgpPeer

bgp参数，包括Asn，AuthKey

被如下接口引用：CreateDirectConnectTunnel, DescribeDirectConnectTunnels, ModifyDirectConnectTunnelAttribute.

名称	类型	必选	描述
Asn	Integer	否	用户侧，BGP Asn
AuthKey	String	否	用户侧BGP密钥

DirectConnectTunnel

专线通道信息列表

被如下接口引用：DescribeDirectConnectTunnels.

名称	类型	描述
DirectConnectTunnelId	String	专线通道ID
DirectConnectId	String	物理专线ID
State	String	专线通道状态 AVAILABLE:就绪或者已连接 PENDING:申请中 ALLOCATING:配置中 ALLOCATED:配置完成 ALTERING:修改中 DELETING:删除中 DELETED:删除完成 CONFIRMING:待接受 REJECTED:拒绝
DirectConnectOwnerAccount	String	物理专线的拥有者，开发商账号 ID

名称	类型	描述
OwnerAccount	String	专线通道的拥有者，开发商账号 ID
NetworkType	String	网络类型，分别为VPC、BMVPC、CCN VPC：私有网络，BMVPC：黑石网络， CCN：云联网
NetworkRegion	String	VPC地域
Vpclid	String	私有网络统一 ID 或者黑石网络统一 ID
DirectConnectGatewayId	String	专线网关 ID
RouteType	String	BGP：BGP路由 STATIC：静态 默认为 BGP 路由
BgpPeer	BgpPeer	用户侧BGP，Asn，AuthKey
RouteFilterPrefixes	Array of RouteFilterPrefix	用户侧网段地址
Vlan	Integer	专线通道的Vlan
TencentAddress	String	TencentAddress，腾讯侧互联 IP
CustomerAddress	String	CustomerAddress，用户侧互联 IP
DirectConnectTunnelName	String	专线通道名称
CreatedTime	Timestamp	专线通道创建时间
Bandwidth	Integer	专线通道带宽值

Filter

用于条件过滤查询

被如下接口引用：DescribeDirectConnectTunnels.

名称	类型	必选	描述
Name	String	是	需要过滤的字段。
Values	Array of String	是	字段的过滤值。

RouteFilterPrefix

用户侧网段地址

被如下接口引用：CreateDirectConnectTunnel, DescribeDirectConnectTunnels, ModifyDirectConnectTunnelAttribute.

名称	类型	必选	描述
Cidr	String	否	用户侧网段地址

专线接入 API 2017

简介

最近更新时间：2018-06-25 14:47:58

欢迎使用腾讯云专线接入(DC)。本文档提供的API供您使用请求调用的方式来操作专线网关及专线通道。请确保在使用这些接口前，已充分了解DC产品、及其使用和收费方式。

在本文档的接口说明部分，凡出现任何参数可选范围等方面与腾讯云官网上给出的数值发生矛盾时，**均以官网上给出的值为准。**

API概览

最近更新时间：2018-06-25 14:26:17

接口功能	Action ID
查询物理专线列表	DescribeDirectConnects
创建专线通道	CreateDirectConnectTunnel
查询专线通道列表	DescribeDirectConnectTunnels

物理专线

查询专线列表

最近更新时间：2018-06-25 17:52:55

功能描述

DescribeDirectConnects 用于查询专线列表。

接口请求域名：`dc.api.qcloud.com`

请求

语法示例：

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DescribeDirectConnects
&<公共请求参数>
&directConnectId=dc-kd7d06of
```

请求参数

以下请求参数列表仅列出了接口请求参数，正式调用时需要加上公共请求参数，见[公共请求参数](#)页面。其中，此接口的Action字段为DescribeDirectConnects。

参数名称	是否必选	类型	描述
directConnectId	否	String	专线 ID，例如：dc-kd7d06of，不传返回开发商创建的所有专线。

响应

响应示例：

```
{
  "code": 0,
  "message": "",
  "data": [
    {
```

```
}  
]  
}
```

响应参数

参数名称	类型	描述
code	Int	错误码，0: 成功，其他值: 失败。
message	String	错误信息。
data.n	Array	返回的数组。
data.n.directConnectId	String	系统分配的专线ID，例如：dc-kd7d06of。
data.n.directConnectName	String	专线名称。
data.n.status	String	专线状态，0：运行中；1：已到期；2：删除中；3：已删除；11：申请中；12：申请驳回；13：待付款；14：已付款；15：建设中；16：施工停止；
data.n.provider	String	专线运营商。
data.n.portType	int	接口类型。1:100Base-T百兆电口；2: 1000Base-T 千兆电口；3: 1000Base-LX 千兆单模光口（10km）；4: 10GBase-LR 万兆单模光口（10km）
data.n.accessPoints	String	接入点。
data.n.bandwidth	String	专线带宽，单位Mbps。
data.n.loaIssueTime	String	专线到期时间。

实际案例

请求

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DescribeDirectConnects  
&<<a href="https://cloud.tencent.com/doc/api/229/6976">公共请求参数</a>>  
&directConnectId=dc-kd7d06of
```

