

**机器翻译**

**API 文档**

**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

### API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

接口鉴权 v3

接口鉴权

返回结果

API接口

图片翻译

语种识别

语音翻译

文本翻译

数据结构

错误码

# API 文档

## 更新历史

最近更新时间：2019-04-26 19:58:19

### 第 3 次发布

发布时间：2019-03-14 16:08:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [SpeechTranslate](#)
  - 新增入参：Mode
  - 新增出参：VadSeq

### 第 2 次发布

发布时间：2018-05-17 16:52:41

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ImageTranslate](#)

修改接口：

- [SpeechTranslate](#)
  - 新增入参：ProjectId
  - 新增出参：SessionUuid, RecognizeStatus, SourceText, TargetText, Seq, Source, Target

新增数据结构：

- [ImageRecord](#)
- [ItemValue](#)

### 第 1 次发布

发布时间：2018-04-24

本次发布包含了以下内容：

改善已有的文档。

---

新增接口：

- [LanguageDetect](#)
- [SpeechTranslate](#)
- [TextTranslate](#)

## 简介

最近更新时间：2019-06-20 19:45:23

**腾讯机器翻译 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。**

腾讯机器翻译（Tencent Machine Translation）结合了神经机器翻译和统计机器翻译的优点，从大规模双语语料库自动学习翻译知识，实现从源语言文本到目标语言文本的自动翻译，目前可支持中文与英文、日文、韩文三种语言的文本互译。

# API 概览

最近更新时间：2019-04-26 19:58:18

## API接口

接口名称	接口功能
<a href="#">ImageTranslate</a>	图片翻译
<a href="#">LanguageDetect</a>	语种识别
<a href="#">SpeechTranslate</a>	语音翻译
<a href="#">TextTranslate</a>	文本翻译

# 调用方式

## 请求结构

最近更新时间：2019-04-26 19:58:18

### 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 `tmt.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `tmt.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 `tmt.ap-guangzhou.tencentcloudapi.com` 是一致的。

**注意：对时延敏感的业务，建议指定带地域的域名。**

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>tmt.tencentcloudapi.com</code>
华南地区(广州)	<code>tmt.ap-guangzhou.tencentcloudapi.com</code>
华东地区(上海)	<code>tmt.ap-shanghai.tencentcloudapi.com</code>
华北地区(北京)	<code>tmt.ap-beijing.tencentcloudapi.com</code>
西南地区(成都)	<code>tmt.ap-chengdu.tencentcloudapi.com</code>
西南地区(重庆)	<code>tmt.ap-chongqing.tencentcloudapi.com</code>
东南亚地区(中国香港)	<code>tmt.ap-hongkong.tencentcloudapi.com</code>
东南亚地区(新加坡)	<code>tmt.ap-singapore.tencentcloudapi.com</code>
亚太地区(曼谷)	<code>tmt.ap-bangkok.tencentcloudapi.com</code>
亚太地区(孟买)	<code>tmt.ap-mumbai.tencentcloudapi.com</code>
亚太地区(首尔)	<code>tmt.ap-seoul.tencentcloudapi.com</code>
亚太地区(东京)	<code>tmt.ap-tokyo.tencentcloudapi.com</code>
美国东部(弗吉尼亚)	<code>tmt.na-ashburn.tencentcloudapi.com</code>
美国西部(硅谷)	<code>tmt.na-siliconvalley.tencentcloudapi.com</code>
北美地区(多伦多)	<code>tmt.na-toronto.tencentcloudapi.com</code>
欧洲地区(法兰克福)	<code>tmt.eu-frankfurt.tencentcloudapi.com</code>
欧洲地区(莫斯科)	<code>tmt.eu-moscow.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
华东地区(上海金融)	tmt.ap-shanghai-fsi.tencentcloudapi.com
华南地区(深圳金融)	tmt.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

支持的 HTTP 请求方法:

- POST (推荐)
- GET

POST 请求支持的 Content-Type 类型：

- application/json (推荐)，必须使用 TC3-HMAC-SHA256 签名方法。
- application/x-www-form-urlencoded，必须使用 HmacSHA1 或 HmacSHA256 签名方法。
- multipart/form-data (仅部分接口支持)，必须使用 TC3-HMAC-SHA256 签名方法。

