

密钥管理系统

API 文档



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 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。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

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默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

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

签名方法 v3

签名方法

返回结果

参数类型

密钥相关接口

修改主密钥描述信息

修改别名

密文刷新

获取主密钥列表

获取主密钥列表详情

查询服务状态

查询密钥轮换状态

生成数据密钥

加密

批量启动主密钥

开启密钥轮换

启用主密钥

批量禁用主密钥

禁止密钥轮换

禁用主密钥

获取多个主密钥属性

获取主密钥属性

解密

创建主密钥

CMK计划删除接口

取消CMK计划删除操作

导入密钥材料

获取导入主密钥（CMK）材料的参数

删除导入的密钥材料

随机数生成接口

列出当前Region支持的加密方式

解绑CMK和云资源的关联关系

绑定密钥和云产品资源的使用关系

获取可以提供KMS服务的地域列表

取消密钥归档

密钥归档

后量子密码算法加密

后量子密码算法解密

非对称密钥相关接口

获取非对称密钥的公钥

非对称密钥Sm2解密

非对称密钥RSA解密

验证签名

签名

后量子密码算法验签

后量子密码算法签名

白盒密钥相关接口

使用白盒密钥进行加密

- 批量启用白盒密钥
- 启用白盒密钥
- 批量禁用白盒密钥
- 禁用白盒密钥
- 获取白盒密钥服务状态
- 获取白盒密钥列表
- 展示白盒密钥的信息
- 获取白盒解密密钥
- 删除白盒密钥
- 创建白盒密钥
- 覆盖指定密钥的设备指纹信息
- 获取指定密钥的设备指纹列表

数据结构

错误码

密钥管理系统 API/SDK 2017

简介

API 概览

更新历史

调用方式

请求结构

公共请求参数

最终请求形式

请求结构简介

返回结果

错误返回结果

正确返回结果

错误码

签名方法

地域说明

加解密相关接口

加密

解密

密钥管理相关接口

修改主密钥属性

创建主密钥

启用主密钥

生成数据密钥

禁用主密钥

获取主密钥列表

获取主密钥属性

API 文档

更新历史

最近更新时间：2025-04-17 01:21:26

第 23 次发布

发布时间：2025-04-17 01:21:23

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetServiceStatus](#)
 - 新增出参：ExclusiveHSMList

新增数据结构：

- [ExclusiveHSM](#)

第 22 次发布

发布时间：2025-03-14 01:19:50

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetServiceStatus](#)
 - 新增出参：CmkUserCount, CmkLimit

第 21 次发布

发布时间：2024-11-06 01:46:44

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [KeyMetadata](#)
 - 新增成员：RotateDays, LastRotateTime

第 20 次发布

发布时间：2023-04-28 01:28:18

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [PostQuantumCryptoDecrypt](#)
- [PostQuantumCryptoEncrypt](#)
- [PostQuantumCryptoSign](#)
- [PostQuantumCryptoVerify](#)

第 19 次发布

发布时间：2023-03-03 01:26:52

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetServiceStatus](#)
 - 新增出参：SubscriptionInfo

第 18 次发布

发布时间：2022-12-13 10:37:12

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [EnableKeyRotation](#)
 - 新增入参：RotateDays

第 17 次发布

发布时间：2022-05-31 06:13:25

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateKey](#)
 - 新增入参：HsmClusterId
 - 新增出参：HsmClusterId
- [GetServiceStatus](#)
 - 新增出参：ExclusiveVSMEnabled, ExclusiveHSMEnabled
- [ListKeyDetail](#)
 - 新增入参：HsmClusterId
- [ListKeys](#)
 - 新增入参：HsmClusterId

修改数据结构：

- [KeyMetadata](#)
 - 新增成员：HsmClusterId

第 16 次发布

发布时间：2021-12-23 08:12:09

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [Decrypt](#)
 - 新增入参：EncryptionPublicKey, EncryptionAlgorithm
- [GenerateDataKey](#)
 - 新增入参：EncryptionPublicKey, EncryptionAlgorithm

第 15 次发布

发布时间：2021-01-15 08:04:59

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [SignByAsymmetricKey](#)
- [VerifyByAsymmetricKey](#)

修改接口：

- [ListAlgorithms](#)
 - 新增出参：AsymmetricSignVerifyAlgorithms

第 14 次发布

发布时间：2020-10-29 08:04:45

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetServiceStatus](#)
 - 新增出参：UserLevel, ProExpireTime, ProRenewFlag, ProResourceId

第 13 次发布

发布时间：2020-09-01 08:04:17

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ArchiveKey](#)
- [CancelKeyArchive](#)

第 12 次发布

发布时间：2020-08-25 08:05:11

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetRegions](#)

第 11 次发布

发布时间：2020-08-17 08:13:41

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeWhiteBoxKeyDetails](#)
 - 新增入参：Offset, Limit, TagFilters
 - 新增出参：TotalCount
- [ListKeyDetail](#)
 - 新增入参：TagFilters

新增数据结构：

- [TagFilter](#)

修改数据结构：

- [WhiteboxKeyInfo](#)
 - 新增成员：DeviceFingerprintBind

第 10 次发布

发布时间：2020-07-28 08:12:44

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [BindCloudResource](#)
- [UnbindCloudResource](#)

第 9 次发布

发布时间：2020-06-29 08:12:11

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeWhiteBoxDeviceFingerprints](#)
- [OverwriteWhiteBoxDeviceFingerprints](#)

新增数据结构：

- [DeviceFingerprint](#)

第 8 次发布

发布时间：2020-06-03 08:12:31

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateKey](#)
 - 新增入参：Tags
 - 新增出参：TagCode, TagMsg
- [CreateWhiteBoxKey](#)
 - 新增入参：Tags
 - 新增出参：TagCode, TagMsg

新增数据结构：

- [Tag](#)

第 7 次发布

发布时间：2020-04-27 08:11:13

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [KeyMetadata](#)
 - 新增成员：ResourceId
- [WhiteboxKeyInfo](#)
 - 新增成员：ResourceId

第 6 次发布

发布时间：2020-04-15 08:11:54

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateWhiteBoxKey](#)
- [DeleteWhiteBoxKey](#)
- [DescribeWhiteBoxDecryptKey](#)
- [DescribeWhiteBoxKey](#)
- [DescribeWhiteBoxKeyDetails](#)
- [DescribeWhiteBoxServiceStatus](#)
- [DisableWhiteBoxKey](#)
- [DisableWhiteBoxKeys](#)
- [EnableWhiteBoxKey](#)
- [EnableWhiteBoxKeys](#)
- [EncryptByWhiteBox](#)

新增数据结构：

- [WhiteboxKeyInfo](#)

第 5 次发布

发布时间：2020-03-12 07:59:55

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AsymmetricRsaDecrypt](#)
- [AsymmetricSm2Decrypt](#)
- [GetPublicKey](#)
- [ListAlgorithms](#)

修改接口：

- [ListKeyDetail](#)
 - 新增入参：KeyUsage

新增数据结构：

- [AlgorithmInfo](#)

第 4 次发布

发布时间：2019-10-28 15:55:31

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GenerateRandom](#)

第 3 次发布

发布时间：2019-10-17 19:17:56

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteImportedKeyMaterial](#)
- [GetParametersForImport](#)
- [ImportKeyMaterial](#)

修改接口：

- [ListKeyDetail](#)
 - 新增入参：Origin

修改数据结构：

- [KeyMetadata](#)
 - 新增成员：Origin, ValidTo

第 2 次发布

发布时间：2019-07-04 17:15:14

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CancelKeyDeletion](#)
- [ScheduleKeyDeletion](#)

修改接口：

- [GetServiceStatus](#)
 - 新增出参：InvalidType

修改数据结构：

- [KeyMetadata](#)
 - 新增成员：DeletionDate

第 1 次发布

发布时间：2019-04-25 17:03:55

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateKey](#)
- [Decrypt](#)
- [DescribeKey](#)
- [DescribeKeys](#)
- [DisableKey](#)
- [DisableKeyRotation](#)
- [DisableKeys](#)
- [EnableKey](#)
- [EnableKeyRotation](#)
- [EnableKeys](#)
- [Encrypt](#)
- [GenerateDataKey](#)
- [GetKeyRotationStatus](#)
- [GetServiceStatus](#)
- [ListKeyDetail](#)

- [ListKeys](#)
- [ReEncrypt](#)
- [UpdateAlias](#)
- [UpdateKeyDescription](#)

新增数据结构：

- [Key](#)
- [KeyMetadata](#)

简介

最近更新时间：2025-04-25 01:40:52

腾讯云密钥管理系统（Key Management Service）是一项保护数据及密钥安全的密钥服务，使用该服务，您的数据或密钥等信息将获得更高安全性，腾讯云为您保证服务的高可用，使您可以更专注于数据保护的开发。

注意：

- 本章节密钥管理系统 API 接口均为最新 API 3.0 接口，后续密钥管理系统相关新增功能都会在此章节更新。我们强烈推荐您使用最新 API3.0 接口。

API 概览

最近更新时间：2024-08-02 01:47:39

非对称密钥相关接口

接口名称	接口功能	频率限制（次/秒）
AsymmetricRsaDecrypt	非对称密钥RSA解密	200
AsymmetricSm2Decrypt	非对称密钥Sm2解密	200
GetPublicKey	获取非对称密钥的公钥	100
PostQuantumCryptoSign	后量子密码算法签名	20
PostQuantumCryptoVerify	后量子密码算法验签	20
SignByAsymmetricKey	签名	100
VerifyByAsymmetricKey	验证签名	100

白盒密钥相关接口

接口名称	接口功能	频率限制（次/秒）
CreateWhiteBoxKey	创建白盒密钥	100
DeleteWhiteBoxKey	删除白盒密钥	200
DescribeWhiteBoxDecryptKey	获取白盒解密密钥	200
DescribeWhiteBoxDeviceFingerprints	获取指定密钥的设备指纹列表	100
DescribeWhiteBoxKey	展示白盒密钥的信息	200
DescribeWhiteBoxKeyDetails	获取白盒密钥列表	100
DescribeWhiteBoxServiceStatus	获取白盒密钥服务状态	200
DisableWhiteBoxKey	禁用白盒密钥	200
DisableWhiteBoxKeys	批量禁用白盒密钥	200
EnableWhiteBoxKey	启用白盒密钥	200
EnableWhiteBoxKeys	批量启用白盒密钥	200
EncryptByWhiteBox	使用白盒密钥进行加密	100
OverwriteWhiteBoxDeviceFingerprints	覆盖指定密钥的设备指纹信息	100

密钥相关接口

接口名称	接口功能	频率限制（次/秒）
ArchiveKey	密钥归档	100
BindCloudResource	绑定密钥和云产品资源的使用关系	100
CancelKeyArchive	取消密钥归档	100
CancelKeyDeletion	取消CMK计划删除操作	100
CreateKey	创建主密钥	100
Decrypt	解密	300
DeleteImportedKeyMaterial	删除导入的密钥材料	100

接口名称	接口功能	频率限制（次/秒）
DescribeKey	获取主密钥属性	100
DescribeKeys	获取多个主密钥属性	100
DisableKey	禁用主密钥	100
DisableKeyRotation	禁止密钥轮换	100
DisableKeys	批量禁用主密钥	100
EnableKey	启用主密钥	100
EnableKeyRotation	开启密钥轮换	100
EnableKeys	批量启动主密钥	100
Encrypt	加密	300
GenerateDataKey	生成数据密钥	100
GenerateRandom	随机数生成接口	150
GetKeyRotationStatus	查询密钥轮换状态	100
GetParametersForImport	获取导入主密钥（CMK）材料的参数	100
GetRegions	获取可以提供KMS服务的地域列表	100
GetServiceStatus	查询服务状态	100
ImportKeyMaterial	导入密钥材料	100
ListAlgorithms	列出当前Region支持的加密方式	100
ListKeyDetail	获取主密钥列表详情	100
ListKeys	获取主密钥列表	100
PostQuantumCryptoDecrypt	后量子密码算法解密	20
PostQuantumCryptoEncrypt	后量子密码算法加密	20
ReEncrypt	密文刷新	100
ScheduleKeyDeletion	CMK计划删除接口	100
UnbindCloudResource	解绑CMK和云资源的关联关系	100
UpdateAlias	修改别名	100
UpdateKeyDescription	修改主密钥描述信息	100

注意：

以上给出的接口频率限制维度为 API + 接入地域 + 子账号，有关限频更多说明参考：[API 频率限制说明](#)

调用方式

请求结构

最近更新时间：2025-05-21 01:22:50

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 kms.tencentcloudapi.com ，也支持指定地域域名访问，例如广州地域的域名为 kms.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 kms.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。

注意：域名是 API 的接入点，并不代表产品或者接口实际提供服务的地域。产品支持的地域列表请在调用方式/公共参数文档中查阅，接口支持的地域请在接口文档输入参数中查阅。

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）	kms.tencentcloudapi.com
华南地区（广州）	kms.ap-guangzhou.tencentcloudapi.com
华东地区（上海）	kms.ap-shanghai.tencentcloudapi.com
华东地区（南京）	kms.ap-nanjing.tencentcloudapi.com
华北地区（北京）	kms.ap-beijing.tencentcloudapi.com
西南地区（成都）	kms.ap-chengdu.tencentcloudapi.com
西南地区（重庆）	kms.ap-chongqing.tencentcloudapi.com
港澳台地区（中国香港）	kms.ap-hongkong.tencentcloudapi.com
亚太东南（新加坡）	kms.ap-singapore.tencentcloudapi.com
亚太东南（雅加达）	kms.ap-jakarta.tencentcloudapi.com
亚太东南（曼谷）	kms.ap-bangkok.tencentcloudapi.com
亚太东北（首尔）	kms.ap-seoul.tencentcloudapi.com
亚太东北（东京）	kms.ap-tokyo.tencentcloudapi.com
美国东部（弗吉尼亚）	kms.na-ashburn.tencentcloudapi.com
美国西部（硅谷）	kms.na-siliconvalley.tencentcloudapi.com
南美地区（圣保罗）	kms.sa-saopaulo.tencentcloudapi.com
欧洲地区（法兰克福）	kms.eu-frankfurt.tencentcloudapi.com

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）	kms.ap-shanghai-fsi.tencentcloudapi.com
华南地区（深圳金融）	kms.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), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。
- application/x-www-form-urlencoded, 必须使用签名方法 v1 (HmacSHA1 或 HmacSHA256)。
- multipart/form-data (仅部分接口支持), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。

GET 请求的请求包大小不得超过32KB。POST 请求使用签名方法 v1 (HmacSHA1、HmacSHA256) 时不得超过1MB。POST 请求使用签名方法 v3 (TC3-HMAC-SHA256) 时支持10MB。

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2025-05-16 01:42:33

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的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。

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1（有些文档可能会简称签名方法），更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升，推荐使用该签名方法计算签名。完整介绍详见 [签名方法 v3](#)。

注意：出于简化的目的，部分接口文档中的示例使用的是签名方法 v1 GET 请求，而不是更安全的签名方法 v3。

使用签名方法 v3 时，公共参数需要统一放到 HTTP Header 请求头部中，如下表所示：

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 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	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 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	HTTP 请求头：X-TC-Timestamp。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。
Version	String	是	HTTP 请求头：X-TC-Version。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=AKID***/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，AKID*** 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 为具体产品名，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 kms；tc3_request 为固定字符串； - 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，计算过程详见 文档 。
Token	String	否	HTTP 请求头：X-TC-Token。即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表中的前十个，接口参数设置为偏移量 Offset=0，返回数量 Limit=10，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=AKID*****/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=AKID*****/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=AKID*****/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--
```

使用签名方法 v1 的公共参数

使用签名方法 v1（有时会称作 HmacSHA256 和 HmacSHA1），公共参数需要统一放到请求串中，完整介绍详见[文档](#)

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章节关于公共参数 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	—	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 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 文档 。

参数名称	类型	必选	描述
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 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	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

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=AKID*****

Host: cvm.tencentcloudapi.com
```

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=AKID*****
***
```

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。

地域	取值
亚太东南（曼谷）	ap-bangkok
华北地区（北京）	ap-beijing
华南地区（广州）	ap-guangzhou
港澳台地区（中国香港）	ap-hongkong
亚太东南（雅加达）	ap-jakarta
华东地区（南京）	ap-nanjing
亚太东北（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
亚太东南（新加坡）	ap-singapore
亚太东北（东京）	ap-tokyo
欧洲地区（法兰克福）	eu-frankfurt

签名方法 v3

最近更新时间：2024-12-25 01:43:50

以下文档说明了签名方法 v3 的签名过程，但仅在您编写自己的代码来调用腾讯云 API 时才有用。我们推荐您使用 [腾讯云 API Explorer](#)，[腾讯云 SDK](#) 和 [腾讯云命令行工具（TCCLI）](#) 等开发者工具，从而无需学习如何对 API 请求进行签名。

推荐使用 API Explorer

</> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

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参数中指定该签名结果并以指定的方式和格式发送请求。

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份

签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 [云 API 密钥](#) 页面获取密钥相关信息。

2. 保护传输中的数据

为了防止请求在传输过程中被篡改，腾讯云 API 会使用请求参数来计算请求的哈希值，并将生成的哈希值加密后作为请求的一部分，发送到腾讯云 API 服务器。服务器会使用收到的请求参数以同样的过程计算哈希值，并验证请求中的哈希值。如果请求被篡改，将导致哈希值不一致，腾讯云 API 将拒绝本次请求。

签名方法 v3（TC3-HMAC-SHA256）功能上覆盖了以前的签名方法 v1，而且更安全，支持更大的请求，支持 JSON 格式，POST 请求支持传空数组和空字符串，性能有一定提升，推荐使用该签名方法计算签名。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以对生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

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。如已泄露，请立刻禁用该安全凭证。

申请安全凭证的具体步骤如下：

- 登录 [腾讯云管理中心控制台](#)。
- 前往 [云API密钥](#) 的控制台页面。
- 在 [云API密钥](#) 页面，单击【新建密钥】创建一对密钥。

签名版本 v3 签名过程

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：

- 云服务器默认已开通，该接口很常用；
- 该接口是只读的，不会改变现有资源的状态；
- 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。此外，这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数（在 HTTP 头部设置，添加 X-TC-前缀）。

假设用户的 SecretId 和 SecretKey 分别是：AKID***** 和 *****。用户想查看广州区云服务器名为“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=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f" \
-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 中间号 (?) 后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 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\nx-tc-action:describeinstances\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。
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;x-tc-action
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

字段名称	解释
	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
x-tc-action:describeinstances

content-type;host;x-tc-action
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。
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 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

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，可能运行一段时间后，请求失败，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
7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

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

派生出的密钥 SecretDate、SecretService 和 SecretSigning 是二进制的数，可能包含不可打印字符，将其转为十六进制字符串打印的输出分别为：
da98fb70dcf6b112dc21038d1eeeb3a95c74b4dcb12c1131f864f6066bd02be0，
8d70cbefb03939f929db64d32dc2ba89b1095620119fe3e050e2b18c5bd2752f，
b596b923aad85185e2d1f6659d2a062e0a86731226e021e61bfe06f7ed05f5af。

请注意，不同的编程语言，HMAC 库函数中参数顺序可能不一样，请以实际情况为准。此处的伪代码密钥参数 key 在前，消息参数 data 在后。通常标准库函数会提供二进制格式的返回值，也可能会提供打印友好的十六进制格式的返回值，此处使用的是二进制格式。

字段名称	解释
SecretKey	原始的 SecretKey，即 *****。
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))
```

此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

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 AKID*****。
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;x-tc-action。
Signature	签名值。此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
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"}]}
```

⚠ 注意:

请求发送时的 HTTP 头部（Header）和请求体（Payload）必须和签名计算过程中的内容完全一致，否则会返回签名不一致错误。可以通过打印实际请求内容，网络抓包等方式对比排查。

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](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚地解释签名过程，下面以实际编程语言为例，将上述的签名过程完整实现。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。

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.DatatypeConverter;

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
}
```

```
private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
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" + "x-tc-action:" + action.toLowerCase() + "\n";
    String signedHeaders = "content-type;host;x-tc-action";

    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
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

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": [{"Values": [u"未命名"], "Name": "instance-name"}]}

# ***** 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\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
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 + '"')
```

Golang

```
package main

import (
    "crypto/hmac"
```

```
"crypto/sha256"
"encoding/hex"
"fmt"
"os"
"strings"
"time"
)

func sha256hex(s string) string {
    b := sha256.Sum256([]byte(s))
    return hex.EncodeToString(b[:])
}

func hmacsha256(s, key string) string {
    hashed := hmac.New(sha256.New, []byte(key))
    hashed.Write([]byte(s))
    return string(hashed.Sum(nil))
}

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

    // step 1: build canonical request string
    httpRequestMethod := "POST"
    canonicalURI := "/"
    canonicalQueryString := ""
    canonicalHeaders := fmt.Sprintf("content-type:%s\nhost:%s\nx-tc-action:%s\n",
        "application/json; charset=utf-8", host, strings.ToLower(action))
    signedHeaders := "content-type;host;x-tc-action"
    payload := `{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}`
    hashedRequestPayload := sha256hex(payload)
    canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
        httpRequestMethod,
        canonicalURI,
        canonicalQueryString,
        canonicalHeaders,
        signedHeaders,
        hashedRequestPayload)
    fmt.Println(canonicalRequest)

    // step 2: build string to sign
    date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
    credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
    hashedCanonicalRequest := sha256hex(canonicalRequest)
    string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
        algorithm,
```

```
timestamp,
credentialScope,
hashedCanonicalRequest)
fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm,
secretId,
credentialScope,
signedHeaders,
signature)
fmt.Println(authorization)

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = implode("\n", [
"content-type:application/json; charset=utf-8",
"host:".$host,
"x-tc-action:".strtolower($action),
```

```
""
});
$signedHeaders = implode(";", [
    "content-type",
    "host",
    "x-tc-action",
]);
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmdate("Y-m-d", $timestamp);
$credentialScope = $date."/". $service."/tc3_request";
$hashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

// step 4: build authorization
$authorization = $algorithm
." Credential=". $secretId."/". $credentialScope
.", SignedHeaders=". $signedHeaders.", Signature=". $signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"
.' -H "Content-Type: application/json; charset=utf-8"
.' -H "Host: '.$host.'"
.' -H "X-TC-Action: '.$action.'"
.' -H "X-TC-Timestamp: '.$timestamp.'"
.' -H "X-TC-Version: '.$version.'"
.' -H "X-TC-Region: '.$region.'"
." -d '". $payload.'"";
echo $curl.PHP_EOL;
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'digest'
```

```
require 'json'
require 'time'
require 'openssl'

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\nxc-action:#{action.downcase}\n"
signed_headers = 'content-type;host;x-tc-action'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['未命名'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
  http_request_method,
  canonical_uri,
  canonical_querystring,
  canonical_headers,
  signed_headers,
  hashed_request_payload,
].join("\n")

puts canonical_request

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
  algorithm,
  timestamp.to_s,
  credential_scope,
  hashed_request_payload,
].join("\n")
puts string_to_sign

# ***** 步骤 3: 计算签名 *****
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
```

```
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

# ***** 步骤 4: 拼接 Authorization *****
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=#{signature}"
puts authorization

puts '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: ' + timestamp.to_s + '"' \
+ ' -H "X-TC-Version: ' + version + '"' \
+ ' -H "X-TC-Region: ' + region + '"' \
+ " -d '" + payload + '"
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

public class Application
{
    public static string SHA256Hex(string s)
    {
        using (SHA256 algo = SHA256.Create())
        {
            byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
            StringBuilder builder = new StringBuilder();
            for (int i = 0; i < hashbytes.Length; ++i)
            {
                builder.Append(hashbytes[i].ToString("x2"));
            }
            return builder.ToString();
        }
    }

    public static byte[] HmacSHA256(byte[] key, byte[] msg)
    {
        using (HMACSHA256 mac = new HMACSHA256(key))
        {
            return mac.ComputeHash(msg);
        }
    }

    public static Dictionary<String, String> BuildHeaders(string secretid,
        string secretkey, string service, string endpoint, string region,
        string action, string version, DateTime date, string requestPayload)
    {
        string datestr = date.ToString("yyyy-MM-dd");
```

```
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;
// ***** 步骤 1: 拼接规范请求串 *****

string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

// ***** 步骤 2: 拼接待签名字符串 *****

string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n"
+ requestTimestamp.ToString() + "\n"
+ credentialScope + "\n"
+ hashedCanonicalRequest;
Console.WriteLine(stringToSign);

// ***** 步骤 3: 计算签名 *****

byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);

// ***** 步骤 4: 拼接 Authorization *****

string authorization = algorithm + " "
+ "Credential=" + secretid + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", "
+ "Signature=" + signature;
Console.WriteLine(authorization);

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}

public static void Main(string[] args)
```

```
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

string service = "cvm";
string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";

// 此处由于示例规范的原因, 采用时间戳2019-02-26 00:44:25, 此参数作为示例, 如果在项目中, 您应当使用:
// DateTime date = DateTime.UtcNow;
// 注意时区, 建议此时间统一采用UTC时间戳, 则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}\"";

Dictionary<string, string> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service
, endpoint, region, action, version, date, requestPayload);

Console.WriteLine("POST https://cvm.tencentcloudapi.com");
foreach (KeyValuePair<string, string> kv in headers)
{
Console.WriteLine(kv.Key + ": " + kv.Value);
}
Console.WriteLine();
Console.WriteLine(requestPayload);
}
}
```