响应

```
{
  "code": 0,
  "message": "",
  "data": [
    {
      "directConnectId": "dc-3cavza1z",
      "directConnectName": "李昊测试专线@123112",
      "status": 15,
      "provider": "中国电信",
      "portType": 8,
      "accessPoints": "1",
      "bandwidth": 200,
      "loalssueTime": "2017-09-02 15:41:00"
    }
  ]
}
```

专线通道

创建专线通道

最近更新时间：2018-06-25 14:27:12

接口描述

用于创建专线通道的接口。

域名：dc.api.qcloud.com

接口名：CreateDirectConnectTunnel

请求参数

以下请求参数列表仅列出了接口请求参数，正式调用时需要加上公共请求参数，详情请参见 [公共请求参数](#) 页面。其中，此接口的 Action 字段为 CreateDirectConnectTunnel。

参数	必选	类型	描述
directConnectId	是	String	专线 ID，例如：dc-kd7d06of
directConnectTunnelName	是	String	专线通道名称
ownerAccount	否	String	物理专线 owner，缺省为当前客户（物理专线 owner） 共享专线时这里需要填写共享专线的开发商账号 ID
networkType	否	Int	网络类型，默认为 1 1：私有网络 0：黑石网络
region	是	String	网络地域
vpclId	是	String	私有网络统一 ID 或者黑石网络统一 ID
directConnectGatewayId	是	String	专线网关 ID，例如 dcg-d545ddf
bandwidth	否	Int	专线带宽，单位：Mbps 0：不限制
routeMode	否	Int	0：BGP 路由 1：静态 默认为 BGP 路由

参数	必选	类型	描述
bgpPeers.asn	否	String	BGP asn
bgpPeers.authKey	否	String	BGP 密钥
routeFilterPrefixes.n.cidr	否	String	对端网段
vlanId	是	Int	vlanId, 范围: 0 ~ 3000 0: 不开启子接口
localGatewayIp	否	String	localGatewayIp, 腾讯侧互联 IP
peerGatewayIp	否	String	peerGatewayIp, 用户侧互联 IP
peeringSubnetMask	否	String	互联 IP 的掩码, 必须定义在同一子网内, 支持 24 ~ 30 位, 点分十进制, 如 255.255.255.252
remark	否	String	备注

响应参数

参数	类型	描述
code	Int	错误码 0: 成功 其他值: 失败
message	String	错误信息
directConnectTunnelId	String	专线通道ID

代码示例

请求示例

```
GET https://dc.api.qcloud.com/v2/index.php?Action=CreateDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">公共请求参数</a>>
&directConnectId=dc-kd7d06of
&directConnectTunnelName=baytest
&region=gz
&vpclId=vpc-abcdefg
```



```
&directConnectGatewayId=dcg-abcdefg
&vlanId=400
&routeMode=1
&routeFilterPrefixes.0.cidr=172.256.12.0/24
&routeFilterPrefixes.1.cidr=172.256.13.0/24
&localGatewayIp=169.254.64.1
&peerGatewayIp=169.254.64.2
&peeringSubnetMask=255.255.255.252
&remark=create
```

响应示例

```
{
  "code": 0,
  "message": ""
  "directConnectTunnelId":""
}
```

修改专线通道

最近更新时间：2018-06-25 15:04:51

功能描述

用于修改专线通道参数的接口。

域名：`dc.api.qcloud.com`

接口名：`ModifyDirectConnectTunnel`

请求参数

参数	类型	必选	描述
<code>directConnectTunnelId</code>	String	是	专线通道 ID 例如： <code>dcx-abcdefgh</code>
<code>directConnectTunnelName</code>	String	否	专线通道名称
<code>peerAsn</code>	Int	否	用户侧，BGP asn
<code>authKey</code>	String	否	用户侧，BGP 密钥
<code>routeFilterPrefixes.n.cidr</code>	String	否	用户侧，对端网段 如 <code>169.254.0.0/28</code>
<code>localGatewayIp</code>	String	否	<code>localGatewayIp</code> ，腾讯侧互联 IP
<code>peerGatewayIp</code>	String	否	<code>peerGatewayIp</code> ，用户侧互联 IP
<code>peeringSubnetMask</code>	String	否	互联 IP 的掩码，必须定义在同一子网内，支持 24 ~ 30 位，点分十进制，如 <code>255.255.255.252</code>