GET 请求的请求包大小不得超过 32 KB。POST 请求使用签名方法为 HmacSHA1、HmacSHA256 时不得超过 1 MB。POST 请求使用签名方法为 TC3-HMAC-SHA256 时支持 10 MB。

## 4. 字符编码

均使用 UTF-8 编码。

## 公共参数

最近更新时间：2019-06-20 19:45:23

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。

### 签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
X-TC-Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，通常为域名前缀，例如域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 tmt； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要。
X-TC-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，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST ( application/json ) 请求结构示例 :

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

{"Offset":0,"Limit":10}
```

HTTP POST ( multipart/form-data ) 请求结构示例 ( 仅特定的接口支持 ) :

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

## 签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
------	----	----	----

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
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 <a href="#">云API密钥</a> 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	操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 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，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

```
Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/
```

```
Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded
```

```
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKIDEXAMPLE
```

## 地域列表

本产品所有接口 Region 字段的可选值如下表所示。如果接口不支持该表中的所有地域，则会在接口文档中单独说明。

区域	取值
亚太地区(曼谷)	ap-bangkok
华北地区(北京)	ap-beijing
西南地区(成都)	ap-chengdu
西南地区(重庆)	ap-chongqing
华南地区(广州)	ap-guangzhou
华南地区(广州Open)	ap-guangzhou-open
东南亚地区(中国香港)	ap-hongkong
亚太地区(孟买)	ap-mumbai
亚太地区(首尔)	ap-seoul
华东地区(上海)	ap-shanghai
华东地区(上海金融)	ap-shanghai-fsi
华南地区(深圳金融)	ap-shenzhen-fsi
东南亚地区(新加坡)	ap-singapore
欧洲地区(法兰克福)	eu-frankfurt
美国东部(弗吉尼亚)	na-ashburn
美国西部(硅谷)	na-siliconvalley
北美地区(多伦多)	na-toronto

# 接口鉴权 v3

最近更新时间：2019-06-20 19:45:23

腾讯云 API 会对每个请求进行身份验证，用户需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共请求参数中指定该签名结果并以指定的方式和格式发送请求。

## 申请安全凭证

本文使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。每个用户最多可以拥有两对密钥。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- **用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄漏，请立刻禁用该安全凭证。**

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】即可以创建一对密钥。

## 使用开发者资源

腾讯云 API 配套了 7 种常见的编程语言 SDK，均已开源，包括 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)。同时还提供了 [API Explorer](#) 供用户在线调用、签名验证、生成 SDK 代码。如果在签名时遇到困难，请善加利用这些资源。

## TC3-HMAC-SHA256 签名方法

TC3-HMAC-SHA256 签名方法相比以前的 HmacSHA1 和 HmacSHA256 签名方法，功能上覆盖了以前的签名方法，而且更安全，支持更大的请求，支持 json 格式，性能有一定提升，建议使用该签名方法计算签名。

云 API 支持 GET 和 POST 请求。对于 GET 方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于 POST 方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式绝大多数接口均支持，multipart 格式只有特定接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包。

下面以云服务器查询广州实例列表作为例子，分步骤介绍签名的计算过程。我们选择该接口是因为：

1. 云服务器默认已开通，该接口很常用；
2. 该接口是只读的，不会改变现有资源的状态；
3. 接口覆盖的参数种类较全，可以演示包含数据结构的数组如何使用。

在示例中，不论公共参数或者接口的参数，我们尽量选择容易犯错的情况。在实际调用接口时，请根据实际情况来，每个接口的参数并不相同，不要照抄这个例子的参数和值。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE。用户想查看广州虚拟机名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2019-02-25/cvm/tc3_re
quest, SignedHeaders=content-type;host, Signature=72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a9
6525168" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

## 1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST）。此示例取值为 POST。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠（/）。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要经过 URL 编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接； 2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。  此示例计算结果是 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。

字段名称	解释
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key (小写) 按照 ASCII 升序进行拼接，并且以分号 (;) 分隔。  此示例为 content-type;host
HashedRequestPayload	请求正文 (payload, 即 body, 此示例为 {"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}) 的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com

content-type;host
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

## 2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
Algorithm + \n +
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

字段名称	解释
Algorithm	签名算法，目前固定为 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 1551113065。
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串 (tc3_request)。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。



字段名称	解释
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031。

注意：

1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败，返回签名过期错误。

根据以上规则，示例中得到的待签名字符串如下：

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031
```