NodeJS

```
const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
const hmac = crypto.createHmac('sha256', secret)
return hmac.update(message).digest(encoding)
}

function getHash(message, encoding = 'hex') {
const hash = crypto.createHash('sha256')
return hash.update(message).digest(encoding)
}

function getDate(timestamp) {
const date = new Date(timestamp * 1000)
const year = date.getUTCFullYear()
const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
const day = ('0' + date.getUTCDate()).slice(-2)
return `${year}-${month}-${day}`
}

function main(){
```

```
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

const endpoint = "cvm.tencentcloudapi.com"
const service = "cvm"
const region = "ap-guangzhou"
const action = "DescribeInstances"
const version = "2017-03-12"
//const timestamp = getTime()
const timestamp = 1551113065
//时间处理, 获取世界时间日期
const date = getDate(timestamp)

// ***** 步骤 1: 拼接规范请求串 *****
const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"

const hashedRequestPayload = getHash(payload);
const httpRequestMethod = "POST"
const canonicalUri = "/"
const canonicalQueryString = ""
const canonicalHeaders = "content-type:application/json; charset=utf-8\\n"
+ "host:" + endpoint + "\\n"
+ "x-tc-action:" + action.toLowerCase() + "\\n"
const signedHeaders = "content-type;host;x-tc-action"

const canonicalRequest = httpRequestMethod + "\\n"
+ canonicalUri + "\\n"
+ canonicalQueryString + "\\n"
+ canonicalHeaders + "\\n"
+ signedHeaders + "\\n"
+ hashedRequestPayload
console.log(canonicalRequest)

// ***** 步骤 2: 拼接待签名字符串 *****
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\\n" +
timestamp + "\\n" +
credentialScope + "\\n" +
hashedCanonicalRequest
console.log(stringToSign)

// ***** 步骤 3: 计算签名 *****
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)

// ***** 步骤 4: 拼接 Authorization *****
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
```

```
"Signature=" + signature
console.log(authorization)

const curlcmd = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + '"'
console.log(curlcmd)
}
main()
```

C++

```
#include <algorithm>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

string get_data(int64_t &timestamp)
{
    string utcDate;
    char buff[20] = {0};
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
    utcDate = string(buff);
    return utcDate;
}

string int2str(int64_t n)
{
    std::stringstream ss;
    ss << n;
    return ss.str();
}

string sha256Hex(const string &str)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
```

```
SHA256_Update(&sha256, str.c_str(), str.size());
SHA256_Final(hash, &sha256);
std::string NewString = "";
for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
{
    snprintf(buf, sizeof(buf), "%02x", hash[i]);
    NewString = NewString + buf;
}
return NewString;
}

string HmacSha256(const string &key, const string &input)
{
    unsigned char hash[32];

    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;
    #else
    h = HMAC_CTX_new();
    #endif

    HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
    HMAC_Update(h, ( unsigned char* )&input[0], input.length());
    unsigned int len = 32;
    HMAC_Final(h, hash, &len);

    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX_cleanup(h);
    #else
    HMAC_CTX_free(h);
    #endif

    std::stringstream ss;
    ss << std::setfill('0');
    for (int i = 0; i < len; i++)
    {
        ss << hash[i];
    }

    return (ss.str());
}

string HexEncode(const string &input)
{
    static const char* const lut = "0123456789abcdef";
    size_t len = input.length();

    string output;
    output.reserve(2 * len);
    for (size_t i = 0; i < len; ++i)
    {
        const unsigned char c = input[i];
        output.push_back(lut[c >> 4]);
        output.push_back(lut[c & 15]);
    }
}
```

```
}
return output;
}

int main()
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

string service = "cvm";
string host = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
int64_t timestamp = 1551113065;
string date = get_data(timestamp);

// ***** 步骤 1: 拼接规范请求串 *****
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string lower = action;
std::transform(action.begin(), action.end(), lower.begin(), ::tolower);
string canonicalHeaders = string("content-type:application/json; charset=utf-8\n")
+ "host:" + host + "\n"
+ "x-tc-action:" + lower + "\n";
string signedHeaders = "content-type;host;x-tc-action";
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;
cout << canonicalRequest << endl;

// ***** 步骤 2: 拼接待签名字符串 *****
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;

// ***** 步骤 3: 计算签名 *****
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;

// ***** 步骤 4: 拼接 Authorization *****
```

```
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;

string curlcmd = "curl -X POST https://" + host + "\n"
+ " -H \"Authorization: \" + authorization + "\"\n"
+ " -H \"Content-Type: application/json; charset=utf-8\" + "\n"
+ " -H \"Host: \" + host + "\"\n"
+ " -H \"X-TC-Action: \" + action + "\"\n"
+ " -H \"X-TC-Timestamp: \" + RequestTimestamp + "\"\n"
+ " -H \"X-TC-Version: \" + version + "\"\n"
+ " -H \"X-TC-Region: \" + region + "\"\n"
+ " -d '\" + payload + '\"';
cout << curlcmd << endl;
return 0;
};
```

C

```
#include <ctype.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdint.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

void get_utc_date(int64_t timestamp, char* utc, int len)
{
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(utc, len, "%Y-%m-%d", &sttime);
}

void sha256_hex(const char* str, char* result)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str, strlen(str));
    SHA256_Final(hash, &sha256);
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        strcat(result, buf);
    }
}

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
unsigned char* output, unsigned int* output_len)
{
    HMAC_CTX *h;
```

```
#if OPENSSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

HMAC_Init_ex(h, key, key_len, EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )input, input_len);
HMAC_Final(h, output, output_len);

#if OPENSSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif

}

void hex_encode(const char* input, int input_len, char* output)
{
    static const char* lut = "0123456789abcdef";

    char add_out[128] = {0};
    char temp[2] = {0};
    for (size_t i = 0; i < input_len; ++i)
    {
        const unsigned char c = input[i];
        temp[0] = lut[c >> 4];
        strcat(add_out, temp);
        temp[0] = lut[c & 15];
        strcat(add_out, temp);
    }
    strncpy(output, add_out, 128);
}

void lowercase(const char * src, char * dst)
{
    for (int i = 0; src[i]; i++)
    {
        dst[i] = tolower(src[i]);
    }
}

int main()
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
    const char* service = "cvm";
    const char* host = "cvm.tencentcloudapi.com";
    const char* region = "ap-guangzhou";
    const char* action = "DescribeInstances";
```

```
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

// ***** 步骤 1: 拼接规范请求串 *****
const char* http_request_method = "POST";
const char* canonical_uri = "/";
const char* canonical_query_string = "";
char canonical_headers[100] = {"Content-type:application/json; charset=utf-8\nhost:"};
strcat(canonical_headers, host);
strcat(canonical_headers, "\nx-tc-action:");
char value[100] = {0};
lowercase(action, value);
strcat(canonical_headers, value);
strcat(canonical_headers, "\n");
const char* signed_headers = "content-type;host;x-tc-action";
const char* payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]\"}";
char hashed_request_payload[100] = {0};
sha256_hex(payload, hashed_request_payload);

char canonical_request[256] = {0};
sprintf(canonical_request, "%s\n%s\n%s\n%s\n%s\n%s", http_request_method,
canonical_uri, canonical_query_string, canonical_headers,
signed_headers, hashed_request_payload);
printf("%s\n", canonical_request);

// ***** 步骤 2: 拼接待签名字符串 *****
const char* algorithm = "TC3-HMAC-SHA256";
char request_timestamp[16] = {0};
sprintf(request_timestamp, "%d", timestamp);
char credential_scope[64] = {0};
strcat(credential_scope, date);
sprintf(credential_scope, "%s/%s/tc3_request", date, service);
char hashed_canonical_request[100] = {0};
sha256_hex(canonical_request, hashed_canonical_request);
char string_to_sign[256] = {0};
sprintf(string_to_sign, "%s\n%s\n%s\n%s", algorithm, request_timestamp,
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s", "TC3", SECRET_KEY);
unsigned char k_date[64] = {0};
unsigned int output_len = 0;
hmac_sha256(k_key, strlen(k_key), date, strlen(date), k_date, &output_len);
unsigned char k_service[64] = {0};
hmac_sha256(k_date, output_len, service, strlen(service), k_service, &output_len);
unsigned char k_signing[64] = {0};
hmac_sha256(k_service, output_len, "tc3_request", strlen("tc3_request"), k_signing, &output_len);
unsigned char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, output_len, string_to_sign, strlen(string_to_sign), k_hmac_sha_sign, &output_len);

char signature[128] = {0};
```

```
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);

// ***** 步骤 4: 拼接 Authorization *****
char authorization[512] = {0};
sprintf(authorization, "%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm, SECRET_ID, credential_scope, signed_headers, signature);
printf("%s\n", authorization);

char curlcmd[10240] = {0};
sprintf(curlcmd, "curl -X POST https://%s\n \
-H \"Authorization: %s\"\n \
-H \"Content-Type: application/json; charset=utf-8\"\n \
-H \"Host: %s\"\n \
-H \"X-TC-Action: %s\"\n \
-H \"X-TC-Timestamp: %s\"\n \
-H \"X-TC-Version: %s\"\n \
-H \"X-TC-Region: %s\"\n \
-d '%s'",
host, authorization, host, action, request_timestamp, version, region, payload);
printf("%s\n", curlcmd);
return 0;
}
```

其他语言

- Lua: [GitHub](#)
- Swift: [GitHub](#)
- Dart: [GitHub](#)
- Shell(Bash): [GitHub](#)

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相符合，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）。

签名方法

最近更新時間：2024-12-25 01:43:50

签名方法 v1 简单易用，但是功能和安全性都不如签名方法 v3，推荐使用签名方法 v3。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

推荐使用 API Explorer

</> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

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，单击【新建密钥】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。以下是使用签名方法 v1 生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKID*****
- SecretKey: *****

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	AKID*****
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

这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数。

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' : 'AKID*****',
  '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"，因此格式化后即为 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=AKID*****
&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。

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-gu
angzhou&SecretId=AKID*****&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 = '*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&R
egion=ap-guangzhou&SecretId=AKID*****&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：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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 7RAM2xfNMO9EiVTNmPg06MRnCvQ=，最终得到的签名串请求参数（Signature）为：

7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D，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 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的网络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
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](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

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=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&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，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
        // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID*****
        params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 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
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "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 os
import time

import requests

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

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',
        'InstanceIds.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)
```

Golang

```
package main

import (
    "bytes"
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "fmt"
    "os"
    "sort"
    "strconv"
)

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    params := map[string]string{
        "Nonce": "11886",
        "Timestamp": strconv.Itoa(1465185768),
        "Region": "ap-guangzhou",
        "SecretId": secretId,
        "Version": "2017-03-12",
        "Action": "DescribeInstances",
        "InstanceIds.0": "ins-09dx96dg",
        "Limit": strconv.Itoa(20),
        "Offset": strconv.Itoa(0),
    }

    var buf bytes.Buffer
    buf.WriteString("GET")
    buf.WriteString("evm.tencentcloudapi.com")
    buf.WriteString("/")
    buf.WriteString("?")

    // sort keys by ascii asc order
    keys := make([]string, 0, len(params))
    for k, _ := range params {
        keys = append(keys, k)
    }
    sort.Strings(keys)

    for i := range keys {
        k := keys[i]
        buf.WriteString(k)
        buf.WriteString("=")
        buf.WriteString(params[k])
        buf.WriteString("&")
    }
    buf.Truncate(buf.Len() - 1)
```

```
hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceIds.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($params);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $params as $key => $value ) {
    $signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $params["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($params);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
```

```
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.0' => 'ins-09dx96dg',
  'Limit' => 20,
  'Nonce' => 11886,
  'Offset' => 0,
  'Region' => 'ap-guangzhou',
  'SecretId' => secret_id,
  'Timestamp' => 1465185768, # Time.now.to_i
  'Version' => '2017-03-12',
}
sign = method + endpoint + '/?'
params = []
data.sort.each do |item|
  params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

# require 'net/http'
# uri = URI('https://' + endpoint)
# uri.query = URI.encode_www_form(data)
# p uri
# res = Net::HTTP.get_response(uri)
# puts res.body
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Security.Cryptography;
using System.Text;

public class Application {
  public static string Sign(string signKey, string secret)
  {
    string signRet = string.Empty;
    using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
    {
      byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
      signRet = Convert.ToBase64String(hash);
    }
    return signRet;
  }

  public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
  {
    string retStr = "";
    retStr += requestMethod;
```

```
retStr += requestHost;
retStr += requestPath;
retStr += "?";
string v = "";
foreach (string key in requestParams.Keys)
{
    v += string.Format("{0}={1}&", key, requestParams[key]);
}
retStr += v.TrimEnd('&');
return retStr;
}

public static void Main(string[] args)
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

    string endpoint = "cvm.tencentcloudapi.com";
    string region = "ap-guangzhou";
    string action = "DescribeInstances";
    string version = "2017-03-12";
    double RequestTimestamp = 1465185768; // 时间戳 2019-02-26 00:44:25, 此参数作为示例, 以实际为准
    // long timestamp = ToTimestamp() / 1000;
    // string requestTimestamp = timestamp.ToString();
    Dictionary<string, string> param = new Dictionary<string, string>();
    param.Add("Limit", "20");
    param.Add("Offset", "0");
    param.Add("InstanceIds.0", "ins-09dx96dg");
    param.Add("Action", action);
    param.Add("Nonce", "11886");
    // param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

    param.Add("Timestamp", RequestTimestamp.ToString());
    param.Add("Version", version);

    param.Add("SecretId", SECRET_ID);
    param.Add("Region", region);
    SortedDictionary<string, string> headers = new SortedDictionary<string, string>(param, StringComparer.Ordinal);
    string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
    string sigOutParam = Sign(SECRET_KEY, sigInParam);
    Console.WriteLine(sigOutParam);
}
}
```

NodeJS

```
const crypto = require('crypto');

function get_req_url(params, endpoint){
    params['Signature'] = encodeURIComponent(params['Signature']);
    const url_strParam = sort_params(params)
    return "https://" + endpoint + "/" + "?" + url_strParam.slice(1);
}
```

```
function formatSignString(reqMethod, endpoint, path, strParam){
let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}

function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

function sort_params(params){
let strParam = "";
let keys = Object.keys(params);
keys.sort();
for (let k in keys) {
//k = k.replace(/_/g, '.');
strParam += ("%&" + keys[k] + "=" + params[keys[k]]);
}
return strParam
}

function main(){
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

const endpoint = "cvm.tencentcloudapi.com"
const Region = "ap-guangzhou"
const Version = "2017-03-12"
const Action = "DescribeInstances"
const Timestamp = 1465185768 // 时间戳 2016-06-06 12:02:48, 此参数作为示例, 以实际为准
// const Timestamp = Math.round(Date.now() / 1000)
const Nonce = 11886 // 随机正整数
//const nonce = Math.round(Math.random() * 65535)

let params = {};
params['Action'] = Action;
params['InstanceIds.0'] = 'ins-09dx96dg';
params['Limit'] = 20;
params['Offset'] = 0;
params['Nonce'] = Nonce;
params['Region'] = Region;
params['SecretId'] = SECRET_ID;
params['Timestamp'] = Timestamp;
params['Version'] = Version;

// 1. 对参数排序, 并拼接请求字符串
strParam = sort_params(params)

// 2. 拼接签名原字符串
const reqMethod = "GET";
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
// console.log(strSign)
```

```
// 3. 生成签名串
params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])

// 4. 进行url编码并拼接请求url
// const req_url = get_req_url(params, endpoint)
// console.log(params['Signature'])
// console.log(req_url)
}
main()
```

返回结果

最近更新时间：2024-03-12 01:37:15

云 API 3.0 接口默认返回 JSON 数据，返回非 JSON 格式的接口会在文档中做出说明。返回 JSON 数据时最大限制为 50 MB，如果返回的数据超过最大限制，请求会失败并返回内部错误。请根据接口文档中给出的过滤功能（例如时间范围）或者分页功能，控制返回数据不要过大。

注意：目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为 200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是 200，而不是 401。

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 [腾讯云客服](#) 或 [提交工单](#)，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 [腾讯云客服](#) 或 [提交工单](#)，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。完整的错误码列表请参考本产品“API 文档”目录下的“错误码”页面。

参数类型

最近更新时间：2022-08-10 06:38:17

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

- String: 字符串。
- Integer: 整型，上限为无符号64位整数。SDK 3.0 不同编程语言支持的类型有所差异，建议以所使用编程语言的最大整型定义，例如 Golang 的 `uint64`。
- Boolean: 布尔型。
- Float: 浮点型。
- Double: 双精度浮点型。
- Date: 字符串，日期格式。例如：2022-01-01。
- Timestamp: 字符串，时间格式。例如：2022-01-01 00:00:00。
- Timestamp ISO8601: ISO 8601 是由国际标准化组织（International Organization for Standardization，ISO）发布的关于日期和时间格式的国际标准，对应国标《GB/T 7408-2005数据元和交换格式信息交换日期和时间表示法》。建议以所使用编程语言的标准库进行格式解析。例如：2022-01-01T00:00:00+08:00。
- Binary: 二进制内容，需要以特定协议请求和解析。

密钥相关接口

修改主密钥描述信息

最近更新时间：2025-04-25 01:40:47

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

该接口用于对指定的cmk修改描述信息。对于处于PendingDelete状态的CMK禁止修改。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：UpdateKeyDescription。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Description	是	String	新的描述信息，最大支持1024字节 示例值：NewDescription
KeyId	是	String	需要修改描述信息的CMK ID 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改主密钥描述信息示例

对指定的CMK修改描述信息

输入示例

```
https://kms.tencentcloudapi.com/?Action=UpdateKeyDescription
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&Description=NewDescription
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

修改别名

最近更新时间：2025-04-25 01:40:47

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

用于修改CMK的别名。对于处于PendingDelete状态的CMK禁止修改。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：UpdateAlias。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Alias	是	String	新的别名，1-60个字符或数字的组合 示例值：NewAlias
KeyId	是	String	CMK的全局唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改别名示例

对指定的CMK修改别名。

输入示例

```
https://kms.tencentcloudapi.com/?Action=UpdateAlias
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&Alias=NewAlias
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.AliasAlreadyExists	别名已经存在。
InvalidParameterValue.InvalidAlias	别名格式错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

密文刷新

最近更新时间：2025-04-25 01:40:48

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

使用指定CMK对密文重新加密。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：ReEncrypt。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CiphertextBlob	是	String	需要重新加密的密文 示例值： M1UPpYhWviWGrfn78MLVBG1n5G2H6+pGK5u7xOmoWoCcWpxVKC10AQujzQSLHWBEk-fKVP3WIIgpg8m9LMW4jEkQ==k-8VyKJSv97jE/+mKGeevmUUvCXLYzWs6Zxn3PZoe
DestinationKeyId	否	String	重新加密使用的CMK，如果为空，则使用密文原有的CMK重新加密（若密钥没有轮换则密文不会刷新） 示例值：87ff856e-973c-11ef-947b-525400d834e5
SourceEncryptionContext	否	String	CiphertextBlob 密文加密时使用的key/value对的json字符串。如果加密时未使用，则为空 示例值：{"key":"value"}
DestinationEncryptionContext	否	String	重新加密使用的key/value对的json字符串，如果使用该字段，则返回的新密文在解密时需要填入相同的字符串 示例值：{"key1":"value1"}

3. 输出参数

参数名称	类型	描述
CiphertextBlob	String	重新加密后的密文 示例值： M1UPpYhWviWGrfn78MLVBG1n5G2H6+pGK5u7xOmoWoCcWpxVKC10AQujzQSLHWBE54p3Kd8Qji+7FOPNk-XqqalxTyNKIC1rITRePFGQ==k-8VyKJSv97jE/+mKGeevmUUvCXLYzWs6Zxn3PZoeBW4aPCjYX
KeyId	String	重新加密使用的CMK 示例值：87ff856e-973c-11ef-947b-525400d834e5
SourceKeyId	String	重新加密前密文使用的CMK 示例值：87ff856e-973c-11ef-947b-525400d834e5
ReEncrypted	Boolean	true表示密文已经重新加密。同一个CMK进行重加密，在密钥没有发生轮换的情况下不会进行实际重新加密操作，返回原密文 示例值：true
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题次请求的 RequestId。

4. 示例

示例1 重加密接口示例

对密文进行重新加密。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ReEncrypt
<公共请求参数>

{
  "CiphertextBlob": "M1UPpYhWviWGrfn78MLVBG1n5G2H6+pGK5u7xOmoWoCcWpxVKC1OAQujzQSLHWBE54p3Kd8Qji+7FOPN3nkQeA==k-fKVP3WIlGpg8m9LMW4jEkQ==k-8VyKJSv97jE/+mKGeevmUUvCXLYzWs6Zxn3PZoeBW4aPCjYX",
  "DestinationKeyId": "87ff856e-973c-11ef-947b-525400d834e5",
  "DestinationEncryptionContext": "{\"key1\":\"value1\"}"
}
```

输出示例

```
{
  "Response": {
    "CiphertextBlob": "M1UPpYhWviWGrfn78MLVBG1n5G2H6+pGK5u7xOmoWoCcWpxVKC1OAQujzQSLHWBE54p3Kd8Qji+7FOPN3nkQeA==k-XqqalxTyNKIC1rITRePFGQ==k-8VyKJSv97jE/+mKGeevmUUvCXLYzWs6Zxn3PZoeBW4aPCjYX",
    "KeyId": "87ff856e-973c-11ef-947b-525400d834e5",
    "ReEncrypted": true,
    "RequestId": "6c3281fb-5286-49d5-8aeb-4a4424bc4271",
    "SourceKeyId": "87ff856e-973c-11ef-947b-525400d834e5"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidCiphertext	密文格式错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkDisabled	CMK已被禁用。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

获取主密钥列表

最近更新时间：2025-04-25 01:40:48

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

列出账号下面状态为Enabled， Disabled 和 PendingImport 的CMK KeyId 列表

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：ListKeys。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Offset	否	Integer	含义跟 SQL 查询的 Offset 一致，表示本次获取从按一定顺序排列数组的第 Offset 个元素开始，缺省为0 示例值：0
Limit	否	Integer	含义跟 SQL 查询的 Limit 一致，表示本次获最多获取 Limit 个元素。缺省值为10，最大值为200 示例值：2
Role	否	Integer	根据创建者角色筛选，默认 0 表示用户自己创建的cmk， 1 表示授权其它云产品自动创建的cmk 示例值：0
HsmClusterId	否	String	KMS 高级版对应的 HSM 集群 ID（仅对 KMS 独占版/托管版服务实例有效）。 示例值：cls-hsm-3dfmo9g

3. 输出参数

参数名称	类型	描述
Keys	Array of Key	CMK列表数组 示例值：[{ "KeyId": "515d337d-70b4-11ef-bc87-02ac442a5a1f" }, { "KeyId": "0ed49d3e-69d3-11ef-9841-52540089bc41" }]
TotalCount	Integer	CMK的总数量 示例值：100
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取主密钥列表示例

获取主密钥列表

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: ListKeys
<公共请求参数>

{
  "Offset": 0,
  "Limit": 2,
  "Role": 0,
  "HsmClusterId": "cls-hsm-3df1mo9g"
}
```