响应参数

参数	类型	描述
<code>code</code>	Int	错误码 0：成功 其他值：失败

参数	类型	描述
message	String	错误信息

代码示例

请求示例

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DeleteDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">公共请求参数</a>>
&directConnectTunnelId=dcx-abcdefgh
&bandwidth=10
&routeFilterPrefixes.0.cidr=172.256.12.0/24
&routeFilterPrefixes.1.cidr=172.256.13.0/24
&localGatewayIp=169.254.64.1
&peerGatewayIp=169.254.64.2
&peeringSubnetMask=255.255.255.252
```

响应示例

```
{
  "code": 0,
  "message": ""
}
```

删除专线通道

最近更新时间：2018-06-25 15:05:39

功能描述

用于删除专线通道的接口。

域名：`dc.api.qcloud.com`

接口名：`DeleteDirectConnectTunnel`

请求参数

参数	类型	必选	描述
<code>directConnectTunnelId</code>	String	是	专线通道 ID 例如： <code>dcx-abcdefgh</code>

响应参数

参数	类型	描述
<code>code</code>	Int	错误码 0：成功 其他值：失败
<code>message</code>	String	错误信息

代码示例

请求示例

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DeleteDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">公共请求参数</a>>
&directConnectTunnelId=dcx-abcdefgh
```

响应示例

```
{  
  "code": 0,  
  "message": ""  
}
```

查询专线通道列表

最近更新时间：2018-06-28 15:22:36

功能描述

用于查询专线通道列表。

域名：`dc.api.qcloud.com`

接口名：`DescribeDirectConnectTunnels`

请求参数

以下请求参数列表仅列出了接口请求参数，正式调用时需要加上公共请求参数，详情请参见 [公共请求参数](#) 页面。其中，此接口的 Action 字段为 `DescribeDirectConnectTunnels`。

参数	必选	类型	描述
<code>directConnectId</code>	否	String	专线 ID 例如： <code>dc-kd7d06of</code>
<code>directConnectTunnelId</code>	否	String	专线通道 ID 例如： <code>dcx-kd7d0125</code>

响应参数

参数	类型	描述
<code>code</code>	Int	错误码 0：成功 其他值：失败
<code>message</code>	String	错误信息
<code>data.n</code>	Array	返回的数组
<code>data.n.directConnectTunnelId</code>	String	系统分配的专线通道 ID 例如： <code>dcx-kd7d0125</code>
<code>data.n.directConnectTunnelName</code>	String	专线通道名称

参数	类型	描述
data.n.directConnectId	String	系统分配的专线 ID 例如：dc-kd7d06of
data.n.ownerAccount	String	专线的开发商账号 ID
data.n.networkType	Int	0：黑石网络 1：私有网络
data.n.region	String	网络地域
data.n.vpcId	String	私有网络统一 ID 或者黑石网络统一 ID
data.n.directConnectGatewayId	String	专线网关 ID 例如：dcg-d545ddf
data.n.bandwidth	Int	专线带宽，单位：Mbps
data.n.routeMode	Int	0：BGP 路由 1：静态 默认为 BGP 路由
data.n.bgpPeers.asn	string	BGP asn
data.n.bgpPeers.authKey	String	BGP 密钥
data.n.routeFilterPrefixes.n.cidr	String	对端网段
data.n.status	Int	专线通道状态 0：已连接 1：申请中 2：配置中 6：配置完成 20：等待连接 21：拒绝
data.n.vlan	Int	vlan Id
data.n.localGatewayIp	String	腾讯侧互联 IP
data.n.peerGatewayIp	String	用户侧互联 IP
data.n.peeringSubnetMask	String	互联 IP 的掩码 例如：255.255.255.252
data.n.remark	String	备注

代码示例

请求示例

```
GET https://dc.api.qcloud.com/v2/index.php?Action=DescribeDirectConnectTunnels
&<<a href="https://cloud.tencent.com/doc/api/229/6976">公共请求参数</a>>
&directConnectId=dc-kd7d06of
```

响应示例

```
{
  "code": 0,
  "message": "",
  "data": [
    {
      "directConnectTunnelId": "dcx-2nakhj58",
      "directConnectTunnelName": "barrytest2",
      "directConnectId": "dc-5e8ak079",
      "networkType": 1,
      "ownerAccount": "",
      "region": "gz",
      "vpcId": "vpc-kx49lmyv",
      "bandwidth": 0,
      "routeMode": 0,
      "bgpPeers": {
        "asn": "10",
        "authKey": "124545d"
      },
      "routeFilterPrefixes": [
        {
          "cidr": ""
        }
      ],
      "status": 1,
      "vlan": 0,
      "localGatewayIp": "169.254.64.1",
      "peerGatewayIp": "169.254.64.2",
      "peeringSubnetMask": "255.255.255.252",
      "remark": ""
    }
  ]
}
```