### 3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

字段名称	解释
SecretKey	原始的 SecretKey，即 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2) 计算签名，伪代码如下：

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

### 4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
Algorithm + ' ' +
```

```
'Credential=' + SecretId + '/' + CredentialScope + ',' +
'SignedHeaders=' + SignedHeaders + ',' +
'Signature=' + Signature
```

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId，即 AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host。
Signature	签名值。此示例计算结果是 72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=72e494ea809ad7a8c8f7a4507b9bddcbaa8e581f516e8da2f66e2c5a96525168
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}
```

## 5. 签名演示

### Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DataConverter;

public class TencentCloudAPITC3Demo {
```

```

private final static Charset UTF8 = StandardCharsets.UTF_8;
private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
private final static String CT_JSON = "application/json; charset=utf-8";

public static byte[] hmac256(byte[] key, String msg) throws Exception {
    Mac mac = Mac.getInstance("HmacSHA256");
    SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
    mac.init(secretKeySpec);
    return mac.doFinal(msg.getBytes(UTF8));
}

public static String sha256Hex(String s) throws Exception {
    MessageDigest md = MessageDigest.getInstance("SHA-256");
    byte[] d = md.digest(s.getBytes(UTF8));
    return DatatypeConverter.printHexBinary(d).toLowerCase();
}

public static void main(String[] args) throws Exception {
    String service = "cvm";
    String host = "cvm.tencentcloudapi.com";
    String region = "ap-guangzhou";
    String action = "DescribeInstances";
    String version = "2017-03-12";
    String algorithm = "TC3-HMAC-SHA256";
    String timestamp = "1551113065";
    //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    // 注意时区, 否则容易出错
    sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
    String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

    // ***** 步骤 1 : 拼接规范请求串 *****
    String httpRequestMethod = "POST";
    String canonicalUri = "/";
    String canonicalQueryString = "";
    String canonicalHeaders = "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n";
    String signedHeaders = "content-type;host";

    String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]";
    String hashedRequestPayload = sha256Hex(payload);
    String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
    + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
    System.out.println(canonicalRequest);

    // ***** 步骤 2 : 拼接待签名字符串 *****
    String credentialScope = date + "/" + service + "/" + "tc3_request";
    String hashedCanonicalRequest = sha256Hex(canonicalRequest);
    String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
    System.out.println(stringToSign);

    // ***** 步骤 3 : 计算签名 *****
    byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
    byte[] secretService = hmac256(secretDate, service);
    
```

```
byte[] secretSigning = hmac256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
System.out.println(signature);

// ***** 步骤 4 : 拼接 Authorization *****
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: ").append(authorization).append("\")")
.append(" -H \"Content-Type: application/json; charset=utf-8\"")
.append(" -H \"Host: ").append(host).append("\")")
.append(" -H \"X-TC-Action: ").append(action).append("\")")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\")")
.append(" -H \"X-TC-Version: ").append(version).append("\")")
.append(" -H \"X-TC-Region: ").append(region).append("\")")
.append(" -d ").append(payload).append("");
System.out.println(sb.toString());
}
}
```

## Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Name": "instance-name", "Values": [u"未命名"]}]}
```

```

# ***** 步骤 1 : 拼接规范请求串 *****
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ***** 步骤 2 : 拼接待签名字符串 *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)

# ***** 步骤 3 : 计算签名 *****
# 计算签名摘要函数
def sign(key, msg):
return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ***** 步骤 4 : 拼接 Authorization *****
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"
+ ' -H "Host: ' + host + '"
+ ' -H "X-TC-Action: ' + action + '"
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"
+ ' -H "X-TC-Version: ' + version + '"
+ ' -H "X-TC-Region: ' + region + '"
+ " -d '" + payload + "'")
    
```

## 签名失败

存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

# 接口鉴权

最近更新时间：2019-06-20 19:45:24

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。

注意：每个账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

参数名称	中文	参数值
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 "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&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

## 2.3. 拼接签名原字符串

此步骤生成签名原字符串。签名原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。

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：**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，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先用 UTF-8 进行编码。

**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。

**注意：**其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

## 4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在

错误代码	错误描述
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 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&InstanceIds.0=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.MAX_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() / 1000);
    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("InstanceIds.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, params[k]) for k in sorted(params))
    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：2019-04-26 19:58:19