输出示例

```
{
  "Response": {
    "Keys": [
      {
        "KeyId": "515d337d-70b4-11ef-bc87-02ac442a5a1f"
      },
      {
        "KeyId": "0ed49d3e-69d3-11ef-9841-52540089bc41"
      }
    ],
    "RequestId": "28c0e45c-c034-4d8b-b006-f018aa88597d",
    "TotalCount": 14
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。

错误码	描述
InvalidParameter	参数错误。
InvalidParameterValue.InvalidHsmClusterId	无效的 HSM 集群 ID。
UnauthorizedOperation	未授权操作。

获取主密钥列表详情

最近更新时间：2025-04-25 01:40:48

1. 接口描述

接口请求域名：`kms.tencentcloudapi.com`。

根据指定Offset和Limit获取主密钥列表详情。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：ListKeyDetail。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Offset	否	Integer	含义跟 SQL 查询的 Offset 一致，表示本次获取从按一定顺序排列数组的第 Offset 个元素开始，缺省为0 示例值：0
Limit	否	Integer	含义跟 SQL 查询的 Limit 一致，表示本次最多获取 Limit 个元素。缺省值为10，最大值为200 示例值：10
Role	否	Integer	根据创建者角色筛选，默认 0 表示用户自己创建的cmk，1 表示授权其它云产品自动创建的cmk 示例值：0
OrderType	否	Integer	根据CMK创建时间排序，0 表示按照降序排序，1表示按照升序排序 示例值：0
KeyState	否	Integer	根据CMK状态筛选，0表示全部CMK，1 表示仅查询Enabled CMK，2 表示仅查询Disabled CMK，3 表示查询PendingDelete 状态的CMK(处于计划删除状态的Key)，4 表示查询 PendingImport 状态的CMK，5 表示查询 Archived 状态的 CMK 示例值：0
SearchKeyAlias	否	String	根据KeyId或者Alias进行模糊匹配查询 示例值：lzctest
Origin	否	String	根据CMK类型筛选，"TENCENT_KMS" 表示筛选密钥材料由KMS创建的CMK，"EXTERNAL" 表示筛选密钥材料需要用户导入的 EXTERNAL类型CMK，"ALL" 或者不设置表示两种类型都查询，大小写敏感。 示例值：ALL
KeyUsage	否	String	根据CMK的KeyUsage筛选，ALL表示筛选全部，可使用的参数为：ALL 或 ENCRYPT_DECRYPT 或 ASYMMETRIC_DECRYPT_RSA_2048 或 ASYMMETRIC_DECRYPT_SM2 或 ASYMMETRIC_SIGN_VERIFY_SM2 或 ASYMMETRIC_SIGN_VERIFY_RSA_2048 或 ASYMMETRIC_SIGN_VERIFY_ECC，为空则默认筛选ENCRYPT_DECRYPT类型 示例值：ASYMMETRIC_SIGN_VERIFY_RSA_2048
TagFilters.N	否	Array of TagFilter	标签过滤条件 示例值：[{"TagKey":"env","TagValue":["dev"]}]]
HsmClusterId	否	String	KMS 高级版对应的 HSM 集群 ID（仅对 KMS 独占版/托管版服务实例有效）。 示例值：cls-hsm-3dfmo9g

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	CMK的总数量

参数名称	类型	描述
		示例值：10
KeyMetadatas	Array of KeyMetadata	返回的属性信息列表。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取主密钥列表详情

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ListKeyDetail
<公共请求参数>

{
  "Offset": 0,
  "Limit": 10,
  "Role": 0,
  "OrderType": 0,
  "KeyState": 0,
  "SearchKeyAlias": "Weijiali",
  "Origin": "TENCENT_KMS",
  "KeyUsage": "ALL",
  "TagFilters": [
    {
      "TagKey": "env",
      "TagValue": [
        "dev"
      ]
    }
  ],
  "HsmClusterId": "cls-hsm-3dflmo9g"
}
```

输出示例

```
{
  "Response": {
    "KeyMetadatas": [
      {
        "Alias": "Weijiali_test_000000005",
        "CreateTime": 1725354067,
        "CreatorUin": 700001028674,
        "DeletionDate": 0,
        "Description": "weijiali_test_00000000000005",
        "HsmClusterId": "cls-hsm-3dflmo9g",
        "KeyId": "0ed49d3e-69d3-11ef-9841-52540089bc41",
        "KeyRotationEnabled": false,
        "KeyState": "Enabled",
        "KeyUsage": "ASYMMETRIC_SIGN_VERIFY_SM2",

```

```
"NextRotateTime": 1756890067,
"Origin": "TENCENT_KMS",
"Owner": "user",
"ResourceId": "creatorUin/700001028674/0ed49d3e-69d3-11ef-9841-52540089bc41",
"Type": 4,
"ValidTo": 0
},
{
  "Alias": "Weijiali_test_00000000000004",
  "CreateTime": 1725354055,
  "CreatorUin": 700001028674,
  "DeletionDate": 0,
  "Description": "Weijiali_test_00000000000004",
  "HsmClusterId": "cls-hsm-3dfmo9g",
  "KeyId": "07ad2031-69d3-11ef-9841-52540089bc41",
  "KeyRotationEnabled": false,
  "KeyState": "Enabled",
  "KeyUsage": "ASYMMETRIC_DECRYPT_SM2",
  "NextRotateTime": 1756890055,
  "Origin": "TENCENT_KMS",
  "Owner": "user",
  "ResourceId": "creatorUin/700001028674/07ad2031-69d3-11ef-9841-52540089bc41",
  "Type": 4,
  "ValidTo": 0
},
{
  "Alias": "weijiali_test_00000000000003",
  "CreateTime": 1725354040,
  "CreatorUin": 700001028674,
  "DeletionDate": 0,
  "Description": "weijiali_test_00000000000003",
  "HsmClusterId": "cls-hsm-3dfmo9g",
  "KeyId": "fea98175-69d2-11ef-910d-525400d834e5",
  "KeyRotationEnabled": false,
  "KeyState": "Enabled",
  "KeyUsage": "ENCRYPT_DECRYPT",
  "NextRotateTime": 1756890040,
  "Origin": "TENCENT_KMS",
  "Owner": "user",
  "ResourceId": "creatorUin/700001028674/fea98175-69d2-11ef-910d-525400d834e5",
  "Type": 4,
  "ValidTo": 0
}
],
"RequestId": "57c12411-dd1a-4f3c-aaff-aec49a442b6c",
"TotalCount": 3
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidHsmClusterId	无效的 HSM 集群 ID。
UnauthorizedOperation	未授权操作。

查询服务状态

最近更新时间：2025-04-25 01:40:49

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

用于查询该用户是否已开通KMS服务

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：GetServiceStatus。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

3. 输出参数

参数名称	类型	描述
ServiceEnabled	Boolean	KMS服务是否开通，true 表示已开通 示例值：1
InvalidType	Integer	服务不可用类型：0-未购买，1-正常，2-欠费停服，3-资源释放 示例值：1
UserLevel	Integer	0-普通版，1-旗舰版 示例值：0
ProExpireTime	Integer	旗舰版到期时间（Epoch Unix Timestamp）。 示例值：1603701385
ProRenewFlag	Integer	旗舰版是否自动续费：0-不自动续费，1-自动续费 示例值：1
ProResourceId	String	旗舰版购买记录的唯一性标识。如果为开通旗舰版，则返回值为空 示例值：kms_pro_12345
ExclusiveVSMEnabled	Boolean	是否开通 KMS 托管版 示例值：false
ExclusiveHSMEnabled	Boolean	是否开通 KMS 独享版 示例值：false
SubscriptionInfo	String	KMS 订阅信息。 示例值：Prepaid_KMS
CmkUserCount	Integer	返回KMS用户密钥使用数量 示例值：10
CmkLimit	Integer	返回KMS用户密钥规格数量 示例值：200
ExclusiveHSMList	Array of ExclusiveHSM	返回独享集群组 示例值：[{"HsmClusterId":"cls-hsm-a1b2c3d4","HsmClusterName":"cls-hsm-a1b2c3d4-

参数名称	类型	描述
		name"},"{"HsmClusterId":"cls-hsm-a1b2cd5","HsmClusterName":"cls-hsm-a1b2c3d5-name"]]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询服务状态

查询服务状态

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: GetServiceStatus
<公共请求参数>

{}
```

输出示例

```
{
  "Response": {
    "ServiceEnabled": true,
    "InvalidType": 1,
    "UserLevel": 0,
    "ProResourceId": "kms_pro_12345",
    "ProRenewFlag": 1,
    "ProExpireTime": 1603701385,
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00",
    "ExclusiveHSMEnabled": false,
    "ExclusiveVSMEnabled": false,
    "SubscriptionInfo": "Prepaid_KMS"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

查询密钥轮换状态

最近更新时间：2025-04-25 01:40:49

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

查询指定的CMK是否开启了密钥轮换功能。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：GetKeyRotationStatus。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
KeyRotationEnabled	Boolean	密钥轮换是否开启 示例值：false
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询密钥轮换状态

查询指定的CMK密钥轮换功能是否开启。

输入示例

```
https://kms.tencentcloudapi.com/?Action=GetKeyRotationStatus
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "KeyRotationEnabled": false,
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

生成数据密钥

最近更新时间：2025-04-25 01:40:49

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

本接口生成一个数据密钥，您可以用这个密钥进行本地数据的加密。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：GenerateDataKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK全局唯一标识符 示例值：93866e69-9755-11ef-8e65-52540089bc41
KeySpec	否	String	指定生成Datakey的加密算法以及Datakey大小，AES_128或者AES_256。KeySpec 和 NumberOfBytes 必须指定一个 示例值：AES_256
NumberOfBytes	否	Integer	生成的DataKey的长度，同时指定NumberOfBytes和KeySpec时，以NumberOfBytes为准。最小值为1，最大值为1024。KeySpec 和 NumberOfBytes 必须指定一个 示例值：32
EncryptionContext	否	String	key/value对的json字符串，如果使用该字段，则返回的DataKey在解密时需要填入相同的字符串 示例值：{"key1":"value1"}
EncryptionPublicKey	否	String	PEM 格式公钥字符串，支持 RSA2048 和 SM2 公钥，用于对返回数据中的 Plaintext 值进行加密。若为空，则不对 Plaintext 值加密。 示例值：-----BEGIN PUBLIC KEY----- MFkwEwYJfJiHQuh2Jr9lMspgK58UVMJTvQCAU+Hztyhd6Aw== -----END PUBLIC KEY-----
EncryptionAlgorithm	否	String	非对称加密算法，配合 EncryptionPublicKey 对返回数据进行加密。目前支持：SM2（以 C1C3C2 格式返回密文），SM2_C1C3C2_ASN1（以 C1C3C2 ASN1 格式返回密文），RSAES_PKCS1_V1_5，RSAES_OAEP_SHA_1，RSAES_OAEP_SHA_256。若为空，则默认为 SM2。 示例值：SM2

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的全局唯一标识 示例值：93866e69-9755-11ef-8e65-52540089bc41
Plaintext	String	若调用时未提供 EncryptionPublicKey，该字段值为生成的数据密钥 DataKey 的 Base64 编码的明文，需进行 Base64 解码以获 若调用时提供了 EncryptionPublicKey，则该字段值为使用 EncryptionPublicKey 公钥进行非对称加密后的 Base64 编码的密文 示例值： q+EouJ/tGeiZlo9/tll2baxQOBFxcN0PNn7F6ElEvPDR6kvQmHohD5PTbUCKPkct6K8jOiYpbuaWZthxco0phMRSE
CiphertextBlob	String	数据密钥DataKey加密后的密文，用户需要自行保存该密文，KMS不托管用户的数据密钥。可以通过Decrypt接口从CiphertextBlob 示例值：GPJUsgImPcSjHKlktXGyzn33yestz+7NteW36M6FuL8hPfJ1xVfdO6Akr8sF69x3yBXMwto4njumJMIsi9W

参数名称	类型	描述
		xAL9z9eXIHV/p+WT1RsHP3dm6f43bkomXjwmvWdMXH+JQoFORTThfXgcRh1f9lPNLENK4+fCOiQVG1VvLdA0F
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时

4. 示例

示例1 生成数据密钥示例

使用指定的CMK生成Datakey。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: GenerateDataKey
<公共请求参数>

{
  "KeyId": "93866e69-9755-11ef-8e65-52540089bc41",
  "KeySpec": "AES_256",
  "NumberOfBytes": 32,
  "EncryptionContext": "{\"key1\":\"value1\"}",
  "EncryptionPublicKey": "-----BEGIN PUBLIC KEY-----\nMFkwEwYHKoZIzj0CAQYIKoEcz***srLVydfJiHQh2Jr9lMspgK58UVMJTvQCAU+Hztyh
d6Aw==\n-----END PUBLIC KEY-----",
  "EncryptionAlgorithm": "SM2"
}
```

输出示例

```
{
  "Response": {
    "CiphertextBlob": "GPJUsGlmPcSjHKIktXGyzn33yestz+7NteW36M6FuL8hPfJ1xVfdO6Akr8sF69x3yBXMwto4njumJMSi9WKow==k-XqqalxTyNKIC
1rITRePFGQ==k-xAL9z9eXIHV/p+WT1RsHP3dm6f43bkomXjwmvWdMXH+JQoFORTThfXgcRh1f9lPNLENK4+fCOiQVG1VvLdA0RvcuRvU=",
    "KeyId": "93866e69-9755-11ef-8e65-52540089bc41",
    "Plaintext": "q+EouJ/tGeiZIo9/tIl2baxQ0BFxcN0PNn7F6EIEvpDR6kvQmHohD5PTbUCKPkct6K8jOiYpbuaWZthxco0phMRSE4+HpB17rX4jmlW8pw3e
HW0Zo8yRyq/c7RVVo0+DtZofszwhMirQyjcBTJWhLt7xywtE5zqhDjngeEktAEw=",
    "RequestId": "044e823a-7a0c-4603-b03c-e99be5df998d"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.EncryptionError	加密操作失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkDisabled	CMK已被禁用。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.KeyPendingDelete	不可用密钥：密钥待删除。
UnauthorizedOperation	未授权操作。

加密

最近更新时间：2025-04-25 01:40:49

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

本接口用于加密最多为4KB任意数据，可用于加密数据库密码，RSA Key，或其它较小的敏感信息。对于应用的数据加密，使用GenerateDataKey生成的DataKey进行本地数据的加解密操作

默认接口请求频率限制：300次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：Encrypt。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	调用CreateKey生成的CMK全局唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01
Plaintext	是	String	被加密的明文数据，该字段必须使用base64编码，原文最大长度支持4K 示例值：dGVzdAo=
EncryptionContext	否	String	key/value对的json字符串，如果指定了该参数，则在调用Decrypt API时需要提供同样的参数，最大支持1024个字符 示例值：{"key1":"value1"}

3. 输出参数

参数名称	类型	描述
CiphertextBlob	String	加密后的密文，base64编码。注意：本字段中打包了密文和密钥的相关信息，不是对明文的直接加密结果，只有将该字段作为Decrypt接口的输入参数，才可以解密出原文。 示例值： GPJUUsGlmPcSjHKIktXGyzn33yestz+7NteW36M6FuL8hPfJ1xVfdO6Akr8sF69x3yBXMwto4njumJMIsi9WKow== k-XqqalxTyNKIC1rITRePFGQ==-k-GqsANzNxvjXhKwJAOY85vxYYm/2RtsCXRq7Q1iWbCAoeAtZ
KeyId	String	加密使用的CMK的全局唯一标识 示例值：93866e69-9755-11ef-8e65-52540089bc41
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 加密接口示例

使用CMK对小于4KB的数据加密。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: Encrypt
<公共请求参数>

{
  "KeyId": "93866e69-9755-11ef-8e65-52540089bc41",
  "Plaintext": "dGVzdAo=",
  "EncryptionContext": "{\"key1\":\"value1\"}"
}
```

输出示例

```
{
  "Response": {
    "CiphertextBlob": "GPJUsGlmPcSjHKIktXGyzn33yestz+7NteW36M6FuL8hPFJ1xVfdO6Akr8sF69x3yBXMwto4njumJMI9WKow==k-XqqalxTyNKIC1rITRePFGQ==k-GqsANzNxvjJXhKwJAOY85vxYYm/2RtsCXRq7Q1iWbCAoeAtZ",
    "KeyId": "93866e69-9755-11ef-8e65-52540089bc41",
    "RequestId": "017f4b2a-4ae8-4d2c-a0d2-ec15e07d3deb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
InvalidParameterValue.InvalidPlaintext	Plaintext不合法。
ResourceUnavailable.CmkArchived	CMK已存档。

错误码	描述
ResourceUnavailable.CmkDisabled	CMK已被禁用。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

批量启动主密钥

最近更新时间：2025-04-25 01:40:50

1. 接口描述

接口请求域名：`kms.tencentcloudapi.com`。

该接口用于批量启用CMK。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：EnableKeys。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyIds.N	是	Array of String	需要批量启用的CMK Id 列表， CMK数量最大支持100 示例值：["23e80852-1e38-11e9-b129-5cb9019b4b02"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 批量启用主密钥示例

对指定的CMK 列表进行批量启用。

输入示例

```
https://kms.tencentcloudapi.com/?Action=EnableKeys
&KeyIds.0=23e80852-1e38-11e9-b129-5cb9019b4b01
&KeyIds.1=23e80852-1e38-11e9-b129-5cb9019b4b02
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.DuplicatedKeyId	KeyId重复。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

开启密钥轮换

最近更新时间：2025-04-25 01:40:50

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

对指定的CMK开启密钥轮换功能。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：EnableKeyRotation。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01
RotateDays	否	Integer	密钥轮转周期，单位天，允许范围 7 ~ 365，默认值 365。 示例值：365

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 启用密钥交换示例

对指定的CMK开启密钥轮换功能。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: EnableKeyRotation
<公共请求参数>

{
  "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidKeyId	KeyId 不合法。
ResourceUnavailable.CmkNotFound	CMK 不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ExternalCmkCanNotRotate	用户导入类型的 CMK 禁止轮换。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

启用主密钥

最近更新时间：2025-04-25 01:40:50

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

用于启用一个指定的CMK。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：EnableKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需提供该次请求的 RequestId。

4. 示例

示例1 启用主密钥示例

启用一个主密钥，允许该 CMK 被使用。

输入示例

```
https://kms.tencentcloudapi.com/?Action=EnableKey
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

批量禁用主密钥

最近更新时间：2025-04-25 01:40:50

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

该接口用于批量禁止CMK的使用。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DisableKeys。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyIds.N	是	Array of String	需要批量禁用的CMK Id 列表，CMK数量最大支持100 示例值：["23e80852-1e38-11e9-b129-5cb9019b4b02\r\n"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 批量禁止CMK

对指定的CMK 列表进行批量禁止。

输入示例

```
https://kms.tencentcloudapi.com/?Action=DisableKeys
&KeyIds.0=23e80852-1e38-11e9-b129-5cb9019b4b01
&KeyIds.1=23e80852-1e38-11e9-b129-5cb9019b4b02
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CmkUsedByCloudProduct	CMK正在被云产品使用。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.DuplicatedKeyId	KeyId重复。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

禁止密钥轮换

最近更新时间：2025-04-25 01:40:50

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

对指定的CMK禁止密钥轮换功能。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DisableKeyRotation。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 禁止密钥交换示例

对指定的CMK禁止密钥交换功能。

输入示例

```
https://kms.tencentcloudapi.com/?Action=DisableKeyRotation
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

禁用主密钥

最近更新时间：2025-04-25 01:40:50

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

本接口用于禁用一个主密钥，处于禁用状态的Key无法用于加密、解密操作。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DisableKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 禁用主密钥示例

禁用指定的CMK。

输入示例

```
https://kms.tencentcloudapi.com/?Action=DisableKey
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CmkUsedByCloudProduct	CMK正在被云产品使用。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

获取多个主密钥属性

最近更新时间：2025-04-25 01:40:50

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

该接口用于批量获取主密钥属性信息。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DescribeKeys。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyIds.N	是	Array of String	查询CMK的ID列表，批量查询一次最多支持100个KeyId 示例值：["87ff856e-973c-11ef-947b-525400d834e5"]

3. 输出参数

参数名称	类型	描述
KeyMetadatas	Array of KeyMetadata	返回的属性信息列表 示例值： { "Alias": "test11lzc", "CreateTime": 1730357927, "CreatorUin": 700001224419, "DeletionDate": 0, "Description": "入外部密钥", "HsmClusterId": "", "KeyId": "93866e69-9755-11ef-8e65-52540089bc41", "KeyRotationEnabled": false }
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取多个主密钥属性

获取多个主密钥属性。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeKeys
<公共请求参数>

{
  "KeyIds": [
    "87ff856e-973c-11ef-947b-525400d834e5",
    "93866e69-9755-11ef-8e65-52540089bc41"
  ]
}
```

```
]
}
```

输出示例

```
{
  "Response": {
    "KeyMetadatas": [
      {
        "Alias": "test8-lzc",
        "CreateTime": 1730347170,
        "CreatorUin": 700001224419,
        "DeletionDate": 0,
        "Description": "test描述",
        "HsmClusterId": "cls-hsm-3dfmo9g",
        "KeyId": "87ff856e-973c-11ef-947b-525400d834e5",
        "KeyRotationEnabled": false,
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "NextRotateTime": 1761883170,
        "Origin": "TENCENT_KMS",
        "Owner": "user",
        "ResourceId": "creatorUin/700001224419/87ff856e-973c-11ef-947b-525400d834e5",
        "Type": 4,
        "ValidTo": 0
      },
      {
        "Alias": "test11-lzc",
        "CreateTime": 1730357927,
        "CreatorUin": 700001224419,
        "DeletionDate": 0,
        "Description": "倒入外部密钥",
        "HsmClusterId": "cls-hsm-3dfmo9g",
        "KeyId": "93866e69-9755-11ef-8e65-52540089bc41",
        "KeyRotationEnabled": false,
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "NextRotateTime": 1761893927,
        "Origin": "EXTERNAL",
        "Owner": "user",
        "ResourceId": "creatorUin/700001224419/93866e69-9755-11ef-8e65-52540089bc41",
        "Type": 4,
        "ValidTo": 0
      }
    ],
    "RequestId": "05278c40-eb6c-473d-a1e6-00a8177f9c6a"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.DuplicatedKeyId	KeyId重复。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

获取主密钥属性

最近更新时间：2025-04-25 01:40:51

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

用于获取指定KeyId的主密钥属性详情信息。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DescribeKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK全局唯一标识符 示例值：93866e69-9755-11ef-8e65-52540089bc41

3. 输出参数

参数名称	类型	描述
KeyMetadata	KeyMetadata	密钥属性信息 示例值： { "Alias": "test11lzc", "CreateTime": 1730357927, "CreatorUin": 700001224419, "DeletionDate": 0, "Description": "入外部密钥", "HsmClusterId": "", "KeyId": "93866e69-9755-11ef-8e65-52540089bc41", "KeyRotationEnabled": false }
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取主密钥属性示例

获取指定CMK的属性详情信息。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeKey
<公共请求参数>

{
  "KeyId": "93866e69-9755-11ef-8e65-52540089bc41"
}
```