接受专线通道申请

最近更新时间：2018-06-25 15:07:05

功能描述

用于接受专线通道的接口。

域名：`dc.api.qcloud.com`

接口名：`AcceptDirectConnectTunnel`

请求参数

参数	类型	必选	描述
<code>directConnectTunnelId</code>	String	是	专线通道 ID 例如： <code>dcx-abcdefgh</code>

响应参数

参数	类型	描述
<code>code</code>	Int	错误码 0：成功 其他值：失败
<code>message</code>	String	错误信息

代码示例

请求示例

```
GET https://dc.api.qcloud.com/v2/index.php?Action=AcceptDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">公共请求参数</a>>
&directConnectTunnelId=dcx-abcdefgh
```

响应示例

```
{  
  "code": 0,  
  "message": ""  
}
```

拒绝专线通道申请

最近更新时间：2018-06-25 17:52:18

功能描述

用于拒绝专线通道的接口。

域名：`dc.api.qcloud.com`

接口名：`RefuseDirectConnectTunnel`

请求参数

参数	类型	必选	描述
<code>directConnectTunnelId</code>	String	是	专线通道 ID 例如： <code>dcx-abcdefgh</code>

响应参数

参数	类型	描述
<code>code</code>	Int	错误码 0：成功 其他值：失败
<code>message</code>	String	错误信息

代码示例

请求示例

```
GET https://dc.api.qcloud.com/v2/index.php?Action=RefuseDirectConnectTunnel
&<<a href="https://cloud.tencent.com/doc/api/229/6976">公共请求参数</a>>
&directConnectTunnelId=dcx-abcdefgh
```

响应示例

```
{  
  "code": 0,  
  "message": ""  
}
```

调用方式

请求结构

最近更新时间：2018-07-03 11:53:35

对腾讯云的 API 接口的调用是通过向腾讯云 API 的服务端地址发送请求，并按照接口说明在请求中加入相应的请求参数来完成的。腾讯云 API 的请求结构由：服务地址，通信协议，请求方法，请求参数和字符编码组成。具体描述如下：

服务地址

腾讯云 API 的服务接入地址与具体模块相关，详细请参见各接口相关描述。

通信协议

腾讯云 API 的大部分接口都通过 HTTPS 进行通信，为您提供高安全性的通信通道。

请求方法

腾讯云 API 同时支持 POST 和 GET 请求。

注意：

1. POST 和 GET 请求不能混合使用，若使用 GET 方式，则参数均从 Querystring 取得；若使用 POST 方式，则参数均从 Request Body 中取得，而 Querystring 中的参数将忽略。两种请求方式的参数格式规则相同，一般情况下使用 GET 请求，当参数字符串过长时推荐使用 POST。
2. 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，若为 POST，则无需对参数编码。
3. GET 请求的最大长度根据不同的浏览器和服务器设置有所不同，比如，传统 IE 浏览器限制为 2K，Firefox 限制为 8K；对于一些参数较多、长度较长的 API 请求，建议您使用 POST 方法以免在请求过程中会由于字符串超过最大长度而导致请求失败。
4. 对于 POST 请求，您需要使用 `x-www-form-urlencoded` 的形式传参，因为云 API 侧是从 `$_POST` 中取出请求参数的。

请求参数

腾讯云 API 的每个请求都需要指定两类参数：公共请求参数以及接口请求参数。其中公共请求参数是每个接口都要用到的请求参数，具体可参见 [公共请求参数](#)，而接口请求参数是各个接口所特有的，具体见各个接口的“请求参数”描述。

字符编码

腾讯云 API 的请求及返回结果均使用 UTF-8 字符集进行编码。

接口鉴权

最近更新时间：2018-07-06 17:07:44

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 SecretId 和 SecretKey，若用户还没有安全凭证，则需要在腾讯云官网上自主申请，否则无法调用云 API 接口。

申请安全凭证

在第一次使用腾讯云 API 之前，用户需要在【腾讯云控制台】>【[API 密钥管理](#)】上申请安全凭证。安全凭证包括 SecretId 和 SecretKey，其中：

- **SecretId**：用于标识 API 调用者身份；
- **SecretKey**：用于加密签名字符串和服务器端验证签名字符串的密钥。

注意：

API 密钥是构建腾讯云 API 请求的重要凭证，使用腾讯云 API 可以操作您名下的所有腾讯云资源，为了您的财产和服务安全，请妥善保存并定期更换密钥，当您更换密钥后，请及时删除旧密钥。