### 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。
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。

# API接口

## 图片翻译

最近更新时间：2019-05-09 19:37:07

### 1. 接口描述

接口请求域名：tmt.tencentcloudapi.com。

提供中文到英文、英文到中文两种语言的图片翻译服务，可自动识别图片中的文本内容并翻译成目标语言，识别后的文本按行翻译，后续会提供可按段落翻译的版本。

提示：对于一般开发者，我们建议优先使用SDK接入简化开发。SDK使用介绍请直接查看 [5. 开发者资源](#) 部分。

默认接口请求频率限制：1次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，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，例如：tmt.ap-shanghai-fsi.tencentcloudapi.com。

### 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：ImageTranslate
Version	是	String	公共参数，本接口取值：2018-03-21
Region	是	String	公共参数，详见产品支持的 <a href="#">地域列表</a> 。
SessionUuid	是	String	唯一id，返回时原样返回
Scene	是	String	doc:文档扫描
Data	是	String	图片数据的Base64字符串，图片大小上限为4M，建议对源图片进行一定程度压缩
Source	是	String	源语言，支持语言列表 <ul style="list-style-type: none"><li>zh：中文</li><li>en：英文</li></ul>
Target	是	String	目标语言，支持语言列表 <ul style="list-style-type: none"><li>zh：中文</li><li>en：英文</li></ul>
ProjectId	是	Integer	项目ID，可以根据控制台-账号中心-项目管理中的配置填写，如无配置请填写默认项目ID:0

### 3. 输出参数

参数名称	类型	描述
SessionUuid	String	请求的SessionUuid返回
Source	String	源语言
Target	String	目标语言
ImageRecord	<a href="#">ImageRecord</a>	图片翻译结果，翻译结果按识别的文本每一行独立翻译，后续会推出按段落划分并翻译的版本
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 API调用

#### 输入示例

```
https://tmt.tencentcloudapi.com/?Action=ImageTranslate
&SessionUuid=session-00001
&Scene=doc
&Source=zh
&Target=en
&ProjectId=0
&Data=iVBORw0KGgoAAAANSUhEUGAAAdkAAABPCAYAAACnD7%2FoAAAAAXNSR0IArs4c6QAAARnQU1BAACxjwv8Y
QUAAAJcEhZcwAADsMAAA7DAJ
.....
CCyAAAAwCQgsgAAAMakILIAAADAJCCyAAAAwCQgsgAAAMakILIAADAFJbIX8%2F8oMOWYN6nAAAAAEIFTkSuQmCC
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "ImageRecord": {
      "Value": [
        {
          "H": 21,
          "SourceText": " 我们可以比较容易的将字典(ict)类型转为字符串(string)类型。",
          "TargetText": "We can easily convert the dictionary icttype to the string) type.",
          "W": 373,
          "X": 12,
          "Y": 53
        }
      ]
    },
    "Source": "zh",
    "Target": "en",
    "SessionUuid": "session-00001",
    "RequestId": "6e698139-615a-4d42-8dea-6bfada24e83c"
  }
}
```



## 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.ServiceIsolate	账号因为欠费停止服务，请在腾讯云账户充值。
FailedOperation.UserNotRegistered	服务未开通，请在腾讯云官网机器翻译控制台开通服务。
InternalServerError	内部错误。
InternalServerError.BackendTimeout	后台服务超时，请稍后重试。
InternalServerError.ErrorUnknown	未知错误。
InvalidParameter	参数错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
UnauthorizedOperation.ActionNotFound	请填写正确的Action字段名称。
UnsupportedOperation	操作不支持。
UnsupportedOperation.TextTooLong	单次请求text超过长长度限制，请保证单次请求长长度低于2000。

错误码	描述
UnsupportedOperation.UnSupportedTargetLanguage	不支持的目标语言，请参照语言列表。
UnsupportedOperation.UnsupportedLanguage	不支持的语言，请参照语言列表。
UnsupportedOperation.UnsupportedSourceLanguage	不支持的源语言，请参照语言列表。

# 语种识别

最近更新时间：2019-05-09 19:37:07

## 1. 接口描述

接口请求域名：tmt.tencentcloudapi.com。

可自动识别文本内容的语言种类，轻量高效，无需额外实现判断方式，使面向客户的服务体验更佳。

提示：对于一般开发者，我们建议优先使用SDK接入简化开发。SDK使用介绍请直接查看 [5. 开发者资源](#) 部分。

默认接口请求频率限制：5次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，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，例如：tmt.ap-shanghai-fsi.tencentcloudapi.com。