输出示例

```
{
  "Response": {
    "KeyMetadata": {
      "Alias": "test11-lzc",
      "CreateTime": 1730357927,
      "CreatorUin": 700001224419,
      "DeletionDate": 0,
      "Description": "倒入外部密钥",
      "HsmClusterId": "cls-hsm-3dfmo9g",
      "KeyId": "93866e69-9755-11ef-8e65-52540089bc41",
      "KeyRotationEnabled": false,
      "KeyState": "Enabled",
      "KeyUsage": "ENCRYPT_DECRYPT",
      "NextRotateTime": 1761893927,
      "Origin": "EXTERNAL",
      "Owner": "user",
      "ResourceId": "creatorUin/700001224419/93866e69-9755-11ef-8e65-52540089bc41",
      "Type": 4,
      "ValidTo": 0
    },
    "RequestId": "5575a949-8eda-497f-bee3-9d8a07763747"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。

错误码	描述
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

解密

最近更新时间：2025-04-25 01:40:51

1. 接口描述

接口请求域名：`kms.tencentcloudapi.com`。

本接口用于解密密文，得到明文数据。

默认接口请求频率限制：300次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：Decrypt。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CiphertextBlob	是	String	待解密的密文数据 示例值： GPJUsgImPcSjHKlktXGyzn33yestz+7NteW36M6FuL8hPfJ1xVfdO6Akr8sF69x3yBXMwto4njumJMk-XqqalxTyNKIC1rITRePFGQ==--k-GqsANzNxvjJXhKwJAOY85vxYYm/2RtsCXRq7Q1iWbCAoeAtZ
EncryptionContext	否	String	key/value对的json字符串，如果Encrypt指定了该参数，则在调用Decrypt API时需要提供同样的参数，最大支持10 示例值：{"key1":"value1"}
EncryptionPublicKey	否	String	PEM 格式公钥字符串，支持 RSA2048 和 SM2 公钥，用于对返回数据中的 Plaintext 值进行加密。若为空，则不对密。 示例值：-----BEGIN PUBLIC KEY----- MFkwEwYHKoZIzj0CAQYIKoEcz1UBgi0DQgAEtXY****ydfJiHQuh2Jr9IMAU+Hztyhd6Aw== -----E KEY-----
EncryptionAlgorithm	否	String	非对称加密算法，配合 EncryptionPublicKey 对返回数据进行加密。目前支持：SM2（以 C1C3C2 格式返回密文） SM2_C1C3C2_ASN1（以 C1C3C2 ASN1 格式返回密文），RSAES_PKCS1_V1_5，RSAES_OAEP_SHA RSAES_OAEP_SHA_256。若为空，则默认为 SM2。 示例值：SM2

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的全局唯一标识 示例值：93866e69-9755-11ef-8e65-52540089bc41
Plaintext	String	若调用时未提供 EncryptionPublicKey，该字段值为 Base64 编码的明文，需进行 Base64 解码以获取明文。 若调用时提供了 EncryptionPublicKey，则该字段值为使用 EncryptionPublicKey 公钥进行非对称加密后的 Base64 编码的密文。 需在 Base64 解码后，使用用户上传的公钥对应的私钥进行进一步解密，以获取明文。 示例值：dGVzdAo=
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 解密接口示例

用于对密文解密

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: Decrypt
<公共请求参数>

{
  "CiphertextBlob": "GPJUSGlmPcSjHKiktXGyzn33yestz+7NteW36M6FuL8hPfJ1xVfdO6Akr8sF69x3yBXMwto4njumJMIsi9WKow==k-XqqalxTyNKIC1rITRePFGQ==k-GqsANzNxvjJXhKwJA0Y85vxYYm/2RtsCXRq7Q1iWbCAoeAtZ",
  "EncryptionContext": "{\"key1\":\"value1\"}",
  "EncryptionAlgorithm": "SM2"
}
```

输出示例

```
{
  "Response": {
    "KeyId": "93866e69-9755-11ef-8e65-52540089bc41",
    "Plaintext": "dGVzdAo=",
    "RequestId": "4c857b16-e977-4086-b33a-532613f03abd"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.EncryptionError	加密操作失败。
InternalError	内部错误。

错误码	描述
InvalidParameter	参数错误。
InvalidParameterValue.InvalidCiphertext	密文格式错误。
ResourceUnavailable.CmkDisabled	CMK已被禁用。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

创建主密钥

最近更新时间：2025-04-25 01:40:51

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

创建用户管理数据密钥的主密钥CMK（Custom Master Key）。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：CreateKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Alias	是	String	作为密钥更容易辨识，更容易被人看懂的别名，不可为空，1-60个字母数字 - _ 的组合，首字符必须为字母或者数字。以 kms- 作为前缀的用于云产品使用，Alias 不可重复。 示例值：test6-lzc
Description	否	String	CMK 的描述，最大1024字节 示例值：描述信息
KeyUsage	否	String	指定key的用途，默认为 "ENCRYPT_DECRYPT" 表示创建对称加解密密钥，其它支持用途“ASYMMETRIC_DECRYPT_RSA_2048”表示创建用于加解密的RSA2048非对称密钥，“ASYMMETRIC_DECRYPT_SM2”表示创建用于加解密的SM2非对称密钥，“ASYMMETRIC_SIGN_VERIFY_SM2”表示创建用于签名验签的SM2非对称密钥，“ASYMMETRIC_SIGN_VERIFY_ECC”表示创建用于签名验签的ECC非对称密钥，“ASYMMETRIC_SIGN_VERIFY_RSA_2048”表示创建用于签名验签的RSA_2048非对称密钥，“ASYMMETRIC_SIGN_VERIFY_ECDSA384”表示创建用于签名验签的 ECDSA384 非对称密钥。完整的密钥用途与算法支持列表可通过 ListAlgorithms 接口获取。 示例值：ENCRYPT_DECRYPT
Type	否	Integer	指定key类型，默认为1，1表示默认类型，由KMS创建CMK密钥，2 表示EXTERNAL 类型，该类型需要用户导入密钥材料，参考 GetParametersForImport 和 ImportKeyMaterial 接口 示例值：1
Tags.N	否	Array of Tag	标签列表 示例值：[{"TagKey":"env","TagValue":"dev"}]
HsmClusterId	否	String	KMS 高级版对应的 HSM 集群 ID（仅对 KMS 独占版/托管版服务实例有效）。 示例值：cluster-123

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的全局唯一标识符 示例值：36ae6a81-973c-11ef-947b-525400d834e5
Alias	String	作为密钥更容易辨识，更容易被人看懂的别名 示例值：test7-lzc
CreateTime	Integer	密钥创建时间，unix时间戳

参数名称	类型	描述
		示例值：1552897190
Description	String	CMK的描述 示例值：这是一个test描述
KeyState	String	CMK的状态 示例值：Enabled
KeyUsage	String	CMK的用途 示例值：ENCRYPT_DECRYPT
TagCode	Integer	标签操作的返回码。0: 成功；1: 内部错误；2: 业务处理错误 示例值：0
TagMsg	String	标签操作的返回信息 示例值：ok
HsmClusterId	String	HSM 集群 ID（仅对 KMS 独占版/托管版服务实例有效） 示例值：cls-hsm-1234abcd
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建主密钥示例

创建用户管理数据密钥的主密钥CMK（Custom Master Key），通过CMK后续可以调用其他接口诸如创建数据密钥、加解密等操作。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateKey
<公共请求参数>

{
  "Description": "test描述",
  "Alias": "test8-lzc",
  "KeyUsage": "ENCRYPT_DECRYPT",
  "Type": 1,
  "Tags": [
    {
      "TagKey": "env",
      "TagValue": "dev"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Alias": "test8-lzc",
    "CreateTime": 1730347170,
    "Description": "test描述",
    "HsmClusterId": "cls-hsm-3dfimo9g",
    "KeyId": "87ff856e-973c-11ef-947b-525400d834e5",
    "KeyState": "Enabled",
```

```
"KeyUsage": "ENCRYPT_DECRYPT",
"RequestId": "c44aba41-015a-45aa-9e55-1a08ac9d1d78",
"TagCode": 0,
"TagMsg": "ok"
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.TaggingError	标签服务错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.AliasAlreadyExists	别名已经存在。
InvalidParameterValue.InvalidAlias	别名格式错误。
InvalidParameterValue.InvalidHsmClusterId	无效的 HSM 集群 ID。
InvalidParameterValue.InvalidKeyUsage	KeyUsage参数错误。
InvalidParameterValue.InvalidType	Type参数错误。
InvalidParameterValue.TagKeysDuplicated	标签键重复。
InvalidParameterValue.TagsNotExisted	标签键或标签值不存在。
LimitExceeded.CmkLimitExceeded	CMK数量已达上限。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。
UnsupportedOperation.UnsupportedKeyUsageInCurrentRegion	加密方式在当前region不支持。

CMK计划删除接口

最近更新时间：2025-04-25 01:40:47

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

CMK计划删除接口，用于指定CMK删除的时间，可选时间区间为[7,30]天

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：ScheduleKeyDeletion。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK的唯一标志 示例值：93866e69-9755-11ef-8e65-52540089bc41
PendingWindowInDays	是	Integer	计划删除时间区间[7,30] 示例值：7

3. 输出参数

参数名称	类型	描述
DeletionDate	Integer	计划删除执行时间 示例值：1559318399
KeyId	String	唯一标志被计划删除的CMK 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 计划删除接口请求示例

指定CMK于累计7天后被删除

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ScheduleKeyDeletion
<公共请求参数>

{
  "PendingWindowInDays": "7",
```

```
"KeyId": "\"23e80852-1e38-11e9-b129-5cb9019b4b01\""
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8e8f23a7-50b2-4c8e-bd23-0a98cb643f88",
    "DeletionDate": 1559318399,
    "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CmkUsedByCloudProduct	CMK正在被云产品使用。
InternalServerError	内部错误。
InvalidParameter.InvalidPendingWindowInDays	计划删除时间参数非法。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkShouldBeDisabled	未被禁用的CMK不能被计划删除。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

取消CMK计划删除操作

最近更新时间：2025-04-25 01:40:51

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

取消CMK的计划删除操作

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：CancelKeyDeletion。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	需要被取消删除的CMK的唯一标志 示例值：7fa07bcf-9753-11ef-947b-525400d834e5

3. 输出参数

参数名称	类型	描述
KeyId	String	唯一标志被取消删除的CMK。 示例值：7fa07bcf-9753-11ef-947b-525400d834e5
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 取消计划删除接口示例

取消对CMK的计划删除

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CancelKeyDeletion
<公共请求参数>

{
  "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8e8f23a7-50b2-4c8e-bd23-0a98cb643f88",
    "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId 不合法。
ResourceUnavailable.CmkNotFound	CMK 不存在。
ResourceUnavailable.CmkNotPendingDelete	CMK 不是计划删除状态不能被执行取消计划删除。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

导入密钥材料

最近更新时间：2025-04-25 01:40:48

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

用于导入密钥材料。只有类型为EXTERNAL 的CMK 才可以导入，导入的密钥材料使用 GetParametersForImport 获取的密钥进行加密。可以为指定的 CMK 重新导入密钥材料，并重新指定过期时间，但必须导入相同的密钥材料。CMK 密钥材料导入后不可以更换密钥材料。导入的密钥材料过期或者被删除后，指定的CMK将无法使用，需要再次导入相同的密钥材料才能正常使用。CMK是独立的，同样的密钥材料可导入不同的 CMK 中，但使用其中一个 CMK 加密的数据无法使用另一个 CMK解密。只有Enabled 和 PendingImport状态的CMK可以导入密钥材料。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：ImportKeyMaterial。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EncryptedKeyMaterial	是	String	使用GetParametersForImport 返回的PublicKey加密后的密钥材料base64编码。对于国密版本region的KM料长度要求为 256 bit。
ImportToken	是	String	通过调用GetParametersForImport获得的导入令牌。 示例值： +b1d61DykcWNoKUIjj+OQA5M+lu5BiXfxldNy/wDbZL31mB2+SAA29TZgXBKYYbCts9ULktWZx
KeyId	是	String	指定导入密钥材料的CMK，需要和GetParametersForImport 指定的CMK相同。 示例值：93866e69-9755-11ef-8e65-52540089bc41
ValidTo	否	Integer	密钥材料过期时间 unix 时间戳，不指定或者 0 表示密钥材料不会过期，若指定过期时间，需要大于当前时间点，最大示例值：0

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 导入密钥材料

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ImportKeyMaterial
<公共请求参数>
```

```
{
  "ValidTo": "0",
  "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01",
  "ImportToken": "Sy+GF4f+XxUan1sfSBfqWQmyJeVO30wcqLEMoW4REpxBApJqYG9zc37OM0Z3qT/mvLofzM3hwVJSTzIITcAo6ew3LidOJ3nwNoVNOM2wnrL+g9puwP0GpbbSwTd16X9E",
  "EncryptedKeyMaterial": "dgqtAzLde+xsR5q3kwOsWu4bst7FaE4sbhhf1OHCmujhjTJTnFGEubKowVFnFC96e2vUvO5fwGvfNcPIDkg6q7iuFviM0BepCmF9Fyk3tluJVRf/Nv8FCccANXmsZDPxtSHdcD00/dQg1i2BIaLMP3/VFBDVEK7AzRh1TkrNiout4pavGSCAF+uB12Vmnq6fF4DT37B0Q0WQ4NyfUwDCSR0VSF9vwgommhudJUuhv4+21xHMgUCdbFvVoOmZP/Md5Sb4rbMbWi5qRpLusi8W3KKTyJ8onp7UC+0qi6DTw4XCq4UPniFNJXIxUOSVXpZHQKFCJXr6/cxNAci8eMTTwfA=="
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.DecryptMaterialError	解密EncryptedKeyMaterial失败。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
InvalidParameterValue.MaterialNotMatch	导入的密钥材料和历史导入不同。
ResourceUnavailable.CmkNotFound	CMK不存在。

错误码	描述
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
ResourceUnavailable.TokenExpired	Token已过期。
UnsupportedOperation.NotExternalCmk	CMK类型错误，仅支持External CMK。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

获取导入主密钥（CMK）材料的参数

最近更新时间：2025-04-25 01:40:49

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

获取导入主密钥（CMK）材料的参数，返回的Token作为执行ImportKeyMaterial的参数之一，返回的PublicKey用于对自主导入密钥材料进行加密。返回的Token和PublicKey 24小时后失效，失效后如需重新导入，需要再次调用该接口获取新的Token和PublicKey。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：GetParametersForImport。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK的唯一标识，获取密钥参数的CMK必须是EXTERNAL类型，即在CreateKey时指定Type=2 类型的CMK。 示例值：1ff78c0d-c54b-11e9-9cc9-5254006d0810
WrappingAlgorithm	是	String	指定加密密钥材料的算法，目前支持RSAES_PKCS1_V1_5、RSAES_OAEP_SHA_1、RSAES_OAEP_SHA_256 示例值：RSAES_OAEP_SHA_1
WrappingKeySpec	是	String	指定加密密钥材料的类型，目前只支持RSA_2048 示例值：RSA_2048

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的唯一标识，用于指定目标导入密钥材料的CMK。 示例值：1ff78c0d-c54b-11e9-9cc9-5254006d0810
ImportToken	String	导入密钥材料需要的token，用于作为 ImportKeyMaterial 的参数。 示例值： Sy+GF4f+XxUan1sfSBfqWQmyJeVO30wcqLEMoW4REpw3adjEFXyCP2yqzV8xdD5GMY4glZoDfJ33Snnb
PublicKey	String	用于加密密钥材料的RSA公钥，base64编码。使用PublicKey base64解码后的公钥将导入密钥进行加密后作为 ImportKey
ParametersValidTo	Integer	该导出token和公钥的有效期，超过该时间后无法导入，需要重新调用GetParametersForImport获取。 示例值：1566614716
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位

4. 示例

示例1 获取导入主密钥（CMK）材料的参数

输入示例

```
https://kms.tencentcloudapi.com/?Action=GetParametersForImport
&KeyId=1ff78c0d-c54b-11e9-9cc9-5254006d0810
&WrappingAlgorithm=RSAES_OAEP_SHA_1
&WrappingKeySpec=RSA_2048
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "KeyId": "1ff78c0d-c54b-11e9-9cc9-5254006d0810",
    "ImportToken": "Sy+GF4f+XxUan1sfSBFqWQmyJeVO30wcqLEmOW4REpw3adjEFXyCP2yqzV8xdD5GM4gIZoDfJ33SnnbxDMRND81zh4mZjzFNM8PsjhYrg
VSxIiJOCHupZvD4QcoGco8",
    "PublicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAI1jpYRubNKn97lPU0/eR8UBaV11glfQLNqlub+YspXwBsvVqcnSvu5c8GoLTEYGS
TVbvsREDCq6VfFoSeoIqbw8aWukxbPGWgI1nu55ocvWwzhzROFA5UPGC8LOuH0ftZ5Z7cIaigEenS3ngfo4MYTJtg6/B11jJVWvjnKzgswsZeFclSUrEdSxUcm
xfSI344s6I17DTNAQ/vQqyjFGIyo2+JctaxVlY2XuBZf7tPimNdoxAoJ14QxAl1gQGu959xnRJ4rwZbxsklJnBivQqTBeFIiv3KTzFJS6bkz2eqRJ1p4jTBDWb
HEWTVt6tdXPj4+4D21RFGAt3706vf4PIrwIDAQAB",
    "ParametersValidTo": 1566614716,
    "RequestId": "5e137679-519f-409f-9a99-579a034cc320"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。

错误码	描述
ResourceUnavailable.CmkNotFound	CMK不存在。
UnsupportedOperation.NotExternalCmk	CMK类型错误，仅支持External CMK。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

删除导入的密钥材料

最近更新时间：2025-04-25 01:40:51

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

用于删除导入的密钥材料，仅对EXTERNAL类型的CMK有效，该接口将CMK设置为PendingImport 状态，并不会删除CMK，在重新进行密钥导入后可继续使用。彻底删除CMK请使用 ScheduleKeyDeletion 接口。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DeleteImportedKeyMaterial。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	指定需要删除密钥材料的EXTERNAL CMK。 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除导入的密钥材料

输入示例

```
https://kms.tencentcloudapi.com/?Action=DeleteImportedKeyMaterial
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CmkUsedByCloudProduct	CMK正在被云产品使用。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnsupportedOperation.NotExternalCmk	CMK类型错误，仅支持External CMK。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

随机数生成接口

最近更新时间：2025-04-25 01:40:49

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

随机数生成接口。

默认接口请求频率限制：150次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：GenerateRandom。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
NumberOfBytes	是	Integer	生成的随机数的长度。最小值为1， 最大值为1024。 示例值：16

3. 输出参数

参数名称	类型	描述
Plaintext	String	生成的随机数的明文，该明文使用base64编码，用户需要使用base64解码得到明文。 示例值：17mvQgMl0sbM7OAJ7fpLsA==
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 随机数生成接口示例

生成随机数

输入示例

```
https://kms.tencentcloudapi.com/?Action=GenerateRandom
&NumberOfBytes=16
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Plaintext": "17mvQgMl0sbM7OAJ7fpLsA==",
    "RequestId": "6010cd3d-a85a-4e00-b37b-22606d017420"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
UnauthorizedOperation	未授权操作。

列出当前Region支持的加密方式

最近更新时间：2025-04-25 01:40:48

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

列出当前Region支持的加密方式

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：ListAlgorithms。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

3. 输出参数

参数名称	类型	描述
SymmetricAlgorithms	Array of AlgorithmInfo	本地区支持的对称加密算法 示例值：[{ "Algorithm": "SM4", "KeyUsage": "ENCRYPT_DECRYPT" }]
AsymmetricAlgorithms	Array of AlgorithmInfo	本地区支持的非对称加密算法
AsymmetricSignVerifyAlgorithms	Array of AlgorithmInfo	本地区支持的非对称签名验签算法
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 列出当前Region支持的加密方式

输入示例

```
https://kms.tencentcloudapi.com/?Action=ListAlgorithms
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "460fe7ff-f4db-4509-9b20-208badcfb915",
    "SymmetricAlgorithms": [
      {
        "KeyUsage": "ENCRYPT_DECRYPT",
```

```
"Algorithm": "SM4"
},
],
"AsymmetricAlgorithms": [
{
"KeyUsage": "ASYMMETRIC_DECRYPT_SM2",
"Algorithm": "SM2"
},
{
"KeyUsage": "ASYMMETRIC_DECRYPT_RSA_2048",
"Algorithm": "RSA_2048"
}
],
"AsymmetricSignVerifyAlgorithms": [
{
"KeyUsage": "ASYMMETRIC_SIGN_VERIFY_SM2",
"Algorithm": "SM2"
}
]
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
UnauthorizedOperation	未授权操作。

解绑CMK和云资源的关联关系

最近更新时间：2025-04-25 01:40:47

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

删除指定（key, 资源，云产品）的记录，以表明：指定的云产品的资源已不再使用当前的key。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：UnbindCloudResource。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	cmk的ID 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01
ProductId	是	String	云产品的唯一性标识符 示例值：ssm
ResourceId	是	String	资源/实例ID，由调用方根据自己的云产品特征来定义，以字符串形式做存储。 示例值：resourceId

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 解绑CMK和云资源的关联关系

输入示例

```
https://kms.tencentcloudapi.com/?Action=UnbindCloudResource
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&ProductId=ssm
&ResourceId=resourceId
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "fe11aa29-0cc2-4204-bfea-6ebb30cc00d7"
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CloudResourceBindingNotFound	cmk和云资源的绑定关系不存在。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

绑定密钥和云产品资源的使用关系

最近更新时间：2025-04-25 01:40:51

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

记录当前key被哪个云产品的那个资源所使用。如果当前key设置了自动过期，则取消该设置，确保当前key不会自动失效。如果当前关联关系已经创建，也返回成功。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：BindCloudResource。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	cmk的ID 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01
ProductId	是	String	云产品的唯一性标识符 示例值：ssm
ResourceId	是	String	资源/实例ID，由调用方根据自己的云产品特征来定义，以字符串形式做存储。 示例值：resourceId

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 绑定密钥和云产品资源的使用关系

绑定密钥和云产品资源的使用关系

输入示例

```
https://kms.tencentcloudapi.com/?Action=BindCloudResource
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&ProductId=ssm
&ResourceId=resourceId
&<公共请求参数>
```

输出示例

```
{
  "Response": {
```

```
"RequestId": "fe11aa29-0cc2-4204-bfea-6ebb30cc00d7"
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CmkUsedByCloudProduct	CMK正在被云产品使用。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

获取可以提供KMS服务的地域列表

最近更新时间：2025-04-25 01:40:49

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

获取可以提供KMS服务的地域列表

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：GetRegions。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

3. 输出参数

参数名称	类型	描述
Regions	Array of String	可用region列表 示例值：["ap-guangzhou", "ap-hongkong", "ap-taipei", "ap-shanghai", "ap-jakarta"]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询可用区域列表

输入示例

```
https://kms.tencentcloudapi.com/?Action=GetRegions
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Regions": [
      "ap-hongkong",
      "ap-guangzhou"
    ],
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。

取消密钥归档

最近更新时间：2025-04-25 01:40:51

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

取消密钥归档，取消后密钥的状态变为Enabled。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：CancelKeyArchive。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 取消密钥归档

输入示例

```
https://kms.tencentcloudapi.com/?Action=CancelKeyArchive
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.NotUserCreatedCmk	仅支持对用户自己创建的CMK做更新。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

密钥归档

最近更新时间：2025-04-25 01:40:52

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

对密钥进行归档，被归档的密钥只能用于解密，不能加密

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：ArchiveKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需提供该次请求的 RequestId。

4. 示例

示例1 密钥归档

输入示例

```
https://kms.tencentcloudapi.com/?Action=ArchiveKey
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CmkUsedByCloudProduct	CMK正在被云产品使用。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.NotUserCreatedCmk	仅支持对用户自己创建的CMK做更新。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。

后量子密码算法加密

最近更新时间：2025-04-25 01:40:48

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

本接口使用后量子密码算法密钥，可加密最多为4KB任意数据，可用于加密数据库密码，RSA Key，或其它较小的敏感信息。对于应用的数据加密，使用GenerateDataKey生成的DataKey进行本地数据的加解密操作。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：PostQuantumCryptoEncrypt。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	调用CreateKey生成的CMK全局唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01
PlainText	是	String	被加密的明文数据，该字段必须使用base64编码，原文最大长度支持4K 示例值：dGVzdAo=