申请安全凭证步骤：

1. 登录 [腾讯云控制台](#)。

2. 单击【云产品】，选择【管理工具】栏下的【云 API 密钥】，进入云 API 密钥管理页面。



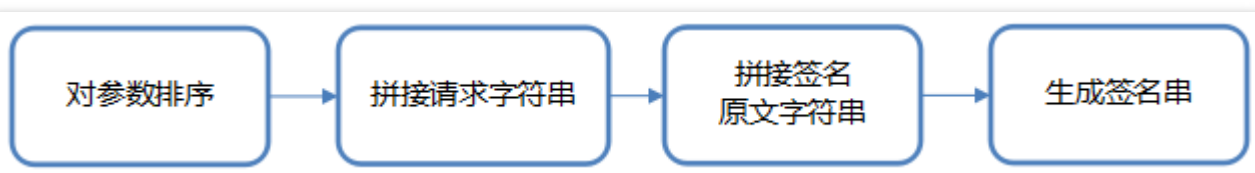
3. 在 [API 密钥管理](#) 页面，单击【新建密钥】即可以创建一对 SecretId/SecretKey。

注意：

- 开发商帐号最多可以拥有两对 SecretId / SecretKey。
- 被开发商添加为子用户的 QQ 帐号，在不同开发商控制台，可以申请不同的安全凭证。
- 子用户的安全凭证，目前仅可调用部分接口的云 API。

生成签名串

有了安全凭证 SecretId 和 SecretKey 后，就可以生成签名串了。生成签名串的详细过程如下：



假设用户的 SecretId 和 SecretKey 分别是：

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 和请求参数进行后续操作。

以腾讯云 CVM 为例，当用户调用腾讯云 CVM 的 [查看实例列表](#) (DescribeInstances)接口时，其请求参数为：

参数名称	描述	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
SignatureMethod	签名方式	HmacSHA256
InstanceIds.0	待查询的实例 ID	ins-09dx96dg

1. 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": "DescribeInstances",
  "InstanceIds.0": "ins-09dx96dg",
  "Nonce": 11886,
  "Region": "ap-guangzhou",
  "SecretId": "AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA",
  "SignatureMethod": "HmacSHA256",
  "Timestamp": 1465185768,
}
```

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。

2. 拼接请求字符串

此步骤将生成请求字符串。

将把上一步排序好的请求参数格式化成为 “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 "DescribeInstances"，因此格式化后就为 Action=DescribeInstances。

注意：

1. “参数值” 为原始值而非 url 编码后的值。
2. 若输入参数的 Key 中包含下划线，则需要将其转换为“.”，但是 Value 中的下划线则不用转换。如 Placement_Zone=CN_GUANGZHOU，则需要将其转换成 Placement.Zone=CN_GUANGZHOU。

然后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（请忽略文中的换行）：

```
Action=DescribeInstances
&InstanceId=ins-09dx96dg
&Nonce=11886
&Region=ap-guangzhou
&SecretId=AKIDz8krbsJ5yKBZQpn74WfkmLPx3gnPhESA
&SignatureMethod=HmacSHA256
&Timestamp=1465185768
```

3. 拼接签名原文字符串

此步骤将生成签名原文字符串。

签名原文字符串的拼接规则为：

请求方法 + 请求主机 + 请求路径 + ? + 请求字符串

参数构成说明：

- **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**：即主机域名，请求域名由接口所属的产品或所属产品的模块决定，不同产品或不同产品的模块的请求域名会有不同。如腾讯云 CVM 的查询实例列表（DescribeInstances）的请求域名为：cvm.api.qcloud.com，具体产品请求域名详见各接口说明。
- **请求路径**：腾讯云 API 对应产品的请求路径，一般是一个产品对应一个固定路径，如腾讯云 CVM 请求路径固定为 /v2/index.php。
- **请求字符串**：即上一步生成的请求字符串。

因此，上述示例的拼接签名原文字符串结果为（请忽略文中的换行）：

```
GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances
&InstanceId=ins-09dx96dg
&Nonce=11886
&Region=ap-guangzhou
&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA
&SignatureMethod=HmacSHA256
&Timestamp=1465185768
```

4. 生成签名串

此步骤生成签名串。

注意：

计算签名的方法有两种：HmacSHA256 和 HmacSHA1 这里要根据您指定的签名算法（即 SignatureMethod 参数）生成签名串。当指定 SignatureMethod 为 HmacSHA256 时，需要使用 HmacSHA256 计算签名，其他情况请使用 HmacSHA1 计算签名。

首先使用签名算法（HmacSHA256 或 HmacSHA1）对上一步中获得的 **签名原字符串** 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（使用其它程序设计语言开发时，可用上述示例中的原字符串进行签名验证，得到的签名串与例子中的一致即可）：

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3Cozk1qA';
$srcStr = 'GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&InstanceId=ins-09dx96dg&Nonce=11886&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&SignatureMethod=HmacSHA256&Timestamp=1465185768';
$signStr = base64_encode(hash_hmac('sha256', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
0EEEm/HtGRr/VJXTAD9tYMth1Bzm3lLHz5RCDv1GdM8s=
```