## 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：LanguageDetect
Version	是	String	公共参数，本接口取值：2018-03-21
Region	是	String	公共参数，详见产品支持的 <a href="#">地域列表</a> 。
Text	是	String	待识别的文本，文本统一使用utf-8格式编码，非utf-8格式编码字符会翻译失败。单次请求的文本长度需要低于2000。
ProjectId	是	Integer	项目ID，可以根据控制台-账号中心-项目管理中的配置填写，如无配置请填写默认项目ID:0

## 3. 输出参数

参数名称	类型	描述
------	----	----

参数名称	类型	描述
Lang	String	识别出的语言种类，参考语言列表 <ul style="list-style-type: none"> <li>• zh : 中文</li> <li>• en : 英文</li> <li>• jp : 日语</li> <li>• kr : 韩语</li> <li>• de : 德语</li> <li>• fr : 法语</li> <li>• es : 西班牙文</li> <li>• it : 意大利文</li> <li>• tr : 土耳其文</li> <li>• ru : 俄文</li> <li>• pt : 葡萄牙文</li> <li>• vi : 越南文</li> <li>• id : 印度尼西亚文</li> <li>• ms : 马来西亚文</li> <li>• th : 泰文</li> </ul>
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 API调用

#### 输入示例

```
https://tmt.tencentcloudapi.com/?Action=LanguageDetect
&Text=hello
&ProjectId=0
&<公共请求参数>
```

#### 输出示例

```
{
  "Response": {
    "Lang": "en",
    "RequestId": "155a6879-2c39-4e36-b79c-50a68d073ffc"
  }
}
```

## 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.ServiceIsolate	账号因为欠费停止服务，请在腾讯云账户充值。
FailedOperation.UserNotRegistered	服务未开通，请在腾讯云官网机器翻译控制台开通服务。
InternalServerError	内部错误。
InternalServerError.BackendTimeout	后台服务超时，请稍后重试。
InternalServerError.ErrorUnknown	未知错误。
InvalidParameter	参数错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
UnauthorizedOperation.ActionNotFound	请填写正确的Action字段名称。
UnsupportedOperation	操作不支持。
UnsupportedOperation.TextTooLong	单次请求text超过长长度限制，请保证单次请求长长度低于2000。
UnsupportedOperation.UnSupportedTargetLanguage	不支持的目标语言，请参照语言列表。
UnsupportedOperation.UnsupportedLanguage	不支持的语言，请参照语言列表。
UnsupportedOperation.UnsupportedSourceLanguage	不支持的源语言，请参照语言列表。

# 语音翻译

最近更新时间：2019-06-20 19:45:24

## 1. 接口描述

接口请求域名：tmt.tencentcloudapi.com。

本接口提供音频内文字识别 + 翻译功能，目前开放中到英的语音翻译服务。待识别和翻译的音频文件可以是 pcm、mp3、amr和speex 格式，音频内语音清晰，采用流式传输和翻译的方式。

提示：对于一般开发者，我们建议优先使用SDK接入简化开发。SDK使用介绍请直接查看 5. 开发者资源 部分。

默认接口请求频率限制：5次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，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，例如：tmt.ap-shanghai-fsi.tencentcloudapi.com。

## 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：SpeechTranslate
Version	是	String	公共参数，本接口取值：2018-03-21
Region	是	String	公共参数，详见产品支持的 <a href="#">地域列表</a> 。
SessionUuid	是	String	一段完整的语音对应一个SessionUuid
Source	是	String	音频中的语言类型，支持语言列表 <ul style="list-style-type: none"><li>zh：中文</li><li>en：英文</li></ul>
Target	是	String	翻译目标语言言类型，支持的语言列表 <ul style="list-style-type: none"><li>zh：中文</li><li>en：英文</li></ul>
AudioFormat	是	Integer	pcm：146 amr：33554432 mp3：83886080
Seq	是	Integer	语音分片的序号，从0开始
IsEnd	是	Integer	是否最后一片语音分片，0-否，1-是
Data	是	String	语音分片内容的base64字符串，音频内容应含有效并可识别的文本
ProjectId	否	Integer	项目ID，可以根据控制台-账号中心-项目管理中的配置填写，如无配置请填写默认项目ID:0
Mode	否	String	识别模式，该参数已废弃