3. 输出参数

参数名称	类型	描述
CiphertextBlob	String	加密后的密文，base64编码。注意：本字段中打包了密文和密钥的相关信息，不是对明文的直接加密结果，只有将该字段作为PostQuantumCryptoDecrypt接口的输入参数，才可以解密出原文。 示例值：ZHNhZHNpbzEyMW5uaz1qb2lqMm4xbW5tc2FkYXMxO*TlxbWExcwo=
KeyId	String	加密使用的CMK的全局唯一标识 示例值：9999aed0-4956-11e9-bc70-5254005e86b4
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 使用后量子密码算法密钥进行数据加密

使用后量子密码算法密钥进行数据加密。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: PostQuantumCryptoEncrypt
<公共请求参数>
```

```
{
  "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01",
  "PlainText": "dGVzdAo="
}
```

输出示例

```
{
  "Response": {
    "CiphertextBlob": "ZHNhZHNpbzEyMW5uaz1qb2lqMm*****TlxWExcwo=",
    "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01",
    "RequestId": "403150d4-1736-4e61-9259-d70bb7799620"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
InvalidParameterValue.InvalidPlaintext	Plaintext不合法。
ResourceUnavailable.CmkArchived	CMK已存档。
ResourceUnavailable.CmkDisabled	CMK已被禁用。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

后量子密码算法解密

最近更新时间：2025-04-25 01:40:48

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

本接口使用后量子密码算法密钥，解密密文，并得到明文数据。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：PostQuantumCryptoDecrypt。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CiphertextBlob	是	String	待解密的密文数据 示例值：Ade234dasdeEWdGVzdCUyMHBsYWlJJIIHL
EncryptionPublicKey	否	String	PEM 格式公钥字符串，支持 RSA2048 和 SM2 公钥，用于对返回数据中的 Plaintext 值进行加密。若为空，则不对 Plaintext 值加密。 示例值：-----BEGIN PUBLIC KEY----- MFkwEwYHKoZIzj0CAQYI**VWhjH2dhziiH/NWCkk3FjlhlwqjmnWhmUTg== -----END PUBLIC KEY-----
EncryptionAlgorithm	否	String	非对称加密算法，配合 EncryptionPublicKey 对返回数据进行加密。目前支持：SM2（以 C1C3C2 格式返回密文），SM2_C1C3C2_ASN1（以 C1C3C2 ASN1 格式返回密文），RSAES_PKCS1_V1_5，RSAES_OAEP_SHA_1，RSAES_OAEP_SHA_256。若为空，则默认为 SM2。 示例值：SM2

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的全局唯一标识 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01
PlainText	String	若调用时未提供 EncryptionPublicKey，该字段值为 Base64 编码的明文，需进行 Base64 解码以获取明文。 若调用时提供了 EncryptionPublicKey，则该字段值为使用 EncryptionPublicKey 公钥进行非对称加密后的 Base64 编码的密文。 需在 Base64 解码后，使用用户上传的公钥对应的私钥进行进一步解密，以获取明文。 示例值：dGVzdCUyMHBsYWlJTlwdGV4dA==
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 使用后量子密码算法解密

使用后量子密码算法解密。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: PostQuantumCryptoDecrypt
<公共请求参数>

{
  "CiphertextBlob": "Ade234dasdeEWdGVzdCUyMHBsYW1JJlIHL",
  "EncryptionPublicKey": "-----BEGIN PUBLIC KEY----- MFkwEwYHKoZIzj0CAQYI*****VWhjH2dhziiH/NWckk3FjIhjIwqjmnWhmUTg== -----E
ND PUBLIC KEY----- ",
  "EncryptionAlgorithm": "SM2"
}
```

输出示例

```
{
  "Response": {
    "KeyId": "23e80852-1e38-11e9-b129-5cb9019b4b01",
    "PlainText": "dGVzdCUyMHBsYW1uJTlwdGV4dA==",
    "RequestId": "cc8f9b3c-620d-4a25-9fb5-decd94357bd8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.EncryptionError	加密操作失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。

错误码	描述
InvalidParameterValue.InvalidCiphertext	密文格式错误。
ResourceUnavailable.CmkDisabled	CMK已被禁用。
ResourceUnavailable.CmkNotFound	CMK不存在。
UnauthorizedOperation	未授权操作。

非对称密钥相关接口

获取非对称密钥的公钥

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

该接口用于获取非对称密钥的公钥信息，可用于本地数据加密或验签。只有处于Enabled状态的非对称密钥才可能获取公钥。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：GetPublicKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK的唯一标识。 示例值：554ef4b3-3071-11ea-a86a-5254006d0810

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的唯一标识。 示例值：554ef4b3-3071-11ea-a86a-5254006d0810
PublicKey	String	经过base64编码的公钥内容。
PublicKeyPem	String	PEM格式的公钥内容。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取非对称密钥的公钥。

输入示例

```
https://kms.tencentcloudapi.com/?Action=GetPublicKey
&KeyId=554ef4b3-3071-11ea-a86a-5254006d0810
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8918bd5b-e189-4e2d-b718-01c9f99acd45",
```

```
"KeyId": "554ef4b3-3071-11ea-a86a-5254006d0810",
"PublicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAz00TVMh28VCjOE5DM1BM8a2qIjgipYByb7WE/GyTRLUiIGDUq44VIM5fNI9nVeIf
2D+4pIiU0s4LUYnLxkGcFKUVWkz3nubUzbNdSHRbNjKhbFyyRGT6YYxqMfvCmNXMA3OE56EmvsWU9VwVXqRPOFTaODCx8bRd+R6O+Aho9GaRfwLKht1X7dochd
Ds9SWC6iybRISgpLEh4tvzcS1BemEuyx5U/X/BoL+sVnSsC/XT8J9w0EvJvHZaBW70hIbBo0lhzWTF8TpL/7ncZZUbt fP4SrAkvKgoEIEfRhv5vh5LfSxiS2zQ
zIShrT6JYqh5IgDIHTdCcPiYmTsk/lmM2wIDAQAB",
"PublicKeyPem": "-----BEGIN PUBLIC KEY-----\nTUlJQk1qQU5CZ2txaGtpRz13MEJBUEUVCUFPQ0FROEFNSU1CQ2dLQ0FRRUF6ME9U\nVnk1oMjhWQ2p
PRTVETWxCTThhMnFJamdpCFlCeWI3V0UvR3lUUKxVaU1HRFVxNDRW\nnSU01Zk5JOW5WZUlmMkQrNHBjaVUwczRMVl1uTHhrR2NGS1VWV2t6M251Y1V6Yk5k\nnU
0hSYk5qS2hiRn15UkdUN11ZeHFNZnZDbU5YTUEzT0U1NkVtdnNXVT1Wd1ZYcVJQ\nnT0ZUYU9EQ3g4Y1JkK1I2TytBaG85R2FSZndMS0h0bFg3ZG9jaGREcz1TV
0M2aXli\nnUk1TZ3BMRWg0dHZ6Y1NsQmVtRXV5eDVVL1gvQm9MK3NWb1NzQy9YVDhKOXcwRXZK\nnVkhaYUJXN09oSWJCT29saHpXVEY4VHBMZduY1paVWJ0Z1A
0U3JBa3ZLZ29FSUVm\nnUmh2NXZoNUxmU3hpUzJ6UXpJU2hyVDZKWXFoNUlnRElVGRDY1BpWW1Uc2svbG1N\nnMndJREFRQUI=\n-----END PUBLIC KEY-----\n"
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。

非对称密钥Sm2解密

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

使用指定的SM2非对称密钥的私钥进行数据解密，密文必须是使用对应公钥加密的。处于Enabled 状态的非对称密钥才能进行解密操作。传入的密文的长度不能超过256字节。

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：AsymmetricSm2Decrypt。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK的唯一标识 示例值：fce1015e-9735-11ef-be86-6abd61583d2d
Ciphertext	是	String	使用PublicKey加密的密文，Base64编码，原始密文格式需要为C1C3C2_ASN1。原始密文长度不能超过256字节。 示例值： MG0CICuRefoqE7g5bpsZNYq+9d909cZl7nFuYFHiDhOqz8GAAiEA4lqmt3nGqpbu0qhbWsStkexXA2cA12nVW

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的唯一标识 示例值：fce1015e-9735-11ef-be86-6abd61583d2d
Plaintext	String	解密后的明文，base64编码 示例值：dGVzdA==
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 非对称密钥Sm2解密

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: AsymmetricSm2Decrypt
<公共请求参数>

{
  "KeyId": "fce1015e-9735-11ef-be86-6abd61583d2d",
```

```
"Ciphertext": "MG0CICuRefoqE7g5bpsZNYq+9d909cZl7nFuYFHiDhOqz8GAAiEA4lqmt3nGqpbu0qhbWsStkexXA2cA12nVWjgruFP5WKkEIG6JwYejm9gEr7CPOOhbirsGhga3W7J6C7kwdmf6VebRBATH1o2G"
}
```

输出示例

```
{
  "Response": {
    "KeyId": "fce1015e-9735-11ef-be86-6abd61583d2d",
    "Plaintext": "dGVzdA==",
    "RequestId": "e2e0945e-7e90-421a-bfc8-64999d789283"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.DecryptError	解密失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId 不合法。
InvalidParameterValue.InvalidKeyUsage	KeyUsage 参数错误。
ResourceUnavailable.CmkNotFound	CMK 不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.UnsupportedKeyUsageInCurrentRegion	加密方式在当前 region 不支持。

非对称密钥RSA解密

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

使用指定的RSA非对称密钥的私钥进行数据解密，密文必须是使用对应公钥加密的。处于Enabled 状态的非对称密钥才能进行解密操作。

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：AsymmetricRsaDecrypt。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	CMK的唯一标识 示例值：554ef4b3-3071-11ea-a86a-5254006d0810
Ciphertext	是	String	使用PublicKey加密的密文，Base64编码
Algorithm	是	String	在使用公钥加密时对应的算法：当前支持RSAES_PKCS1_V1_5、RSAES_OAEP_SHA_1、RSAES_OAEP_SHA_256 示例值：RSAES_OAEP_SHA_256

3. 输出参数

参数名称	类型	描述
KeyId	String	CMK的唯一标识 示例值：edf72a11-69d0-11ef-9841-52540089bc41
Plaintext	String	解密后的明文，base64编码 示例值：dGVzdA==
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 非对称密钥RSA解密

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: AsymmetricRsaDecrypt
<公共请求参数>

{
  "KeyId": "edf72a11-69d0-11ef-9841-52540089bc41",
```

```
"Ciphertext": "vlnJ/RrQE0mXiHGoya/xpKQJHkd5ImvdXug306BreYL60Kb8ucqQ0eurf5mX/mKZDSTGr8FDVK8b4H0opj7xuDQIzegObtvK2VGbXSaihI/fNDkmoWcgBF/V91scYJtrkwoISnvqsxcl9NeAp+H2A6E3vTiDpQK5oVb3OTG39FoYEvfbRNfef+iukiRiqQL7VGSPmx1nThi6Cd4XGBFzB0tvtmP2SnZhZuzryj+jnHVAz2bIuYP8M4OJYkk5muuACWxLGLQj5sh2FqIpbzT4nti4fOD4ieHnf/qCw/9yyiWUED67oelFKaBJEWGq3PT7bGtH4Sxx8SKx2BCBsR95sA==",
"Algorithm": "RSAES_OAEP_SHA_256"
}
```

输出示例

```
{
  "Response": {
    "KeyId": "edf72a11-69d0-11ef-9841-52540089bc41",
    "Plaintext": "dGVzdAo=",
    "RequestId": "8e7c44ba-917c-4e85-a626-a07806e9b7c9"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.DecryptError	解密失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
InvalidParameterValue.InvalidKeyUsage	KeyUsage参数错误。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
UnauthorizedOperation	未授权操作。

验证签名

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

使用非对称密钥验签

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：VerifyByAsymmetricKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	密钥的唯一标识 示例值：d79e97e5-69d1-11ef-bc87-02ac442a5a1f
SignatureValue	是	String	签名值，通过调用KMS签名接口生成 示例值： MEUCIQDunw5PZuSVI6IPX2xxdce6ohrm9Rt8KBqpMr6yogDbdglgVygCOiOs9OVb/frwglpg+43QXMMRr
Message	是	String	消息原文或消息摘要。如果提供的是消息原文，则消息原文的长度（Base64编码前的长度）不超过4096字节。如果提供的是消息摘要，则消息摘要的长度（Base64编码前的长度）必须等于32字节 示例值：dGVzdAo=
Algorithm	是	String	签名算法，支持的算法：SM2DSA，ECC_P256_R1，RSA_PSS_SHA_256，RSA_PKCS1_SHA_256 等。更多支持的算法，请参见 ListAlgorithms 接口进行查询。 示例值：SM2DSA
MessageType	否	String	消息类型：RAW，DIGEST，如果不传，默认为RAW，表示消息原文。 示例值：RAW

3. 输出参数

参数名称	类型	描述
SignatureValid	Boolean	签名是否有效。true：签名有效，false：签名无效。 示例值：true
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 使用非对称密钥验签

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: VerifyByAsymmetricKey
<公共请求参数>

{
  "KeyId": "d79e97e5-69d1-11ef-bc87-02ac442a5a1f",
  "SignatureValue": "MEUCIQDunw5PZuSVI6IPX2xxdce6ohrm9Rt8KBqpMr6yogDbdglgVygCOiOs9OVb/frwglpg+43QXMMRnFKmBd5VNpS4A/o=",
  "Message": "dGVzdAo=",
  "MessageType": "RAW",
```

```
"Algorithm": "SM2DSA"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "020b75ba-22f8-4f53-ab16-46ae5120711a",
    "SignatureValid": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。

签名

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

非对称密钥签名。

注意：只有 KeyUsage 为 ASYMMETRIC_SIGN_VERIFY_SM2、ASYMMETRIC_SIGN_VERIFY_ECC 或其他支持的 ASYMMETRIC_SIGN_VERIFY_\$(ALGORITHM) 的密钥才可以使用签名功能。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：SignByAsymmetricKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Algorithm	是	String	签名算法，支持的算法：SM2DSA, ECC_P256_R1, RSA_PSS_SHA_256, RSA_PKCS1_SHA_256 等。更多支持的算法可通过 ListAlgorithms 接口进行查询。 示例值：SM2DSA
Message	是	String	消息原文或消息摘要。如果提供的是消息原文，则消息原文的长度（Base64编码前的长度）不超过4096字节。如果提供的是消息摘要，消息摘要长度（Base64编码前的长度）必须等于32字节 示例值：dGVzdAo=
KeyId	是	String	密钥的唯一标识 示例值：d79e97e5-69d1-11ef-bc87-02ac442a5a1f
MessageType	否	String	消息类型：RAW, DIGEST，如果不传，默认为RAW，表示消息原文。 示例值：RAW

3. 输出参数

参数名称	类型	描述
Signature	String	签名，Base64编码 示例值： MEYCIQDFunV6ujrE9k4IMOEeH4J0uwENYSf1JH2DI01qIFWKkwlhAMiv6S2s3VsLtMF/+4rDEiF3lyE3zfRGq5B+k7hHf
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提次请求的 RequestId。

4. 示例

示例1 非对称密钥签名

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: SignByAsymmetricKey
```

<公共请求参数>

```
{
  "Algorithm": "SM2DSA",
  "Message": "dGVzdAo=",
  "KeyId": "d79e97e5-69d1-11ef-bc87-02ac442a5a1f",
  "MessageType": "RAW"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "f60b4087-ab02-4c5b-bf50-956e591ccbc4",
    "Signature": "MEYCIQDFunV6ujrE9k4IMOEeH4J0uwENYSf1JH2DI01q1FWKkwIhAMiv6S2s3VsLtMF/+4rDEiF3IyE3zfRGq5B+k7hHfGoQ"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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
AuthFailure	CAM签名/鉴权错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
MissingParameter	缺少参数错误。
ResourceUnavailable.CmkNotFound	CMK不存在。

错误码	描述
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。

后量子密码算法验签

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

使用后量子密码算法密钥对签名进行验证。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：PostQuantumCryptoVerify。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
KeyId	是	String	密钥的唯一标识 示例值：b2b90f78-df57-11ed-979b-525400cf6ac4
SignatureValue	是	String	签名值，通过调用KMS PostQuantumCryptoSign签名接口生成
Message	是	String	Base64 编码的消息原文，消息原文的长度（Base64编码前的长度）不超过4096字节。 示例值：dGVzdAo=

3. 输出参数

参数名称	类型	描述
SignatureValid	Boolean	签名是否有效。true：签名有效，false：签名无效。 示例值：true
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 后量子密码算法验签

后量子密码算法验签。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: PostQuantumCryptoVerify
<公共请求参数>

{
  "KeyId": "b2b90f78-df57-11ed-979b-525400cf6ac4",
  "SignatureValue": "wb8u5CM+Mn1Z2t1DCxwT8Omhs1N28rMGDCD78iJJe8FA6e2KmxciM5MVGH6TqFU53qOaJc7f00jR4DTl4Pp1Mcrr6aFyw1bVErnaF8G8q6jBVGZN520+b8H+kFuySy6xWMcEuTKnaZRF1MU566d0TqLXlqntR641oP0X9UuzHvpMdRYKorg5IDPq3x5aW0tRIsyDBG7Pa6X7+XOdE1+uWbzjP4bOC1m+Q/fl3KPdcbICxEuDvfdb1qdR4qBZ4uy5bG7JElk0z+NpKcpxiaFz4RaHJO8o64FFyD2F3A6rrKY5jku4I13kXbklY1W93KKGGnJI9sEaqSQjETeLgWPaQECQBBYpYzD+fYes9tvUbtQuZ18fr12bnmlCvFzpeig17nzznxByFSYGBefNjnuzzTX26zUhy54I+sNijYzJyur/nKXXnX6ltcOLYBDke06bjPaPW73HryD1WesWgrL4m0kKoyIy8TdGDVbI5sGNmSovOBvccMQ7b/jD2DwoVh7dGWJcLiroITUfUQLvKp710GEkXKwTpeH1PFwFVTSz+F5zXAlTpvCF9h3Ao4e/fpB+zwR1Z1KQ3TAeCJXdnE3Qqos83Xd804kQ+Y/17vERw//OJ6YIEWzxkcyCTUbqi/DHJOexpQFMu0pPnt1U32u+NASKqxpYE4EEp62T/rU2kYcn8eoHYCAVmU6mxoNsHsLZnt3DIcxc4V0by/ruSznQ/BPextecELaaKPA2ARN4Tb2aUUft1DYgYW4M/p3TEJxL04LPCMIHPpNp3Gj9stsT4rlwvkw9ZYYMaR4IdBjJhd0Lod71G5HzMiJC1D5KL4U
```