同理，当您指定签名算法为 **HmacSHA1** 时，生成签名串的代码如下：

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3Cozk1qA';
$srcStr = 'GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&InstanceId=ins-09dx96dg&Nonce=11886&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3gnPhESA&SignatureMethod=HmacSHA1&Timestamp=1465185768';
```

```
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));  
echo $signStr;
```

最终得到的签名串为：

```
nPVnY6njQmwQ8ciqbPI5Qe+Oru4=
```

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 `0EEem/HtGRr/VJXTAD9tYMth1Bzm3lLHz5RCDv1GdM8s=`，则其编码后为 `0EEem/HtGRr/VJXTAD9tYMth1Bzm3lLHz5RCDv1GdM8s=`。因此，最终得到的签名串请求参数 (Signature) 为：`0EEem/HtGRr/VJXTAD9tYMth1Bzm3lLHz5RCDv1GdM8s=`，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，则对所有请求参数的参数值均需要做 URL 编码；此外，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，可能出现如下表的错误：

错误代码	错误类型	错误描述
4100	身份认证失败	身份验证失败，请确保您请求参数中的 Signature 按照上述步骤计算正确，特别注意 Signature 要做 url 编码后再发起请求。
4101	未被开发商授权访问本接口	该子用户未被授权调用此接口。请联系开发商授权，详情请查阅 授权策略 。
4102	未被开发商授权访问本接口中所操作的资源	请问的资源参数中，存在未被开发商授权访问的资源，请在 message 字段中查看无权查看的资源 ID。请联系开发商授权，详情请查阅 授权策略 。
4103	非开发商的 SecretId 暂不支持调用本接口	子用户的 SecretId 不支持调用此接口，只有开发商有权调用。
4104	SecretId 不存在	签名所用的 SecretId 不存在，也可能是密钥状态有误，请确保 API 密钥有效且未被禁用。

错误代码	错误类型	错误描述
4110	鉴权失败	权限校验失败，请确保您有使用所访问资源的权限。
4500	重放攻击错误	请注意 Nonce 参数两次请求不能重复，Timestamp 与腾讯服务器相差不能超过 2 小时。

公共参数

最近更新时间：2018-08-14 18:08:40

一个完整的腾讯云 API 请求需要两类请求参数：公共请求参数和接口请求参数。本文将介绍腾讯云 API 请求需要用到的 6 个公共请求参数，有关接口请求参数的详细说明请参见 [接口请求参数](#) 章节。

公共请求参数是每个接口都需要使用到的请求参数，开发者每次使用腾讯云 API 发送请求时都需要携带这些公共请求参数，否则会导致请求失败。公共请求参数的首字母均为大写，以此区分于接口请求参数。

公共请求参数具体列表如下：

注意：

文章中的接口实例以腾讯云 CVM 为例，各腾讯云产品的具体使用方式请对应实际产品。

参数名称	描述	类型	必选
Action	具体操作的指令接口名称，例如腾讯云 CVM 用户调用 查询实例列表 接口，则 Action 参数即为 DescribeInstances。	String	是
Region	地域参数，用来标识希望操作哪个地域的实例。详细信息可参见 地域和可用区 列表，或使用 查询地域列表 API 接口查看。 注意： 1. 正常情况下此参数是必须的，如无需传入，则会在相应接口中进行说明。 2. 部分区域正在内测中，目前仅面向部分用户开放。	String	否
Timestamp	当前 UNIX 时间戳，可记录发起 API 请求的时间。	UInt	是
Nonce	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。	UInt	是
SecretId	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。具体可参考 签名方法 章节。	String	是
Signature	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。计算方法可参考 签名方法 章节。	String	是
SignatureMethod	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。详细签名计算方法可参考 签名方法 章节。	String	否
Token	临时证书所用的 Token，需要结合临时密钥一起使用。长期密钥不需要 Token。	String	否

使用示例

腾讯各云产品 API 请求链接中，公共请求参数的形式如下，以腾讯云 CVM 为例，假设用户想要查询广州地域的云服务器实例列表，则其请求链接的形式为：

```
https://cvm.api.qcloud.com/v2/index.php?  
Action=DescribeInstances  
&SecretId=xxxxxxx  
&Region=ap-guangzhou  
&Timestamp=1465055529  
&Nonce=59485  
&Signature=mysignature  
&SignatureMethod=HmacSHA256  
&<接口请求参数>
```

返回值

返回值结构

最近更新时间：2018-06-25 14:51:42

如无特别说明, 每次请求的返回值中, 都会包含下面的字段：

名称	类型	描述
code	Int	返回结果的错误码, 0表示成功, 其它值表示失败。具体错误码的含义可以参考 错误码 页面
message	String	请求结果

例如：

使用公共参数部分的示例请求：