### 3. 输出参数

参数名称	类型	描述
SessionUuid	String	请求的SessionUuid直接返回
RecognizeStatus	Integer	语音识别状态 1-进行中 0-完成
SourceText	String	识别出的原文
TargetText	String	翻译出的译文
Seq	Integer	第几个语音分片
Source	String	源语言
Target	String	目标语言
VadSeq	Integer	当请求的Mode参数填写bvad是，启动VadSeq。此时Seq会被设置为后台vad（静音检测）后的新序号，而VadSeq代表客户端原始Seq值
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 API调用

##### 输入示例

```
https://tmt.tencentcloudapi.com/?Action=SpeechTranslate
&ProjectId=0
&SessionUuid=sid-1516105689129
&Source=zh
&Target=en
&AudioFormat=83886080
&Seq=0
&IsEnd=1
&Data=//tQxAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA
.....
MGCwUUAU4UeWgJDOGmC2JgzouqHJQBszIKUicVTU5UGTZ6mVJYIri5s9geZdCjBnExqtEsaw/5rtShQ6UOkolgkGSpWK
&<公共请求参数>
```

##### 输出示例

```
{
  "Response": {
    "RecognizeStatus": 0,
    "RequestId": "6e698139-615a-4d42-8dea-6bfada24e83c",
    "Seq": 0,
    "SessionUuid": "sid-1516105689129",
    "Source": "zh",
```

```
"SourceText": "你好。",
"Target": "en",
"TargetText": "Hello."
}
}
```

## 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.ServiceIsolate	账号因为欠费停止服务，请在腾讯云账户充值。
FailedOperation.UserNotRegistered	服务未开通，请在腾讯云官网机器翻译控制台开通服务。
InternalServerError	内部错误。
InternalServerError.BackendTimeout	后台服务超时，请稍后重试。
InternalServerError.ErrorUnknown	未知错误。
InvalidParameter	参数错误。
InvalidParameter.DuplicatedSessionIdAndSeq	重复的SessionUuid和Seq组合。
InvalidParameter.SeqIntervalTooLarge	Seq之间的间隙请不要大于2000。
LimitExceeded	超过配额限制。



错误码	描述
MissingParameter	缺少参数错误。
UnauthorizedOperation.ActionNotFound	请填写正确的Action字段名称。
UnsupportedOperation	操作不支持。
UnsupportedOperation.TextTooLong	单次请求text超过长长度限制，请保证单次请求长长度低于2000。
UnsupportedOperation.UnSupportedTargetLanguage	不支持的目标语言，请参照语言列表。
UnsupportedOperation.UnsupportedLanguage	不支持的语言，请参照语言列表。
UnsupportedOperation.UnsupportedSourceLanguage	不支持的源语言，请参照语言列表。

# 文本翻译

最近更新时间：2019-06-20 19:45:24

## 1. 接口描述

接口请求域名：tmt.tencentcloudapi.com。

提供中文到英文、英文到中文的等多种语言的文本内容翻译服务，经过大数据语料库、多种解码算法、翻译引擎深度优化，在新闻文章、生活口语等不同语言场景中都有深厚积累，翻译结果专业评价处于行业领先水平。

提示：对于一般开发者，我们建议优先使用SDK接入简化开发。SDK使用介绍请直接查看 5. 开发者资源 部分。

默认接口请求频率限制：5次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，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，例如：tmt.ap-shanghai-fsi.tencentcloudapi.com。

## 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：TextTranslate
Version	是	String	公共参数，本接口取值：2018-03-21
Region	是	String	公共参数，详见产品支持的 <a href="#">地域列表</a> 。
SourceText	是	String	待翻译的文本，文本统一使用utf-8格式编码，非utf-8格式编码字符会翻译失败，请传入有效文本，html标记等非常规翻译文本会翻译失败。单次请求的文本长度需要低于2000。
Source	是	String	源语言，参照Target支持语言列表