qLIpN8bjfX+22U+gN0sVG91ruOhO3/epnsDPvDzoVtYHqkWX7a3XMhgoNxJxr0/KRFFEh1Ogn250qwXg/czqEz1EN/obCkVkrHLdzU2iIJuJAKUZb4Z0J7G1Yz
nd8GU0traDgeKDEnAukU4L5PDVM4jYdrzAF1oF7SNAyePdnNLnWWp8RJnKfD0CZS/eBawq3vYa9avt3++5kwU+BSVBXiQDmV1VYr1VThEX7PoEYNpSwNvWSX
Vw9rTb1I/ek4painAni+QcWJdJVe+z29J5VDpuvrcSONzlep2zMitocxZRxr6VIs1FQLs7McN66enolxJf1/yIaAKP4KxdB1Dt+P2tH+Mqkl1NhvqCAfwm0f7L
oDFWQ6ecAQ/8Pj9gA1bc0Tpq9wLvumS5qmVmTxySr0p5zvJ5NN71mYKAEUzuT/AyKKPHKT/F6iEJd5eWG4njUL128bjaENuDBD12BLxAwmBXiKCrEqCrkGIbK
d/VQ8KtC7PdCH15/R79Xvnu14tEXckfpULevC8Krfsutf5edDMOy0VossBpiyzBmfDOIcX4jxYpg5dK17vQ9q/+Ms/ORR5SUA+TT0w1lKzRwfr7sNR1D1cgg4+
4Ma8WxZOKxPHEPwC7azKLZYmuWWYx74qdney0yUa2r2KY1x1biPi3Jqk91fGyVxoMVucnW2i0X2rtjvV7KZ2MWDBXOkxmDvs6uWMipPO3X6Co3a6VWpuzT+5v19
qswHIwIEan+ovmpBYm57OZxZThfKJff+Tt1ZAnknO7Q2F7uERQZr8yMM+nOMUN04SNUbavm75NaL/mlnNpSrr0pa6IT9iMOYw/Q6HyVMPWEIz5vBOPtKsIAED/
8coDcvSeN9e/dokzhfSTtZgfEX6m4vGefPL1Lr2u7BwkyJRgmefCq6MWCxPA0U089x8oar0PS2tF5PgW1KI/ypzkVQiehLUC9MbTNQIgmXtdaiGrWibZ80o7Bk
WkFWzQ08tbWK7QbhhxNO3bZKXowkiobFWBREAO3NzL4vRcXI47BModygbTd9KmrOK425hN08rfDeO7PfNWyIAjtPtF/alsk1Jnn4MdHooQDsFJJ5dNfCY+men
l3ukgmujn5WZERj+WXV/7X3JOtajb3CXbYc1nVUMtly5/B7KOMCS3+8uC46i2I2UgXbfKM4D8DmRscxmGtuaxuSyZE/XEGPc0aFsVEHaUuGmupaoUnxGqSrXp0
rR+xmTlYnt95IPkwQmQs8R/PmmdXczksrERD6e5grFqZouMgjZP/kmbE4SS1tBDdZDSQfqq8DCTQCtmjQd5pu/Vl/I2xIy3qUfqLOQE+0z4g4B048DNHimo+w5
W2MbWZAigwCsBwwYIL+xZEizU1rE7znd2/ua3WzYB+cD/DiOPJ4sD1zcOn1Vy2tEM7tIPS09hLa8nYX6qLGUL8eg6BC3GNLr+RR/OkNffuTEBMTac36Dk+W9+l
HD3CgNwtWpzSJUS9pqnvTWkCEG4bieFA1NrBk9zPnI1PFbAd8i1pFgv9sLgZQpEBS7ZRSM4bIPcje2BTvdcB9k2sG9Gj7BQ1RCeIso7mSL04eb4d0B18JuqGhN
1shAcK5ts/n/Di5Jk9aBcteGwZrKAwx9W8EXy9rFFNYuq41RH345C6qV+LcxrZWBEqnSx3jvGFRB+Kru11b3Hfw4nd+pwZZ10/ER3LrvWMq+wWhqPBAqCKNg/M
rc7b1JGUu2us2LHWiPxExQhyiVjJnZanBk47RvdCqEAXND8hx+m1rZkWIYzer0aQCtEnvFpqVoHxKbQKXS7XAY3DX9TF3luCFQgsaXF6IehJafL4yoTeJRl/+
TNKHUr+2DAMu3VL35H6OFbMJVPBiPJaFpVUXrOH31ACP+YymHmVRBdbhVSXBafktjm7myj7AZuCGRcO2wrSzpJ5vR7C1rzgl4q+9Rqt/iwrE5FwtLBRVW9CqD
ozIAfeAhPLMAys8GehIHK7n6AXX3/RLR9RZ30j4Egt5FYCVkU9Bf9iq6s17Hyy++MLH9W8W+N5avdVxaV4JtwJPYyZOTzRWfneLYt/r+2/9MYRTJSPPt4TEtv
c+q7m4eC8750s2aZlvHnxahPeodfYMMhtRFQNUYJ2Xei+CORbMuP03PyWbApjeDf9C27KuONNZX81pPltVQYuoIS8nGJRPksPROX0CD58BP4/FT4VWZ365qnI
AORn0p9HjjjFhOlwAieUYfLcOgZAMRQWOy3ye/110hSx27Vp4LlnXZsIuR2Y34jqT0arKUIcZW75myDvL5iUq3z4RDUvsCG0WESBmVzZlZJtGCOPp4iu2HByZk
I8GsDsGQ/Gi8JqRzD2dQzqSJHX8t/gCp3ZZ2pBbD7YWCkLEYti23h0y+y8sShKJkG1hiJHpyCMFP5x0EnSRkuujoAmLi4pzyVhRgGXKqCagw6p+OWX1rVswEG
PyLvVQzjqQsy1Ae5EmbOnzaIKYlndv8PNYqSN8BbtmX6m0jqrUnKFJdpm8YWMtWfn+HJwV7ONMzcG8R5PygB0CgMxPNZbiSrr/OodI8qInyrUbFnF0j76mBkud
ftUKrMVm+gdvTWSfUCPkdkY/JRfNsxk6BJ44y78RXdR8QswxhW6cbP6S9IuZ2hHT29wKfigK0UanoUACtzQev9CvE8YCaPI6yLKT/JopiJI6u6IzxOhzttsRO1
t+XiXrXNQCNGYWA+6IpIrp8WK+WJB91TWDe0WsKtzzbNzmZkkwB6u5vEsbJs2phv3fZjHa7njzeV7Lb+7QqBRPyEZTw6Hfpjr+zH2PDKiDX0j2PiogYqMB4lrm
anZunP0bHY2Q7ON0IEY5ZHKaHoaJHveL45drtoAMRDI/H0izKyCrgrdWZKjmjaLLcx7qIk4N/CzAdxPiNkYVun0pkZXXuVz1F9axy7RoJozmcAXquYpBvTdxu
xi40oKGNQogBh/7H1nYQPYhb+k8TMv5CTVqj1Xqyu/ujv0BUePUBXMA1xxgNK8WfBYLzYlOH9k3BzMrhwditgtALBUWp1pR62w5xxFNN7pn8GQVD/RQW5PIxDu
4X1FyR3Yx40d9tuvLibUSGDCmsHMcNg60A2S/Nztig6qjFKmF9VAZ5KsiRL8s+HzWMKbA1NDBMJdKjJpeMQwvMkrWgew6UFmRHsvxLGL4hMRe0Y3NylSfdPRLM
MYppM+UpDtemx3Od7CreOikz0VUOj/8EQiY3gqZQB1rPKOk1JKKv0Mf+5L5EBj/bVtm7Hv3SUz1jugREtC3jUW5CQniL26tnHZ9JsCrkKjD7b+7kov+Oycx1l2
o89zjxKqixW+fwFKJmEAILxBBHpPRQb+Kou4ypL8hDZ4wMDHYuHhHiELixMaEJYABaa8xZJU0I9WNmMjNTcNxoqR/iHGLkPRTDqLqcbPtkjF11mKJSJpzq71
WrXSJ5BI29ZwPndqJ8yEV1l48HkoiWtCjVGj7q8oyP4rDNv1osb8U7nNg41UkKzA4qESzcC1QoEsUxraVyMjG2e3sv+A7D6Pxs3hTT4W3sVHwHq+EQUCPBUn0m
YyeH2GYfUCCPEPgoliQVdBng1dyn075nv1gKM72unWxWJwicrPO/VoVy1WKxGTS1qiuhxx66+RfbkX9m4FuGei6Zhw+Gxa5JwRYkRQt9vubOVYFbxr2W6O010
68S5G+IMhVuthQ10/VVHZZw1dPJ5NwkqrOads/xiAo/4dGh0898J627G6sQ1pm44slw9KGc7098FVeFHkRnXdXeAdh+vkm+7kQmogpG5W3691BAHeIMbVvXmGz
ipRSlW/S6OR1DOjGXGHuC1n4PnhnzZVBHqPSuas0BORZvhmLcdbej8efn7IXmEy/swrUY/q9sCjVrTS/aQGUXROQ1Rgjxct3sj08akM5TkBOmEgb3RHTkOM5WK
JABd18IaxPXwWX6+ZYYIbN3/NTxOzdaXqsVeF2LHjGNZrvvzuo47uG9Ivp5ERBapDR6We2PdnImkojvWqQmuVxkQ7dM8qjJqi4qnGrDK1BqM+PKhZcN14dKf4C
GRwQa5KN7h5Rm6TS6DtNinVu+hIe2VTSnyS9YxkJ4z2kbPvkzLrmiJR/ZN21vwFFyrZrb/wIn/S61oBKtf+imWJnLVKArwymQnbAjbYotWNdr2KrlfAaep1wlq
Hvvxg1S6KQXjETg1kCXpy/lhJD8rMmsHVF3ZozmRVnOYFmfT1grfYtKGi+7xW25pmKXXgdZgNz7H3o1qPTIbOty0K8ZCv8zxtncFNjf405qRwW02/OO9OCf79n
zc1CKzN1BLsNnQpk1UR3pfinMuYSednc/ElWH8lfsisUa8ub7rH85SSD37tG3Ufe+bIddsiqYds/mX5GZ9d42m7G3FupviPsVMRvsBuy0POhplIOwlJWtGJvrr
PlqbnC36uiW72GveRqyDCG5SUm9HP3oKFAZrAYf8kbz5M0Sn76T9SX1LUUHCjWHSUnDsi+fL1AHkbouSoKHj8XOYxUrSrKpLbJpREE6oA1N4OPWfttSays7b
9sMWDxgE5VJQx2vsrUfSEX8pTdLxqamCQTbO6eSgoDL5g8NMw3n/FDW5xRfBcJsCxiVxvHAYXoSpEBPgdkS3B12O3Y6wARwsp+VAkrDwgbdkjh4v2xECzVyrPc
TEONmp+AyooFZwh7x1fN//gzv22xtQpEbsayAsJIidibma2gFlC+Bc7NwNHCRy3sTB+qb7I61ZTJDF66yXEI3RvJvoADujdvBDQ92mXBenGi2kUbuUtJe4EF8Y
z/x64f7hfnfytz+OpuqCUvn2Hp6BZevQldvQyS1oiJUCrdnkL/9JR5VtK/IvksngQ60subI8TSGsJq0K1Mh9+dJMaUxPxFTLpcFxy346FrB1s6vX7W9PVY2/NG
mDkjbyppHDDxWAdlIVmg2gOg7Ksm6V14/7aMsRZPcJPFjC2dtytgaiWbSfYxGGVifPtFk7SpJ8VYtU6oT4NH/Mq8Y1Ymg7rXnqCEVXdfeo4KY/C05AU0auF6WZY
n5HWJRSnOiA+uAbRSZVC5ggAiD1eqP+Bb1ymgNpbTYoZKD9jiZiZjKkrPSP9DR/B0e29/NcWp8zsnnpbBRV2x1YtbZ557Y7h8/7gyjd43CZnYxCG3Rr7Q8r76d
XJVMgBrb9P8mVOzDgRZBaYlwEalKBZLeJhm09M0MeUka+6DYwiup6ypPwa7INz1hiKLO5OUBcHWLp7naOmafrOT9YY6cqLEmOFhlm9vtKDM3VbHZ/EaBp2Omxg
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACa8VGiEoKy4=",
"Message": "dGVzdAo="

输出示例

```
{  
  "Response": {  
    "RequestId": "e9be01bf-1bed-43d7-bc27-df198a52eef1",  
    "SignatureValid": true  
  }  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。

后量子密码算法签名

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

使用后量子密码算法签名验签密钥进行签名。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：PostQuantumCryptoSign。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Message	是	String	Base64 编码的消息原文。消息原文的长度（Base64编码前的长度）不超过4096字节。 示例值：dGVzdAo=
KeyId	是	String	密钥的唯一标识 示例值：b2b90f78-df57-11ed-979b-525400cf6ac4

3. 输出参数

参数名称	类型	描述
Signature	String	签名值，Base64编码。可使用 PostQuantumCryptoVerify接口对签名值进行验证。 示例值：MEUCICr/JCV52BqGvl0iYxdZ1eL8zzJjx39mWNV2ZWdLOMvRAiEApo6os3Wj0Tg3
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 后量子密码算法签名

后量子密码算法签名。

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: PostQuantumCryptoSign
<公共请求参数>

{
  "Message": "dGVzdAo=",
  "KeyId": "b2b90f78-df57-11ed-979b-525400cf6ac4"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "5a3e2a73-73c3-480c-9e32-36a263803eec",
    "Signature": "wb8u5CM+mNlZ2t1DCxwT80mhs1N28rMGCDc78iJJe8FA6eZKmxoIM5MVGh6TqFU53qOaJc7fOOjR4DTL4Pp1Mcrr6aFyw1bVernaF8G8q6jB
VGNZ520+b8H+kFuySy6xWmcEuTKnaZRF1MU566d0TqLX1qntR641oP0X9UuzHvpMdRYKorg5IDPq3x5aWOtRIsyDBG7Pa6X7+XoDE1+uWbzjP4bOC1m+Q/f13K
PdcBICxEuDvfdb1qdR4qBZ4uy5bG7JElk0z+NpKcpXiaFz4RaHJO8o64FFyD2F3A6rrKY5jku4I13kXbk1Y1W93KKGGnJI9sEaqSQjETeLgWPaQECQBByPyzD+
fYes9tvUbttQuZ18fr12bnmlCvFzpeigl7nzznxByFSYGBefNjnuzzTX26zUhy54I+sNijYzJyur/nKXXnX6ltcOLYBDke06bjPaPW73HryD1WesWgrL4m0kKo
yIy8TdGDVB15sGNmSOvOBvccMQ7b/jD2DWOvh7dGWJcLiroITUfUQLvKp71OGEkXKwTpeH1PFwFVTSz+F5zXAlTpvcF9h3Ao4e/fpB+zWrlZ1KQQ3TAeCJXdnE
3Qqos83XkD804kQ+Y/17vERw//OJ6YIEWzxkcyoTUbqi/DHJOexpPQFMu0Pnt1U32u+NASKqxpyE4EEp62T/rU2kYcn8eoHYCAVMu6mxoNsHsLZnt3D1cxc4VO
by/ruSznQ/BPextecELaaKPA2ARN4Tb2aUuftLDYgYw4M/p3TEJxLO4LPCMIHPNp3Gj9stsT4rlwvkw9ZYyMaR4IdBjJhd0Lod71G5HzMiJC1D5KL4UqLIpN8
bjfX+22U+gN0sVG91ruOhO3/epnsDPvDzoVtYHqkWX7a3XMhgoNxxJxr0/KRFFEHL0gn250qXg/czqEz1EN/obCKvkRHLdzU2iIJujAkUZb4Z0J7G1Yznd8GGU
OtraDgeKDEnAukU4L5PDVM4jYdrzAF1oF7SNAYePdnNLnWwP8RJNkFDoCZS/eBawg3vYa9avt3++5kwU+BSVBXiQqDmv1VYrlVTHeX7PoeYNpSwNvWSXVw9rTb
1I/ek4painAni+QcwJdjVe+z29J5VDpuvrcSONzlep2zMITocxZRxr6VIs1fQLs7McN66eno1xJf1/yIaAKP4KxdB1Dt+P2tH+Mqk1NnhvqCAfwm0f7LoDFWQ6
ecAQ/8Pj9gA1bc0Tpq9wLvumS5qmVmTxySr0p5zvJ5NN71mYKAEUzuT/AyKKPHKT/F6iEJd5eWG4njUL128bjaENuDBD12BLxAwmBXiKCRcEqCrkGibKd/VQ8KT
c7PDCH15/R79Xvnu14tEXckfpULcvC8Krfsut5edDMOy0VossBpiyzBmfDoiGX4jxYpg5dK17vQ9q/+Ms/ORR5SUA+TT0wIlKzRwfr7sNR1D1cgg4+4Ma8WxZ
OKxPHEPwC7azKLZYmuWWYx74qdny0yUa2r2KY1xlbPi3Jqk91fGyVXoMVucnW2i0X2rtjv7VK2MWDBoxkxmDvs6uWmPiPO3X6Co3a6VWpuzT+5v19qswHIWI
EaN+ovmpBYm57O2xzThfKJf+Tt1ZAnkn0Q72F7uERQZr8yMM+nOMUN04SNUbavm75NaL/mlnNpSrr0pa6IT9iMOyW/Q6HyVMPWEIz5vBopTKsIAEd/8coDevS
eN9e/dokzhfSttgfEX6m4vGefPL1Lr2u7BwkyJRgmefCq6MWCxPA0U089x8oar0PS2tF5PgW1KI/ypzkVQiehLUC9MbTNQjGMXTdaiGrWibZ80o7BkKwFwzOQ
08tbWK7QbhhxN03bZXkOwkioBFWBREAO3NzL4vRcXI47BModygbTd9KmrOK425hN08rfDe07PfNWIYIajTptF/alSk1Jnn4MdHooQDsFJJ5dnfCY+men13ukgmu
jn5WZERJ+WXV/7X3Jotajb3CXbYc1nVUMtly5/B7KOMCS3+8uC46i2I2UgXbfKM4D8mRscxmGtuaxuSyZE/XEGpC0aFsVEHaUcGmupaoUnxGqSrXp0rR+xmTl
Ynt95IPkwmQms8R/PmmdXczksrERD6e5grFqZouMgjZP/kmbE4SSltBDdZDSQf8DCTQcTmjQd5pu/Vl/I2xIy3qUfqLQOE+0z4g4B048DNHimo+w5W2MbWZA
igwCsBwwYIL+xZEizUlR7znd2/ua3WzYB+cD/Di0PJ4sD1zcOn1Vy2tEM7tIPS9oH1A8nYX6qLGUL8eg6BC3GNLr+RR/OkNffuTEBMTac36DK+W9+1HD3CgNw
tWpzSJUS9ppqnvTWkCEG4bieFA1NrBk9zPnI1PFbAd8i1pFgv9sLGzQpEBS7ZrM4bIPcJe2BTvdcB9k2sG9Gj7BQ1RCeIs07mSL04eb4d0B18JuqGhN1shAcK5
ts/n/Di5Jk9aBcteGwZrkAwX9W8EXy9rFFNYuc41RH345C6qV+LcxrZWBEqnSx3jvGFRB+Kru11b3Hfw4nd+pwZ210/ER3LrvWMq+wWhqPBAqCKnG/Mrc7b1JG
Uu2us2W1HViPxEQhyiVjNzJanBk47RvdCqEAXND8hx+m1rzKwIYzer0aQCTEnvFpqVoHxKbQKXS7XAY3DX9TF3luCFQgsaXF6IehJafL4yoTeJrL1/+TNKHUr+
D2AMu3VL35H6OFBMJVPBiPjFaJpVUXrOH31ACP+YymHmvrBdhbVXSXBafktjM7myj7AZuCGRcO2wrSzpJ5vR7Clrzgl4q+9Rqt/iwrE5FwtLBRVW9CqDozIAfeA
hPLMAys8GehIHK7n6AXX3/RLR9RZ30j4Egt5FYCVkU9Bf9iq6s17Hy++MLH9W8W+N5avdVxaV4JtwJPyYzOTzRwfneLYt/r+2/9MYRTJSPPT4TEtvc+q7m4e
C8750s2a21vHnxahPeodfYMMhtRFQNUBEYJ2Xei+CORbMuP03PyWbApjeDf9C27KuONNZX81pPltVQYuoIS8nGJRPksPROX0CD58BP4/FT4VWZ365qnIAOrN0p9
HjjjFhOlwaIEUYfLcOgZAMRQWOy3ye/110hSx27Vp4LlnXZsIuR2Y34jqt0arKUiCZW75myDvL5iUq3z4RDUVsCG0wESBmVZzlzJtGC0Pp4iu2HByZk18GsDsG
Q/GiSjQrZDzDQzgSJHX8t/gCp3ZZZpBbD7YWCKLEYti23h0y+y8sShKhJkG1hiJHPYCMFP5x0EnSRkuujoAmLI4pyzVhrgXGKqCagw6p+OWX1rVswEGPyLvVQz
jqQsy1Ae5EmbOnzaIKY1ndv8PNYqSN8BbtmX6m0jqrUnkFJdpm8YWMtWfn+HJwV7ONMZcG8R5PygB0CGMPXNZbiSrr/OodI8qInyrUbFn0j76mBkudftUKrMV
m+gdvTWSfUCPKdKy/JrfNsxk6BJ44y78RXdr8QswxhW6cbp6S9IuZ2HT29wKfigK0UanoUActzQev9CvE8YCaPI6yLKT/JopiJI6u6IzxOhzttsR01t+XiXxR
NQNGYWA+6IpIrp8WK+WJB91TWDe0WsKtzzbNzmZkkwB6u5vEsbJs2phv3fZjHa7njzEv7Lb+7QqBRPyEZTw6Hfpjr+Z2PDKiDX0j2PiogYqMB4lrmzanUnP0
bHY2Q70N0IEYe5ZHKaHoaJHveL45drtoAMRDI/H0izKyCrgdrWZKjmjaLLcx7qIk4N/CzAdxPiNkYVun0pkZXXuVZlF9axy7RojoZmcAXquYpBvTdxuxi40oKG
NQogBh/7H1nYQPYhb+k8TMv5CTVqj1Xqyu/ujv0BUePubXMA1xxgNK8WfBYLzY1OH9k3BzMrhwditgtALBUWp1pR62w5xxFNN7pn8GQVD/RQW5PIxDu4X1FyR3
Yx40d9tvtLiLUSGDCmsHMcNg60A2S/NztiG6qjFKmF9VAZ5Ksirl8s+HZZWMKbA1NDBMJdKjJpeMQwvMkrWgew6UFMRHsvxLG14hMRoeY3NylSfdPRLLMYppM+U
pDtemx3Od7CreOikz0VUOj/8EQiY3gqQBlrPKOk1KKv0Mf+5L5EBj/bVtm7Hv3SUz1jugREtC3jUW5CQniL26tnHZ9JsCrkKjD7b+7kov+Oycxll2o89zjxK
qixW+fWfKJmEAI1xBBhPPrQb+Kou4ypL8hDZ4WMDHYuHhHiELixMaEJYABaa8zJU0I9WNMmJntCnXOqR/iHGLKPRTDqLqciBptKjF11mJkSjPzq71WrxSJ5B
I29ZwPndQj8yEV1148HkoIwtCjVg7j7q8oyP4rDNv1osb8U7nNg41UkKzA4qESzcC1QoEsUxraVyMjG2e3sv+A7D6Pxs3hTT4W3sVHwHq+EQUCPBUn0mYyeH2GY
fUCCpEPgoliQVdBngldyn075nv1gKM72unWxWJWicrzPO/VoVy1WKxGTS1qiuhxx66+RfbkbX9m4FuGei6Zhw+Gxa5JwRYkRQt9vubOVYbXr2W6001068S5G+I
MhVuthQ10/WVHZZw1dPJ5NwkqrOads/xiAo/4dGh0898J627G6sQ1pm44slw9KGc7098FVeFHkRnXdXeAdh+vkM+7kQmogpG5W3691BAHeIMbVVMGzzipRS1W/
S6OR1DOjGXGHuC1n4PnhnzZVBHqPSuas0BORZvhmLcdbej8efn7IXmEy/swrUY/q9sCjVrTS/aQGUXROQ1Rgjxct3sj08akM5TkEOMegb3RHTkOM5WKJABd18I
axPXwWX6+ZYYIbN3/NTxOzdaXqsVeF2LHjGNZRvvzuo47uG9Ivp5ERBapDR6We2PdnImkojvWqQmuVxkQ7dM8qjJqi4qnGrDK1BqM+PKhZCN14dKf4CGRwQa5K
N7h5Rm6TS6DtNinVu+Hie2VTSnyS9Yxk4J2z2kbPVkzLrmi jr/ZN21vwPFYrZRB/win/S61oBktf+imWJnLVKArwymQnbAjbYotWnDr2Kr1fAaep1wlqHvvxg1S
6KQXjETg1kCXPY/lhJD8rMmsHVF3ZozmRVnOYFmfT1grfYtKgi+7xW25pmKXXgdZgNz7H3o1qPTtB0ty0K8ZCv8zxtncFNj405qRwW02/OO9OCf79nzc1CKzN
1BLsNnQpk1UR3pfInMuYSednc/ElWH81fsisUa8ub7rH85SSD37tG3Ufe+bIdDSiqYds/mX5GZ9d42m7G3FupviPsVMRvsBuy0PohplIOW1JWGTJvrrPlqbnC3
6uiW72GveRqyDCG5Sum9HP3oKFAZrAYf8kbz5M0Sn76T9SX1LUUuHCjWHSUnDsi+fL1AHkbousoKHj8XOYxUrSrpkPLbJPREE6oA1N4OPWFttSays7b9sMWDxg
E5VJQx2vsrUfSEX8pTdLxqMCQTbO6eSgoDL5g8NMw3n/FDW5xRfBcJsCxIvXvHAYXoSPeBpgdKs3B12O3Y6wARwsp+VAKrDWgbdKjh4v2xECZVyrPcTE0Nmp+
AyoofZwh7xFln//gzv22xtQpEbsayAsJidibma2gFlC+Bc7NwNHCRy3sTB+qb7I61ZTJDF66yXEI3RvJvoADuJdvBDQ92mXBEgI2kUbuUtJe4EF8Yz/x64f7h
nfyztz+OpuqCUvn2Hp6BZevQldvQyS1oijUcrdnkL/9JR5VtK/IvksngQ60subI8TSGsJq0K1Mh9+dJMaUXpXFTLpcFxy346FrB1s6vX7W9PVY2/NGmDkjbyPH
DDxWAdlIVmg2gOg7Ksm6V14/7aMsRZPoJPFjc2dtytgaIwbSqYxGGVifPtFk7SpJ8VYtU6oT4NH/Mq8Y1YMg7rXnqCEVXdfeo4KY/C05AU0auF6WZYN5HWJRSn
OiA+uAbRSZVC5ggAiD1eqF+Bb1ymgNpbTYoZKD9jiZiZjKkrPSP9DR/B0e29/NcWp8zsnnpbBRV2x1YtbZ557Y7h8/7gyjd43CNzYxCG3Rr7Q8r76dXJVMgBrb
9P8mVozDgrZBaYlweAlKBZLeJhmO9M0MeUKA+6DYwiup6ypPwa7InZ1hiKLO5OUBcHWLp7naOmafrOT9YY6cqLEmOfhlm9vtKDM3VbHZ/EaBp2OmxgAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAC8VGieOKy4="
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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
AuthFailure	CAM签名/鉴权错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
MissingParameter	缺少参数错误。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。

白盒密钥相关接口

使用白盒密钥进行加密

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

使用白盒密钥进行加密

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：EncryptByWhiteBox。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的：ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥的全局唯一标识符 示例值：738c7598-6a69-11ef-9d7c-525400dc620c
PlainText	是	String	待加密的文本，base64编码，文本的原始长度最大不超过4KB 示例值：dGVzdAo=
InitializationVector	否	String	初始化向量，大小为 16 Bytes，加密算法会使用到，base64编码；如果不传，则由后端服务随机生成。用户需要自行保存该值，作为解密的参数。 示例值：3t+gDx/YmtnJuokqAg+cOw==

3. 输出参数

参数名称	类型	描述
InitializationVector	String	初始化向量，加密算法会使用到，base64编码。如果由调用方在入参中传入，则原样返回。如果调用方没有传入，则后端服务随机生成，并返回 示例值：StE4p5N11zwKybR3REtaag==
CipherText	String	加密后的密文，base64编码 示例值：3t+gDx/YmtnJuokqAg+cOw==
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 使用白盒密钥进行加密

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: EncryptByWhiteBox
```

<公共请求参数>

```
{
  "KeyId": "738c7598-6a69-11ef-9d7c-525400dc620c",
  "InitializationVector": "3t+gDx/YmtnJuokqAg+cOw==",
  "PlainText": "dGVzdAo="
}
```

输出示例

```
{
  "Response": {
    "CipherText": "StE4p5N11zwKybR3REtaag==",
    "InitializationVector": "3t+gDx/YmtnJuokqAg+cOw==",
    "RequestId": "a77183bf-8a95-4167-bd4a-84f02df0d4da"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.KeyDisabled	密钥已被禁用。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。

错误码	描述
UnauthorizedOperation	未授权操作。

批量启用白盒密钥

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

批量启用白盒密钥

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：EnableWhiteBoxKeys。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的：ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId.N	是	Array of String	白盒密钥的全局唯一标识符列表。注意：要确保所有提供的KeyId是格式有效的，没有重复，个数不超过50个，并且都是有效存在的。 示例值：["23e80852-1e38-11e9-b129-5cb9019b4b02\r\n"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 批量启用白盒密钥

输入示例