```
https://domain/v2/index.php?Action=DescribeInstances&SecretId=xxxxxxx&Region=gz
&Timestamp=1402992826&Nonce=345122&Signature=mysignature&instanceId=101
```

可能的返回结果如下：

```
{
  "code":0,
  "message": "success",
  "instanceSet":
  [{
    "instanceId":"qcvm1234",
    "cpu":1,
    "mem":2,
    "disk":20,
    "bandwidth":65535,
    "os":"centos_62_64",
    "lanIp":"10.207.248.186",
    "wanIp":null,
    "status":0
  }]
}
```


错误码

最近更新时间：2018-06-25 14:52:26

响应包体内的返回码(code), 反映了腾讯云API调用和执行的概要结果。

当返回码不为 0 时, 表示请求未正常执行, 返回码也称为错误码, 错误描述(message)对该结果进行了细化补充, 用户可根据错误码判断API的执行情况。

message在部分终端(例如浏览器), 中文会显示成unicode编码, 需要解码。

腾讯云 API 可能返回的错误码表如下：

错误代码	错误类型	描述
4000	请求参数非法	缺少必要参数, 或者参数值格式不正确, 具体错误信息请查看错误描述 message 字段。
4100	鉴权失败	签名鉴权失败, 请参考文档中鉴权部分。
4200	请求过期	请求已经过期, 请参考文档中请求有效期部分。
4300	拒绝访问	帐号被封禁, 或者不在接口针对的用户范围内等。
4400	超过配额	请求的次数超过了配额限制, 请参考文档请求配额部分。
4500	重放攻击	请求的 Nonce 和 Timestamp 参数用于确保每次请求只会在服务器端被执行一次, 所以本次的 Nonce 和上次的不能重复, Timestamp 与腾讯服务器相差不能超过 2 小时。
4600	协议不支持	协议不支持, 请参考文档说明。
5000	资源不存在	资源标识对应的实例不存在, 或者实例已经被退还, 或者访问了其他用户的资源。
5100	资源操作失败	对资源的操作失败, 具体错误信息请查看错误描述 message 字段, 稍后重试或者联系客服人员帮忙解决。
5200	资源购买失败	购买资源失败, 可能是不支持实例配置, 资源不足等等。
5300	资源购买失败	购买资源失败, 余额不足。
5400	部分执行成功	批量操作部分执行成功, 详情见方法返回值。
5500	用户资质	购买资源失败, 用户资质审核未通过。

	审核未通过	
6000	服务器内部错误	服务器内部出现错误，请稍后重试或者联系客服人员帮忙解决。
6100	版本暂不支持	本版本内不支持此接口或该接口处于维护状态等。注意: 出现这个错误时, 请先确定接口的域名是否正确, 不同的模块, 域名可能不一样。
6200	接口暂时无法访问	当前接口处于停服维护状态，请稍后重试。

异步任务接口返回格式

最近更新时间：2018-06-25 14:50:47

1. 普通异步任务接口返回格式

一次请求只能操作一个资源的异步任务接口，例如创建负载均衡，重置主机操作系统等。

名称	类型	描述	必选
code	Int	返回结果的错误码，0表示成功，其它值表示失败。	Yes
message	String	返回结果的错误信息	No
requestId	String	任务编号	Yes

2. 批量异步任务接口返回格式

一次请求能操作多个资源的异步任务接口，例如修改密码，启动机器，停止机器等。

名称	类型	描述	必选
code	Int	返回结果的错误码，0表示成功，其它值表示失败。	Yes
message	String	返回结果的错误信息	No
detail	Array	以资源ID为key, 返回对该资源操作的code,message,requestId	Yes

例如：

```
{
  "code":0,
  "message": "success",
  "detail":
  {
    "qcvm6a456b0d8f01d4b2b1f5073d3fb8ccc0":
    {
      "code":0,
      "message": "",
      "requestId": "1231231231231";
    }
  }
}
```

```
"qcv6a456b0d8f01d4b2b1f5073d3fb8ccc0":  
{  
  "code":0,  
  "message":"","  
  "requestId":"1231231231232";  
}
```

注意：

资源全部操作成功，则最外层code为0

资源全部操作失败，则最外层code会返回5100

资源部分操作失败，则最外层code会返回5400

在第3种情况下，终端可以通过detail得到失败部分的操作信息。

示例代码

最近更新时间：2018-08-27 17:56:45

SDK 代码下载

- 示例代码 PHP 版
 - [转到 github 查看 PHP SDK >>](#)
- 示例代码 Java 版
 - [转到 github 查看 Java >>](#)
- 示例代码 Python 版
 - [转到 github 查看 Python 版 >>](#)
- 示例代码 .NET 版
 - [转到 github 查看 .NET SDK >>](#)

将示例代码中的 YOUR_SECRET_ID 和 YOUR_SECRET_KEY 替换成实际的 SecretId 和 SecretKey
示例代码仅供参考，请根据实际情况使用。

PHP 示例代码