参数名称	必选	类型	描述
Target	是	String	目标语言，参照支持语言列表 <ul style="list-style-type: none"> <li>• zh: 简体中文</li> <li>• zh-TW: 繁体中文</li> <li>• en: 英文</li> <li>• jp: 日语</li> <li>• kr: 韩语</li> <li>• de: 德语</li> <li>• fr: 法语</li> <li>• es: 西班牙文</li> <li>• it: 意大利文</li> <li>• tr: 土耳其文</li> <li>• ru: 俄文</li> <li>• pt: 葡萄牙文</li> <li>• vi: 越南文</li> <li>• id: 印度尼西亚文</li> <li>• ms: 马来西亚文</li> <li>• th: 泰文</li> <li>• auto: 自动识别源语言，只能用于source字段</li> </ul>
ProjectId	是	Integer	项目ID，可以根据控制台-账号中心-项目管理中的配置填写，如无配置请填写默认项目ID:0

### 3. 输出参数

参数名称	类型	描述
TargetText	String	翻译后的文本
Source	String	源语言，详见入参Target
Target	String	目标语言，详见入参Target
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 API调用

##### 输入示例

```

https://tmt.tencentcloudapi.com/?Action=TextTranslate
&ProjectId=0
&Source=en
&SourceText=hello
&Target=zh
&<公共请求参数>
    
```

##### 输出示例

```
{
  "Response": {
    "TargetText": "你好",
    "Source": "en",
    "Target": "zh",
    "RequestId": "000ee211-f19e-4a34-a214-e2bb1122d248"
  }
}
```

## 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.ServiceIsolate	账号因为欠费停止服务，请在腾讯云账户充值。
FailedOperation.UserNotRegistered	服务未开通，请在腾讯云官网机器翻译控制台开通服务。
InternalServerError	内部错误。
InternalServerError.BackendTimeout	后台服务超时，请稍后重试。
InternalServerError.ErrorUnknown	未知错误。
InvalidParameter	参数错误。

错误码	描述
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
UnauthorizedOperation.ActionNotFound	请填写正确的Action字段名称。
UnsupportedOperation	操作不支持。
UnsupportedOperation.TextTooLong	单次请求text超过长长度限制，请保证单次请求长长度低于2000。
UnsupportedOperation.UnSupportedTargetLanguage	不支持的目标语言，请参照语言列表。
UnsupportedOperation.UnsupportedLanguage	不支持的语言，请参照语言列表。
UnsupportedOperation.UnsupportedSourceLanguage	不支持的源语言，请参照语言列表。

# 数据结构

最近更新时间：2019-04-26 19:58:19

## ImageRecord

图片翻译结果

被如下接口引用：ImageTranslate。

名称	类型	描述
Value	Array of <a href="#">ItemValue</a>	图片翻译结果

## ItemValue

翻译结果

被如下接口引用：ImageTranslate。

名称	类型	描述
SourceText	String	识别出的源文
TargetText	String	翻译后的译文
X	Integer	X坐标
Y	Integer	Y坐标
W	Integer	宽度
H	Integer	高度

# 错误码

最近更新时间：2019-06-20 19:45:24

## 功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "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 中的 Code 表示错误码，Message 表示该错误的具体信息。

## 错误码列表

### 公共错误码

错误码	说明
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。请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 <a href="#">CAM</a> 文档对鉴权的说明。
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。

错误码	说明
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。
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。

### 业务错误码

错误码	说明
FailedOperation.ServiceIsolate	账号因为欠费停止服务，请在腾讯云账户充值。
FailedOperation.UserNotRegistered	服务未开通，请在腾讯云官网机器翻译控制台开通服务。
InternalServerError	内部错误。
InternalServerError.BackendTimeout	后台服务超时，请稍后重试。
InternalServerError.ErrorUnknown	未知错误。
InvalidParameter	参数错误。
InvalidParameter.DuplicatedSessionIdAndSeq	重复的SessionUuid和Seq组合。
InvalidParameter.SeqIntervalTooLarge	Seq之间的间隙请不要大于2000。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
UnauthorizedOperation.ActionNotFound	请填写正确的Action字段名称。
UnsupportedOperation	操作不支持。



错误码	说明
UnsupportedOperation.TextTooLong	单次请求text超过长长度限制，请保证单次请求长长度低于2000。
UnsupportedOperation.UnSupportedTargetLanguage	不支持的目标语言，请参照语言列表。
UnsupportedOperation.UnsupportedLanguage	不支持的语言，请参照语言列表。
UnsupportedOperation.UnsupportedSourceLanguage	不支持的源语言，请参照语言列表。