```
https://kms.tencentcloudapi.com/?Action=EnableWhiteBoxKeys
&KeyId.0=23e80852-1e38-11e9-b129-5cb9019b4b01
&KeyId.1=23e80852-1e38-11e9-b129-5cb9019b4b02
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "850bf779-2249-4995-8c55-b3966daf0a8c"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.DuplicatedKeyId	KeyId重复。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

启用白盒密钥

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

启用白盒密钥

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：EnableWhiteBoxKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥的全局唯一标识符 示例值：23e80852-1e38-11e9-b129-5cb9019b4b01

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需提供该次请求的 RequestId。

4. 示例

示例1 启用白盒密钥

输入示例

```
https://kms.tencentcloudapi.com/?Action=EnableWhiteBoxKey
&KeyId=23e80852-1e38-11e9-b129-5cb9019b4b01
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "850bf779-2249-4995-8c55-b3966daf0a8c"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

批量禁用白盒密钥

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

批量禁用白盒密钥

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DisableWhiteBoxKeys。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId.N	是	Array of String	白盒密钥的全局唯一标识符列表。注意：要确保所有提供的KeyId是格式有效的，没有重复，个数不超过50个，并且都是有效存在的。 示例值：["23e80852-1e38-11e9-b129-5cb9019b4b02\r\n"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 批量禁用白盒密钥

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DisableWhiteBoxKeys
<公共请求参数>

{
  "KeyId": [
    "23e80852-1e38-11e9-b129-5cb9019b4b02",
    "23e80852-1e38-11e9-b129-5cb9019b4b01"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1b580852-1e38-11e9-b129-5cb9019b4b00"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.DuplicatedKeyId	KeyId重复。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

禁用白盒密钥

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

禁用白盒密钥

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DisableWhiteBoxKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥的全局唯一标识符 示例值：244dab8c-6dad-11ea-80c6-5254006d0810

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需提供该次请求的 RequestId。

4. 示例

示例1 禁用白盒密钥

输入示例

```
https://kms.tencentcloudapi.com/?Action=DisableWhiteBoxKey
&KeyId=244dab8c-6dad-11ea-80c6-5254006d0810
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "850bf779-2249-4995-8c55-b3966daf0a8c"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

获取白盒密钥服务状态

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

获取白盒密钥服务状态

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DescribeWhiteBoxServiceStatus。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。

3. 输出参数

参数名称	类型	描述
ServiceEnabled	Boolean	用户的白盒密钥服务是否可用 示例值：true
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取白盒密钥服务状态

输入示例

```
https://kms.tencentcloudapi.com/?Action=DescribeWhiteBoxServiceStatus
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8a5ab931-bb22-4d55-9042-e1ef1a36c941",
    "ServiceEnabled": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

获取白盒密钥列表

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

获取白盒密钥列表

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DescribeWhiteBoxKeyDetails。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的：ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyStatus	否	Integer	过滤条件：密钥的状态，0：disabled，1：enabled 示例值：1
Offset	否	Integer	含义跟 SQL 查询的 Offset 一致，表示本次获取从按一定顺序排列数组的第 Offset 个元素开始，缺省为0 示例值：0
Limit	否	Integer	含义跟 SQL 查询的 Limit 一致，表示本次最多获取 Limit 个元素。缺省值为0, 表示不分页 示例值：10
TagFilters.N	否	Array of TagFilter	标签过滤条件 示例值：[{"TagKey":"env","TagValue":["dev"]}]

3. 输出参数

参数名称	类型	描述
KeyInfos	Array of WhiteboxKeyInfo	白盒密钥信息列表。
TotalCount	Integer	白盒密钥总数。 示例值：4
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取白盒密钥列表

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeWhiteBoxKeyDetails
```

<公共请求参数>

```
{
  "KeyStatus": 1,
  "Offset": 0,
  "Limit": 1,
  "TagFilters": [
    {
      "TagKey": "env",
      "TagValue": [
        "dev"
      ]
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "KeyInfos": [
      {
        "Algorithm": "SM4",
        "Alias": "testKey1",
        "CreateTime": 1730441467,
        "CreatorUin": 700001224419,
        "DecryptKey": "AACAAAA3iht+5vrtKBnOagzrjeMFY8GZnRWIOgNcaKtY4RzWUmt0chcuELNIt6ZADmXSEdfndSWfAKizW1F4rnp2fncVQppWN1DjSwY6F/RcSCmU0/aDx3Yc2UIq/zM1jsLFX29CUBQD2a1pNzwepAl1A5HbxLYnrwVdB8Z364d9CyNd5YhVUKo4a8vmx0KA66SaLFS1zpWNZWAb+kZGrJDWMMRqLYigukAYDPsPYXMAb/MIoW5ywicGiY04vI59J1LVPezyMhi5ydTHuImNEcgYw2Tt07yG7LiDFVkW05iCXXX9/zu+9eXdX93WGz20oDDBwpcFsKMk9tXvVhx7o9TCSgT7VERIQOWQKTH9iDb5RUyRPqPvyV1KEtbhTx/AlnkZBIIdOM411JIvNucskvBS4RaZ/MS+wxhkUYnFjL6GMg4MLepB87QaA7zivBKc",
        "Description": "测试描述信息",
        "DeviceFingerprintBind": false,
        "EncryptKey": "AACAAAA3iht+5vrtKBnOagzrjeMFY8GZnRWIOgNcaKtY4RzWUmt0chcuELNIt6ZADmXSEdfndSWfAKizW1F4rnp2fncVQppWN1DjSwY6F/RcSCmU0/aDx3Yc2UIq/zM1jsLFX29CUBQD2a1pNzwepAl1A5HbxLYnrwVdB8Z364d9CyNd5YhVUKo4a8vmx0KA66SaLFS1zpWNZWAb+kZGrJDWMMRqLYigukAYDPsPYXMAb/MIoW5ywicGiY04vI59J1LVPezyMhi5ydTHuImNEcgYw2Tt07yG7LiDFVkW05iCXXX9/zu+9eXdX93WGz20oDDBwpcFsKMk9tXvVhx7o9TCSgT7VERIQOWQKTH9iDb5RUyRPqPvyV1KEtbhTx/AlnkZBIIdOM411JIvNucskvBS4RaZ/MS+wxhkUYnFjL6GMg4MLepB87QaA7zivBKc",
        "KeyId": "14a9885b-9818-11ef-8d67-5254008ae90d",
        "OwnerUin": 100000007998,
        "ResourceId": "creatorUin/700001224419/14a9885b-9818-11ef-8d67-5254008ae90d",
        "Status": "Enabled"
      }
    ],
    "RequestId": "052e4c07-795b-4aa1-8b8f-b968d3792dea",
    "TotalCount": 4
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

展示白盒密钥的信息

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

展示白盒密钥的信息

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DescribeWhiteBoxKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥的全局唯一标识符 示例值：244dab8c-6dad-11ea-80c6-5254006d0810

3. 输出参数

参数名称	类型	描述
KeyInfo	WhiteboxKeyInfo	白盒密钥信息
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 展示白盒密钥的信息

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeWhiteBoxKey
<公共请求参数>

{
  "KeyId": "5551ef2e-6742-11ef-8133-525400b2f623"
}
```

输出示例

```
{
  "Response": {
```

```

"KeyInfo": {
  "Algorithm": "SM4",
  "Alias": "lzc2",
  "CreateTime": 1725072007,
  "CreatorUin": 700001224419,
  "DecryptKey": "AACAAAAACMD5Wlv***+rt0t6Qgamj/FwS6qYFZWwp/M0wM00KwxUSp8P",
  "Description": "2",
  "DeviceFingerprintBind": false,
  "EncryptKey": "AACAAAAACMD5WlvPb4qY***rt0t6Qgamj/FwS6qYFZWwp/M0wM00KwxUSp8P",
  "KeyId": "5551ef2e-6742-11ef-8133-525400b2f623",
  "OwnerUin": 100000007998,
  "ResourceId": "creatorUin/700001224419/5551ef2e-6742-11ef-8133-525400b2f623",
  "Status": "Enabled"
},
"RequestId": "13b3e0ed-7d5e-4c6c-99ba-b334f02d7573"
}
}
    
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

获取白盒解密密钥

最近更新时间：2025-04-25 01:40:47

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

获取白盒解密密钥

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DescribeWhiteBoxDecryptKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥的全局唯一标识符 示例值：244dab8c-6dad-11ea-80c6-5254006d0810

3. 输出参数

参数名称	类型	描述
DecryptKey	String	白盒解密密钥，base64编码
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取白盒解密密钥

输入示例

```
https://kms.tencentcloudapi.com/?Action=DescribeWhiteBoxDecryptKey
&KeyId=244dab8c-6dad-11ea-80c6-5254006d0810
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8788b5f3-8eaa-48a8-915a-b7c52f57065e",
    "DecryptKey": "AAAAACL4R9T1M3r3/S9XpMbUbWtHJUKbq/tl10Bop7LG90lHpj+DKmOiEE0Rs5rRrMN*****qaf0ot57NzfuWG92oAe9HQ7bDCcfuBN7nXX+QipQIBxD/LVWk7zrHtLD/TJfAAoTuE7QKc/lSTOpI1xkTxJq/AFLd5K9EaBN7kpAkAsponZc20Tpk+KlyWB3NHdQrION9fsxtUBl5tz/16T1j49SR5J2axZ2nnkA3o7L1WIGiv7r3tkd+"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

删除白盒密钥

最近更新时间：2025-04-25 01:40:47

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

删除白盒密钥, 注意：必须先禁用后，才可以删除。

默认接口请求频率限制：200次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DeleteWhiteBoxKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥的全局唯一标识符 示例值：244dab8c-6dad-11ea-80c6-5254006d0810

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除白盒密钥

输入示例

```
https://kms.tencentcloudapi.com/?Action=DeleteWhiteBoxKey
&KeyId=244dab8c-6dad-11ea-80c6-5254006d0810
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "850bf779-2249-4995-8c55-b3966daf0a8c"
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

创建白盒密钥

最近更新时间：2025-04-25 01:40:47

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

创建白盒密钥。密钥个数的上限为 50。

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：CreateWhiteBoxKey。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
Alias	是	String	作为密钥更容易辨识，更容易被人看懂的别名，不可为空，1-60个字母数字 - _ 的组合，首字符必须为字母或者数字。Alias不可重复。 示例值：test-lzcx
Algorithm	是	String	创建密钥所有的算法类型，支持的取值：AES_256,SM4 示例值：SM4
Description	否	String	密钥的描述，最大1024字节 示例值：白盒测试
Tags.N	否	Array of Tag	标签列表 示例值：[{"TagKey":"env","TagValue":"dev"}]

3. 输出参数

参数名称	类型	描述
EncryptKey	String	用于加密的密钥，base64编码 示例值：AACAAAAAMjCKSiLyF9LdkbAh+E1D/+W8C3rXOT8OSNAApXUk71RZ719MOQyKSdVm/grpbPSSNf2ywtJQaycFr
DecryptKey	String	用于解密的密钥，base64编码 示例值：AACAAAAAMjCKSiLyF9LdkbAh+E1D/+W8C3rXOT8OSNAApXUk71RZ719MOQyKSdVm/grpbPSSNf2ywtJ
KeyId	String	白盒密钥的全局唯一标识符 示例值：6df3665b-9807-11ef-8d67-5254008ae90d
TagCode	Integer	标签操作的返回码。0: 成功；1: 内部错误；2: 业务处理错误 示例值：0
TagMsg	String	标签操作的返回信息 示例值：ok
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需

4. 示例

示例1 创建白盒密钥

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateWhiteBoxKey
<公共请求参数>

{
  "Alias": "test-lzcx2",
  "Description": "白盒测试",
  "Algorithm": "SM4"
}
```

输出示例

```
{
  "Response": {
    "DecryptKey": "AACAAAAA0/BswblZOBqs0JY/ST***x9scSvWUFbSG/tGBrMpzuCndKomu/Ig4/G",
    "EncryptKey": "AACAAAAA0/BswblZO***VTtExsaGx9scSvWUFbSG/tGBrMpzuCndKomu/Ig4/G",
    "KeyId": "9fe1e083-9807-11ef-8d67-5254008ae90d",
    "RequestId": "af2946f2-0a41-4d40-9e6e-b6e1e3eb5447",
    "TagCode": 0,
    "TagMsg": "no tags to save"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AliasAlreadyExists	别名已经存在。
InvalidParameterValue.InvalidAlias	别名格式错误。
InvalidParameterValue.TagKeysDuplicated	标签键重复。
InvalidParameterValue.TagsNotExisted	标签键或标签值不存在。
LimitExceeded.KeyLimitExceeded	创建的密钥个数超过限制。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

覆盖指定密钥的设备指纹信息

最近更新时间：2025-04-25 01:40:45

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

覆盖指定密钥的设备指纹信息

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：OverwriteWhiteBoxDeviceFingerprints。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的：ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥ID 示例值：5551ef2e-6742-11ef-8133-525400b2f623
DeviceFingerprints.N	否	Array of DeviceFingerprint	设备指纹列表，如果列表为空，则表示删除该密钥对应的所有指纹信息。列表最大长度不超过200。 示例值：[{"Identity":"0177ebba-5ad6930a***bba2b703-ddfd218bab945f-0c275b7f7d2c8c68","Description":"test2"}]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 覆盖指定密钥的设备指纹信息

输入示例

```
https://kms.tencentcloudapi.com/?Action=OverwriteWhiteBoxDeviceFingerprints
&KeyId=cd850a3d-9b1b-11ea-a96a-5254006d0810
&DeviceFingerprints.0.Identity=c19c024c-2ba1-11b2-a85c-96f970f4a8e1
&DeviceFingerprints.0.Description=desc1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "c8d1f1b9-632a-4c0b-b369-9bcf8e7e6ef9"
```

```
}  
  
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
LimitExceeded.FingerprintsLimitExceeded	设备指纹个数超过限制。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

获取指定密钥的设备指纹列表

最近更新时间：2025-04-25 01:40:46

1. 接口描述

接口请求域名：kms.tencentcloudapi.com。

获取指定密钥的设备指纹列表

默认接口请求频率限制：100次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 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。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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：DescribeWhiteBoxDeviceFingerprints。
Version	是	String	公共参数 ，本接口取值：2019-01-18。
Region	是	String	公共参数 ，详见产品支持的 地域列表 ，本接口仅支持其中的: ap-beijing, ap-guangzhou, ap-hongkong, ap-shanghai。
KeyId	是	String	白盒密钥ID 示例值：cd850a3d-9b1b-11ea-a96a-5254006d0810

3. 输出参数

参数名称	类型	描述
DeviceFingerprints	Array of DeviceFingerprint	设备指纹列表 示例值：[{ "Description": "test1", "Identity": "0177ebba-5ad6930a5*b945f-0c275b7f7d2c8c68" }]
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取指定密钥的设备指纹列表

输入示例

```
POST / HTTP/1.1
Host: kms.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeWhiteBoxDeviceFingerprints
<公共请求参数>

{
  "KeyId": "5551ef2e-6742-11ef-8133-525400b2f623"
}
```

输出示例

```
{
  "Response": {
    "DeviceFingerprints": [
      {
        "Description": "test1",
        "Identity": "0177ebba-5ad6930a52d808***03-ddfd218bab945f-0c275b7f7d2c8c68"
      }
    ],
    "RequestId": "5dc1c93f-8864-4525-a896-f56019aae02b"
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
ResourceNotFound	资源不存在。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
UnauthorizedOperation	未授权操作。

数据结构

最近更新时间：2025-04-17 01:21:26

AlgorithmInfo

算法的名称 和 标识

被如下接口引用：ListAlgorithms。

名称	类型	描述
KeyUsage	String	算法的标识 示例值：ASYMMETRIC_DECRYPT_SM2
Algorithm	String	算法的名称 示例值：SM2

DeviceFingerprint

设备指纹

被如下接口引用：DescribeWhiteBoxDeviceFingerprints, OverwriteWhiteBoxDeviceFingerprints。

名称	类型	必选	描述
Identity	String	是	指纹信息，由设备指纹采集工具采集获得，格式满足正则表达式：^[0-9a-f]{8}[-][0-9a-f]{14}[-][0-9a-f]{14}[-][0-9a-f]{14}[-][0-9a-f]{16}\$ 示例值：0177ebba-5ad6930a52d808-ce5e6fbba2b**75b7f7d2c8c68
Description	String	否	描述信息，如：IP，设备名称等，最大1024字节 示例值：lzctest

ExclusiveHSM

独享版集群

被如下接口引用：GetServiceStatus。

名称	类型	描述
HsmClusterId	String	独享集群Id 示例值：cls-hsm-3dfmo9g
HsmClusterName	String	独享集群名称 示例值：cls-hsm-3dfmo9g-name

Key

返回CMK列表信息

被如下接口引用：ListKeys。

名称	类型	描述
KeyId	String	CMK的全局唯一标识。 示例值：5551ef2e-6742-11ef-8133-525400b2f623

KeyMetadata

CMK属性信息

被如下接口引用：DescribeKey, DescribeKeys, ListKeyDetail。

名称	类型	描述
KeyId	String	CMK的全局唯一标识 示例值：6cdf26d1-44ff-11eb-841c-5254006d0810
Alias	String	作为密钥更容易辨识，更容易被人看懂的别名 示例值：lzctest
CreateTime	Integer	密钥创建时间 示例值：1730357927
Description	String	CMK的描述 示例值：desc
KeyState	String	CMK的状态，取值为：Enabled Disabled PendingDelete PendingImport Archived 示例值：PendingDelete
KeyUsage	String	CMK用途，取值为：ENCRYPT_DECRYPT ASYMMETRIC_DECRYPT_RSA_2048 ASYMMETRIC_DECRYPT_SM2 ASYMMETRIC_SIGN_VERIFY_SM2 ASYMMETRIC_SIGN_VERIFY_RSA_2048 ASYMMETRIC_SIGN_VERIFY_ECC 示例值：ASYMMETRIC_DECRYPT_RSA_2048
Type	Integer	CMK类型，2 表示符合FIPS标准，4表示符合国密标准 示例值：3
CreatorUin	Integer	创建者 示例值：700001224410
KeyRotationEnabled	Boolean	是否开启了密钥轮换功能 示例值：true
Owner	String	CMK的创建者，用户创建的为 user，授权各云产品自动创建的为对应的产品名 示例值：user
NextRotateTime	Integer	在密钥轮换开启状态下，下次轮换的时间 示例值：1761893927
DeletionDate	Integer	计划删除的时间 示例值：0
Origin	String	CMK 密钥材料类型，由KMS创建的为：TENCENT_KMS， 由用户导入的类型为：EXTERNAL 示例值：EXTERNAL
ValidTo	Integer	在Origin为 EXTERNAL 时有效，表示密钥材料的有效日期， 0 表示不过期 示例值：0
ResourceId	String	资源ID，格式：creatorUin/\$creatorUin/\$keyId 示例值：creatorUin/12345/6cdf26d1-44ff-11eb-841c-5254006d0810
HsmClusterId	String	HSM 集群 ID（仅对 KMS 独占版/托管版服务实例有效） 示例值：cls-hsm-1a2b3c4d
RotateDays	Integer	密钥轮转周期（天） 示例值：365
LastRotateTime	Integer	上次乱转时间（Unix timestamp） 示例值：1730716649

Tag

标签键和标签值

被如下接口引用：CreateKey, CreateWhiteBoxKey。

名称	类型	必选	描述
TagKey	String	是	标签键 示例值：env
TagValue	String	是	标签值 示例值：dev

TagFilter

标签过滤器

被如下接口引用：DescribeWhiteBoxKeyDetails, ListKeyDetail。

名称	类型	必选	描述
TagKey	String	是	标签键 示例值：env
TagValue	Array of String	否	标签值 示例值：dev

WhiteboxKeyInfo

白盒密钥信息

被如下接口引用：DescribeWhiteBoxKey, DescribeWhiteBoxKeyDetails。

名称	类型	描述
KeyId	String	白盒密钥的全局唯一标识符 示例值：9971f40f-69d2-11ef-9d7c-525400dc620c
Alias	String	作为密钥更容易辨识，更容易被人看懂的别名，不可为空，1-60个字母数字 - _ 的组合，首字符必须为字母或者数字. 不可重复 示例值：weijiali_testabc_0000000000000
CreatorUin	Integer	创建者 示例值：700001028674
Description	String	密钥的描述信息 示例值：weijiali_test_000000000000001
CreateTime	Integer	密钥创建时间，Unix时间戳 示例值：1725353871
Status	String	白盒密钥的状态，取值为：Enabled Disabled 示例值：Enabled
OwnerUin	Integer	创建者 示例值：100000007998
Algorithm	String	密钥所用的算法类型 示例值：SM4
EncryptKey	String	白盒加密密钥，base64编码 示例值： AACAAAAyBDDg3NeCzKNLdNR/gVvEOgDlXrhXykeO**77c7A0xfr1vlzVPYzyn2CeYYLyagRV/jn2PtQg
DecryptKey	String	白盒解密密钥，base64编码 示例值：AACAAAAyBDDg3NeCzKNLdNR/gVvEOg****vtHhBaKA48+28gLk
ResourceId	String	资源ID，格式：creatorUin/\$creatorUin/\$keyId 示例值：creatorUin/700001028674/9971f40f-69d2-11ef-9d7c-525400dc620c
DeviceFingerprintBind	Boolean	是否有设备指纹与当前密钥绑定 示例值：false

错误码

最近更新时间：2024-12-20 01:37:44

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
ActionOffline	接口已下线。
AuthFailure.InvalidAuthorization	请求头部的 Authorization 不符合腾讯云标准。
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签名方法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 CAM 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（包括参数格式、类型等错误）。
InvalidParameterValue	参数取值错误。
InvalidRequest	请求 body 的 multipart 格式错误。
IpInBlacklist	IP 地址在黑名单中。
IpNotInWhitelist	IP 地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。

错误码	说明
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
RequestLimitExceeded.GlobalRegionUinLimitExceeded	主账号超过频率限制。
RequestLimitExceeded.IPLimitExceeded	IP 限频。
RequestLimitExceeded.UinLimitExceeded	主账号限频。
RequestSizeLimitExceeded	请求包超过限制大小。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResponseSizeLimitExceeded	返回包超过限制大小。
ServiceUnavailable	当前服务暂时不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误。
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
AuthFailure	CAM签名/鉴权错误。
FailedOperation.CmkUsedByCloudProduct	CMK正在被云产品使用。
FailedOperation.DecryptError	解密失败。
FailedOperation.EncryptionError	加密操作失败。
FailedOperation.TaggingError	标签服务错误。
InvalidParameter.DecryptMaterialError	解密EncryptedKeyMaterial失败。
InvalidParameter.InvalidPendingWindowInDays	计划删除时间参数非法。
InvalidParameterValue.AliasAlreadyExists	别名已经存在。
InvalidParameterValue.DuplicatedKeyId	KeyId重复。
InvalidParameterValue.InvalidAlias	别名格式错误。
InvalidParameterValue.InvalidCiphertext	密文格式错误。
InvalidParameterValue.InvalidHsmClusterId	无效的 HSM 集群 ID。
InvalidParameterValue.InvalidKeyId	KeyId不合法。
InvalidParameterValue.InvalidKeyUsage	KeyUsage参数错误。
InvalidParameterValue.InvalidPlaintext	Plaintext不合法。
InvalidParameterValue.InvalidType	Type参数错误。

错误码	说明
InvalidParameterValue.MaterialNotMatch	导入的密钥材料和历史导入不同。
InvalidParameterValue.TagKeysDuplicated	标签键重复。
InvalidParameterValue.TagsNotExisted	标签键或标签值不存在。
LimitExceeded.CmkLimitExceeded	CMK数量已达上限。
LimitExceeded.FingerprintsLimitExceeded	设备指纹个数超过限制。
LimitExceeded.KeyLimitExceeded	创建的密钥个数超过限制。
ResourceUnavailable.CloudResourceBindingNotFound	cmk和云资源的绑定关系不存在。
ResourceUnavailable.CmkArchived	CMK已存档。
ResourceUnavailable.CmkDisabled	CMK已被禁用。
ResourceUnavailable.CmkNotFound	CMK不存在。
ResourceUnavailable.CmkNotPendingDelete	CMK不是计划删除状态不能被执行取消计划删除。
ResourceUnavailable.CmkShouldBeDisabled	未被禁用的CMK不能被计划删除。
ResourceUnavailable.CmkStateNotSupport	CMK 状态不支持该操作。
ResourceUnavailable.KeyDisabled	密钥已被禁用。
ResourceUnavailable.KeyPendingDelete	不可用密钥：密钥待删除。
ResourceUnavailable.NotPurchased	白盒密钥服务尚未开通。
ResourceUnavailable.TokenExpired	Token已过期。
UnsupportedOperation.ExternalCmkCanNotRotate	用户导入类型的CMK禁止轮换。
UnsupportedOperation.NotExternalCmk	CMK类型错误，仅支持External CMK。
UnsupportedOperation.NotUserCreatedCmk	仅支持对用户自己创建的CMK做更新。
UnsupportedOperation.ServiceTemporaryUnavailable	服务暂时不可用。
UnsupportedOperation.UnsupportedKeyUsageInCurrentRegion	加密方式在当前region不支持。

密钥管理系统 API/SDK 2017 简介

最近更新时间：2023-08-09 09:29:23

❗ Note:

- 当前页面接口为旧版 API，未来可能停止维护。密钥管理系统 API 3.0 版本接口定义更加规范，访问时延下降显著，建议您使用 [密钥管理系统 API 3.0](#)。
- 如果您需要访问密钥管理系统旧版 API，可参见 [API 概览](#)。

欢迎使用密钥管理系统 KMS。

KMS 产品介绍请参考 [产品介绍](#)

支持的全部操作可参见 [API 概览页](#)。

⚠ Note:

协助者账号目前不能用于 kms 的操作。

密码学术语表

本文档涉及的一些常用术语如下：

术语	中文直译	含义
DataKey	数据密钥	实际用于加密业务数据的密钥
CMK	customer master key: 用户主密钥	用户创建的，用于加密小包数据（最多4KB）或者 DataKey 的密钥
KeyId	密钥 ID	CMK 返回给用户的全局唯一标识，注意这不是 CMK 本身，只是一个索引而已
CiphertextBlob	密文结构体	用户调加密或者 GenerateDataKey 接口返回的结构体，其中包含了明文的 KeyId 和密文的数据密钥或者加密后数据
根密钥	—	用于加密 CMKs 的密钥

KMS术语

术语	中文直译	含义
KMS	Key Management Service:密钥管理系统	PAAS 层密钥服务, 提供密钥安全管理和小包数据加解密服务
HSA	Harden Security Appliance:加固的安全模块	一个软件或硬件实现的安全模块，是 KMS 实现密钥管理，加密解密的核心部件
qps throttling	QPS Throttling	qps 限制

输入参数与返回参数释义

• limit 和 offset

用来控制分页的参数；当相应结果是列表形式时，如果数量超过了 limit 所限定的值，那么只返回 limit 个值。用户可以通过 limit 和 offset 两个参数来控制分页：limit 为单次返回的最多条目数量，offset 为偏移量。
举例来说，参数 offset=0&limit=20 返回第0到20项，offset=20&limit=20 返回第20到40项，offset=40&limit=20 返回第40到60项；以此类推。

• id.n

同时输入多个参数的格式。当遇到形如这样的格式时，那么该输入参数可以同时传多个。例如：

id.0= “10.12.243.21” &id.1= “10.12.243.21” &id.2= “10.12.243.21” &id.3= “10.12.243.21” ...

以此类推（以下标0开始）。

API快速入门

用户可以通过 [KMS SDK](#)（推荐，目前提供多个语言版本），或直接调用云API（较为麻烦，建议SDK语言之外的用户使用）使用KMS的服务：

选择地域和内外网。与其他云产品不同，KMS API 的请求域名随地域不同而变化，需选择地域对应的域名，请求域名的构成规则形如

kms-region.api.cloud.tencent.com/v2/index.php，其中 region 字段需用具体地域替换：gz（广州），sh（上海），bj（北京）。如果是用户使用的机器是腾讯云服务器，则应优先选择内网域名，否则选外网域名。

接口功能	接口名	功能描述
加密	Encrypt	用于加密最多为4KB任意数据，诸如 RSA 密钥，数据库密钥，或者其他敏感的客户信息。
解密	Decrypt	用于解密密文数据，得到明文数据。
创建主密钥	CreateKey	创建用户管理数据密钥的主密钥 CMK（Custom Master Key）。
获取主密钥列表	ListKey	用于获取用户所有的 keyId。
获取主密钥属性	GetKeyAttributes	用于获取指定 keyId 的属性信息。
禁用主密钥	DisableKey	用于禁用一个指定的 keyId。
启用主密钥	EnableKey	用于启用一个指定的 keyId。
生成数据密钥	GenerateDataKey	用于生成一个密钥，用户可以使用该密钥用于本地数据的加密。
修改主密钥属性	SetKeyAttributes	用于修改指定 keyId 的属性信息。

API 概览

最近更新时间：2019-11-06 16:13:50

密钥管理系统

加解密相关接口

接口功能	Action ID	功能描述
加密	Encrypt	用于加密最多为4KB任意数据，诸如 RSA 密钥，数据库密钥，或者其他敏感的客户信息。
解密	Decrypt	用于解密密文数据，得到明文数据。

密钥管理相关接口

接口功能	Action ID	功能描述
创建主密钥	CreateKey	创建用户管理数据密钥的主密钥 CMK（Custom Master Key）。
获取主密钥列表	ListKey	用于获取用户所有的 keyId。
获取主密钥属性	GetKeyAttributes	用于获取指定 keyId 的属性信息。
禁用主密钥	DisableKey	用于禁用一个指定的 keyId。
启用主密钥	EnableKey	用于启用一个指定的 keyId。
生成数据密钥	GenerateDataKey	用于生成一个密钥，用户可以使用该密钥用于本地数据的加密。
修改主密钥属性	SetKeyAttributes	用于修改指定 keyId 的属性信息。

更新历史

最近更新时间：2019-09-10 20:36:30

时间	更新内容
2017-2-24	创建密钥 SDK，API 文档

调用方式

请求结构

公共请求参数

最近更新时间：2024-06-04 11:03:51

公共请求参数是每个接口都需要使用到的请求参数，如非必要, 在各个接口单独的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。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：

名称	类型	描述	是否必选
Action	String	具体操作的指令接口名称，例如想要调用 查询实例列表 接口，则Action参数即为DescribeInstances。	是
Region	String	区域参数，用来标识希望操作哪个区域的实例。各区域的参数值为: 北京:bj, 广州:gz, 上海:sh, 中国香港:hk, 北美:ca。、注意：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。	否
Timestamp	UInt	当前UNIX时间戳，可记录发起API请求的时间。	是
Nonce	UInt	随机正整数，与 Timestamp 联合起来, 用于防止重放攻击。	是
SecretId	String	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 页面。	是
Signature	String	请求签名，用来验证此次请求的合法性，由系统根据输入参数自动生成。具体可参考 签名方法 页面。	是

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.api.qcloud.com/v2/index.php?
Action=DescribeInstances
&SecretId=xxxxxxx
&Region=gz
&Timestamp=1465055529
&Nonce=59485
&Signature=mysignature
&<接口请求参数>
```