```
<?php
```

```
/******真实调用时，需要根据不同接口修改下面的参数******/  
/******此处以 DescribeInstances 为例说明 如何获取指定 instanceId 的云服务器******/
```

```
/*DescribeInstances 接口的 URL 地址为 cvm.api.qcloud.com，可从对应的接口说明 “1.接口描述” 章节获  
取该接口的地址*/
```

```
$HttpUrl="cvm.api.qcloud.com";
```

```
/*除非有特殊说明，如MultipartUploadVodFile，其它接口都支持GET及POST*/
```

```
$HttpMethod="GET";
```

```
/*是否https协议，大部分接口都必须为https，只有少部分接口除外（如MultipartUploadVodFile）*/
```

```
$isHttps =true;
```

```
/*需要填写您的密钥，可从 https://console.cloud.tencent.com/capi 获取 SecretId 及 $secretKey*/
```

```
$secretKey='XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX';
```

```

/*下面这五个参数为所有接口的 公共参数；对于某些接口没有地域概念，则不用传递Region（如DescribeDea
ls）*/
$COMMON_PARAMS = array(
'Nonce'=> rand(),
'Timestamp'=>time(NULL),
'Action'=>'DescribeInstances',
'SecretId'=> 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX',
'Region' =>'gz',
);

/*下面这两个参数为 DescribeInstances 接口的私有参数，用于查询特定的云服务器列表*/
$PRIVATE_PARAMS = array(
'instanceIds.0'=> 'qcvm00001',
'instanceIds.1'=> 'qcvm00002',
);

/*****

CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAMS, $isHttp
s);

function CreateRequest($HttpUrl,$HttpMethod,$COMMON_PARAMS,$secretKey, $PRIVATE_PARAM
S, $isHttps)
{
$FullHttpUrl = $HttpUrl."/v2/index.php";

/*****对请求参数 按参数名 做字典序升序排列，注意此排序区分大小写*****/
$reqParaArray = array_merge($COMMON_PARAMS, $PRIVATE_PARAMS);
ksort($reqParaArray);

/*****生成签名原文*****/
* 将 请求方法, URI地址,及排序好的请求参数 按照下面格式 拼接在一起, 生成签名原文, 此请求中的原文为
* GETcvm.api.qcloud.com/v2/index.php?Action=DescribeInstances&Nonce=345122&Region=gz
* &SecretId=AKIDz8krbsJ5yKBZQ·1pn74WFkLPx3gnPhESA&Timestamp=1408704141
* &instanceIds.0=qcvm12345&instanceIds.1=qcvm56789
* *****/
$SigTxt = $HttpMethod.$FullHttpUrl."?";

$isFirst = true;
foreach ($reqParaArray as $key => $value)
{

```

```

if (!$isFirst)
{
    $SigTxt = $SigTxt."&";
}
$isFirst= false;

/*拼接签名原文时，如果参数名称中携带_，需要替换成.*/
if(strpos($key, '_'))
{
    $key = str_replace('_', '.', $key);
}

$SigTxt=$SigTxt.$key."=".$value;
}

/*****根据签名原字符串 $SigTxt，生成签名 Signature*****/
$Signature = base64_encode(hash_hmac('sha1', $SigTxt, $secretKey, true));

/*****拼接请求串,对于请求参数及签名，需要进行urlencode编码*****/
$Req = "Signature=" .urlencode($Signature);
foreach ($ReqParamArray as $key => $value)
{
    $Req=$Req."&".$key."=".$value;
}

/*****发送请求*****/
if($HttpMethod === 'GET')
{
    if($isHttps === true)
    {
        $Req="https://".$FullHttpUrl."?".$Req;
    }
    else
    {
        $Req="http://".$FullHttpUrl."?".$Req;
    }

    $Rsp = file_get_contents($Req);
}
else
{
    if($isHttps === true)
    {

```

```
$Rsp= SendPost("https://".$FullHttpUrl,$Req,$isHttps);
}
else
{
$Rsp= SendPost("http://".$FullHttpUrl,$Req,$isHttps);
}
}

var_export(json_decode($Rsp,true));
}

function SendPost($FullHttpUrl,$Req,$isHttps)
{

$ch = curl_init();
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $Req);

curl_setopt($ch, CURLOPT_URL, $FullHttpUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
if ($isHttps === true) {
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, false);
}

$result = curl_exec($ch);

return $result;
}
```