一个完整的请求需要两类请求参数：公共请求参数和接口请求参数。

最终请求形式

最近更新时间：2019-09-10 23:25:23

最终的请求URL由以下几部分组成：

1. 请求域名： [查询实例列表](#)（DescribeInstances）的请求域名为： `cvm.api.qcloud.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。
2. 请求路径：云 API 的请求路径固定为 `/v2/index.php` 。
3. 最终请求参数串：包括公共请求参数和接口请求参数。

最终的请求 URL 的拼接规则为：

```
https:// + 请求域名 + 请求路径 + ? + 最终请求参数串
```

因此，我们得到最终的请求 URL 如下，其中前6个参数为公共请求参数，后6个参数为接口请求参数。

```
https://cvm.api.qcloud.com/v2/index.php?
Action=DescribeInstances
&SecretId=xxxxxxx
&Region=gz
&Timestamp=1465055529
&Nonce=59485
&Signature=mysignature
&instanceIds.0=ins-0hm4gvho
&instanceIds.1=ins-8oby8q00
&offset=0
&limit=20
&status=2
&zoneId=100003
```

请求结构简介

最近更新时间：2019-09-10 23:21:45

对腾讯云的 API 接口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请求参数来完成的。腾讯云 API 的请求结构由以下几部分组成：

1. 服务地址

腾讯云 API 的服务接入地址与具体模块相关，详见各接口描述。

2. 通信协议

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

KMS API 只支持 HTTPS 协议，目前支持 TLSv1 TLSv1.1 TLSv1.2

3. 请求方法

腾讯云API同时支持 POST 和 GET 请求。

⚠ Note:

1. 不能混合使用这两种请求方式，即如果使用 GET 方式，则参数均从 Querystring 取得；如果使用 POST 方式，则参数均从 Request Body 中取得，而 Querystring 中的参数将忽略。两种方式参数格式规则相同，一般情况下推荐使用 POST，因为 GET 可能会因为长度过长碰到不可预料的问题。
2. 无论用户的请求方法是 GET 或者 POST，都需要对参数编码。

4. 请求参数

腾讯云API的每个请求都需要指定两类参数：即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](#) 小节，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。

5. 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。

返回结果

错误返回结果

最近更新时间：2020-02-18 14:43:08

若API调用失败，则最终返回结果中的错误码code不为0，message字段会显示详细错误信息。您可以根据 code 和 message 在 [错误码](#) 文档查询具体的错误信息。
错误返回示例如下：

```
{
  "code": 5100,
  "message": "(100004)projectId不正确"
}
```

正确返回结果

最近更新时间：2019-09-10 23:17:15

若 API 调用成功，则最终返回结果中的错误码 code 为0，错误信息 message 为空，并且会显示返回的结果数据。

示例如下：

```
{  
  "code": 0,  
  "message": "  
<返回结果数据>  
"
```

错误码

最近更新时间：2023-08-09 09:30:21

message 字段表示与模块相关的错误。
其示例如下：

```
"message": "(100004)projectId不正确"
```

它由两部分组成，() 内的为模块错误代码，() 后的为具体的错误描述。
不同模块可能产生的错误情况是不一样的，用户可以根据具体的错误描述确定错误所在。

 **说明：**
下面的表都是公共错误码，如果您要查找的错误码不在其中，请到接口文档中具体接口说明处查找。

公共错误码

错误代码	模块错误代码	英文提示	描述
4000	10000	invalid request parameters	无效（非法）请求参数。通常是参数没有按照 KMS 云 API 的接口说明导致的错误。
4000	10010	lacked of required parameters	缺少必填参数。应参照 KMS 云 API 接口的说明看缺少哪些必填参数。
4100	10030	authentication failed	鉴权失败。可能的原因有： 生成签名算法的 secret key 错误。 签名算法有误（没用官方 sdk，自己实现签名算法）。
4300	10040	charge overdue	账户欠费。
6000	10050	server internal error	服务器内部错误。一般会有 requestId 返回给用户，用户可根据 requestId 提工单给技术支持人员定位问题。
4100	10080	secret id status error	secret ID 状态错误。secret ID 可能处于禁用状态。
4000	10110	request parameters error	请求参数错误。通常是参数没有按照 KMS 云 API 的接口说明导致的错误。
4420	10250	qps throttling	qps 达到最大值，因此被限制。为了保证后台的稳定性和调用时延，目前实际的限制是每秒的消息数最大5k。对于有特殊要求的客户，可以提工单申请。
4100	10270	secret id is not existed	secret ID 不存在。
4000	10280	action is not existed	参数 Action 不存在。
4000	10310	error: parameter key format error	参数格式错误。
4000	10320	no such parameter	不存在这样的参数。
4000	10330	parameter is NOT a repeatable parameter	参数不是一个可以重复的参数。
4000	10350	parameter value or length is out of range	参数取值或参数长度超过了范围限制。通常对于整型而言，是取值错误。对 String 类型参数而言，是长度错误。
4000	10360	parameter error type	参数类型错误。例如把整型填成了字符串类型。
4000	10370	batch size of parameter is too large	批量接口的批量值超过了最大值。

4000	10380	index of parameter is not consequent	可重复参数的下标不连续。
4000	10390	lacked of required parameter	缺少必填的参数。
4000	10400	cannot find parameter in uri	uri 中找不到参数。
4000	10410	unexpected http method, only GET or POST is supported	不支持的 HTTP 方法。目前只支持 GET 和 POST 方法。由于 GET 有长度的限制，建议使用 POST。
4000	10420	cannot parse	无法解析参数。
4000	10430	action name is not existed	Action所指定的接口名不存在。
4000	10440	account illegal, it may be an assistant account	账号不合法。请检查是不是协助者账号，目前 KMS 不支持协助者账号的操作。
4000	10450	secret id dosen't begin with AKID account	secret ID 不是以 AKID 开头。
4480	10460	exceed interface frequency limit, please slow down	为保护后端系统，控制类接口（例如 ListKey 等）的允许调用频率比 QPS 要低很多，所以看到该错误时，请减少控制类接口调用频率。
4530	10630	key is not existed	key 不存在，表明用户没有该 CMK，请检查该 key 是否存在。
4550	10650	key is disabled	该 key 被禁用了，请检查该 key 的状态是否为启用。

签名方法

最近更新时间：2024-06-05 09:17:23

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，否则就无法调用云 API 接口。

1. 申请安全凭证

在第一次使用云 API 之前，用户需要在腾讯云控制台上申请安全凭证。安全凭证包括 SecretId 和 SecretKey，其中 SecretId 是用于标识 API 调用者身份的，而 SecretKey 是用于加密签名字符串和服务器端验证签名字符串的密钥。用户应严格保管其 SecretKey，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. 单击云产品，选择[监控与管理](#)栏下的[云API密钥](#)，进入云 API 密钥管理页面。



3. 在 [云API访问密钥管理](#) 页面，单击新建即可以创建一对 SecretId/SecretKey，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。下面给出了一个生成签名串的详细过程。

假设用户的 SecretId 和 SecretKey 分别是

```
SecretId: AxxDz8xxxxJ5xxBZxxxx4WFkmLxxxxnPxxSA
SecretKey: Gu5xxxxARNpxxxxd98jxxxxN3xxxx1qA
```

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

以查看实例列表（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	AxxDz8xxxxJ5xxBZxxxx4WFkmLxxxxnPxxSA
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	gz
instanceIds.0	待查询的实例 ID	ins-09dx96dg
offset	偏移量	0
limit	最大允许输出	20

由上表可以看出，请求参数中的公共请求参数只有5个：Action、SecretId、Timestamp、Nonce 和 Region，而不是在“公共请求参数”中所述的6个，事实上，第6个参数 Signature（签名串）正是由其他参数（包括指令请求参数）共同生成的，具体步骤如下：

2.1. 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，例如 PHP 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
    'Action' : 'DescribeInstances',
    'Nonce' : 11886,
    'Region' : 'gz',
    'SecretId' : 'AxxDz8xxxxJ5xxBZxxxx4WFkmLxxxxnPxxSA',
    'Timestamp' : 1465185768,
    'instanceIds.0' : 'ins-09dx96dg',
    'limit' : 20,
    'offset' : 0,
}
```

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

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”，因此格式化后即为 Action=DescribeInstances。

注意：1、“参数值”为原始值而非 URL 编码后的值。2、若输入参数中包含下划线，则需要将其转换为“.”。

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&Nonce=11886&Region=gz&SecretId=AxxDz8xxxxJ5xxBZxxxx4WFkmLxxxxnPxxSA&Timestamp=1465185768&i
nstanceIds.0=ins-09dx96dg&limit=20&offset=0
```

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：

- 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：查看实例列表（DescribeInstances）的请求域名为：cvm.api.qcloud.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。
- 请求路径：云 API 的请求路径固定为 /v2/index.php。
- 请求字符串：即上一步生成的请求字符串。

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.api.qcloud.com/v2/index.php?
Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AxxDz8xxxxJ5xxBZxxxx4WFkmLxxxxnPxxSA&Timestamp=1465185768&i
nstanceIds.0=ins-09dx96dg&limit=20&offset=0
```

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，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 = 'Gu5xxxxARNpxxxxd98jxxxxN3xxxxlqA';
$srcStr = 'GETcvm.api.qcloud.com/v2/index.php?
Action=DescribeInstances&Nonce=11886&Region=gz&SecretId=AxxDz8xxxxJ5xxBZxxxx4WFkmLxxxxnPxxSA&Timestamp=1465185768&i
nstanceIds.0=ins-09dx96dg&limit=20&offset=0';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为：

```
NSI3UqqD99b/UJb4tbG/xZpRW64=
```

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, 则对所有请求参数值均需要做 URL 编码。

如上一步生成的签名串为NSI3UqqD99b/UJb4tbG/xZpRW64=, 则其编码后为NSI3UqqD99b%2FUJb4tbG%2FxZpRW64%3D。因此, 最终得到的签名串请求参数(Signature)为: NSI3UqqD99b%2FUJb4tbG%2FxZpRW64%3D, 它将用于生成最终的请求URL。

地域说明

最近更新时间：2019-09-10 20:32:31

外网接口请求域名：`kms-region.api.qcloud.com`

内网接口请求域名：`kms-region.api.tencentyun.com`

- 任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。
- 只支持 HTTPS 协议。

地域说明

接口请求域名中 region 需用具体地域替换，地域以域名的 region 为准，公共参数的 region 在这里会被忽略。目前 kms 上线 region 如下表所示：

区域	替换值	外网请求域名	内网请求域名
北京	bj	kms-bj.api.qcloud.com	kms-bj.api.tencentyun.com
上海	sh	kms-sh.api.qcloud.com	kms-sh.api.tencentyun.com
广州	gz	kms-gz.api.qcloud.com	kms-gz.api.tencentyun.com

加解密相关接口

加密

最近更新時間：2019-09-10 23:34:01

1. 接口描述

本接口（Encrypt）用于加密最多为4KB任意数据，诸如 RSA 密钥，数据库密钥，或者其他敏感的客户信息。

kms 请求域名接口和说明可参考 [地域和域名说明](#)。

说明：

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有code, message, requestId返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
keyId	是	String	调用 CreateKey 生成的 CMK 全局唯一标识符。
plaintext	是	String	被加密的明文数据，该字符串必须使用 base64 编码。KMS SDK 加密接口已经封装了 base64 编码操作。用户若使用 SDK 则不需要调用 base64 操作。
encryptionContext	否	json 格式字符串	key/value 对的 json 字符串，如果指定了该参数，则在调用 Decrypt API 时需要提供同样的参数。

3. 输出参数

参数名称	类型	描述
code	Int	0：表示成功，others：错误，详细错误见下表。
message	String	错误提示信息。
requestId	String	服务器生成的请求 Id，出现服务器内部错误时，用户可提交此 Id 给后台定位问题。
keyId	String	CMK 的全局唯一标识。
ciphertextBlob	String	加密后的密文。

错误代码	模块错误代码	英文提示	描述
4560	10660	encryption failed, please check input parameters	加密失败，请检查加密参数。
4540	10640	illegal encryptionContext	encryptionContext 非法，该字段必须为json格式字符串。
4590	10690	plaintext is not encoded with base64	plaintext 必须使用base64编码。

注意：

上表所列错误码是接口特有错误码，如果您要查找的错误码不在其中，可能在 [公共错误码](#) 中。

4. 示例

输入

```
https://kms-gz.api.tencentyun.com/v2/index.php?Action=Encrypt
&keyId=kms-30slkdf
&plaintext=databasepassword
```

&<公共请求参数>

输出

```
{
  "code" : 0,
  "message" : "",
  "requestId": "14534664555",
  "msgId": "123345346",
  "keyId": "kms-30slkdF",
  "ciphertextBlob": "string"
}
```

解密

最近更新时间：2019-12-06 10:23:45

1. 接口描述

本接口（Decrypt）用于解密密文，得到明文数据。此接口不需要 keyId 作为入参，因为加密接口得到的密文已经包含了 keyId。解密接口会获取密文中携带的 keyId，并验证权限。

说明：

- kms 请求域名接口和说明可参考 [地域和域名说明](#)。
- 任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有 code，message，requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
ciphertextBlob	是	String	被加密的密文数据。
encryptionContext	否	json 格式字符串	key/value 对的 json 字符串，应输入跟 Encrypt 接口相同的值。

3. 输出参数

参数名称	类型	描述
code	Int	0：表示成功，others：错误，详细错误见下表。
message	String	错误提示信息。
requestId	String	服务器生成的请求 ID，出现服务器内部错误时，用户可提交此 ID 给后台定位问题。
keyId	String	CMK 的全局唯一标识。
plaintext	String	解密后的明文。API 返回的 plaintext 是 base64 编码的，为了得到原始明文，应调用相应的 base64 解码。KMS SDK 解密接口已经封装了 base64 解密操作，若使用 SDK 不需要额外进行 base64 解密操作。

错误代码	模块错误代码	英文提示	描述
4560	10670	decryption failed, please check input parameters	解密失败，请检查输入参数。
4580	10680	invalid ciphertextBlob	ciphertextBlob 非法。

注意：

上表所列错误码是接口特有错误码，如果您要查找的错误码不在其中，可能在 [公共错误码](#) 中。

4. 示例

输入

```
https://kms-gz.api.tencentyun.com/v2/index.php?Action=Decrypt &ciphertextBlob=
<公共请求参数>
输出
```

输出

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

```
"requestId": "14534664555",  
"keyId": "kms-30slkdf",  
"plaintext": "string"  
}
```

密钥管理相关接口

修改主密钥属性

最近更新时间：2020-04-17 10:57:07

1. 接口描述

本接口 (SetKeyAttributes) 用于修改 KeyId 的属性信息。

kms 请求域名接口和说明可参考 [地域和域名说明](#)。

① 说明：

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有 code, message, requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
keyId	是	String	调用 CreateKey 生成的 CMK 的全局唯一标识符
alias	否	String	作为密钥更容易辨识，更容易被人看懂的别名，可为空，0 - 32个字符

3. 输出参数

参数名称	类型	描述
code	Int	0：表示成功，其他返回值的含义可以参考 错误码
message	String	错误提示信息
requestId	String	服务器生成的请求 ID，出现服务器内部错误时，用户可提交此 ID 给后台定位问题

4. 示例

输入

```
https://kms-sh.api.tencentyun.com/v2/index.php?Action=SetKeyAttributes
&keyId=kms-345sdfk
&alias=kms-test-1
&<公共请求参数>
```

输出

```
{
  "code" : 0,
  "message" : "",
  "requestId": "14534664555"
}
```

创建主密钥

最近更新时间：2019-12-06 10:30:04

1. 接口描述

本接口 (CreateKey) 用于创建用户管理数据密钥的主密钥CMK (Custom Master Key)，通过CMK后续可以调用其他接口诸如创建数据密钥、加解密等操作。

每个用户，在每个地区，最多可以创建1000个用户主密钥。

如果用户有提高数量限制的需求，可以通过工单系统向我们提交申请。

kms 请求域名接口和说明可参考 [地域和域名说明](#)。

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有 code，message，requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
description	否	String	key 的描述。
alias	是	String	作为密钥更容易辨识，更容易被人看懂的别名，不可为空，1 - 32个字符
keyUsage	否	String	指定 key 的用途。目前，默认为 'ENCRYPT/DECRYPT'，即 key 用于加密和解密
type	否	Int	指定 key 类型，1为普通类型，2为量子类型，默认为普通类型

3. 输出参数

参数名称	类型	描述
code	Int	0：表示成功，others：错误，详细错误见下表，
message	String	错误提示信息
requestId	String	服务器生成的请求 ID。出现服务器内部错误时，用户可提交此 ID 给后台定位问题
keyMetadata	Struct	密钥信息

keyMetadata 定义如下：

参数名称	类型	描述
keyId	String	key 的全局唯一标识符。
alias	String	作为密钥更容易辨识，更容易被人看懂的别名。
createTime	UInt32	密钥创建时间，UNIX 时间戳。
description	String	key 的描述。
keyState	String	key 的状态。
keyUsage	String	key 的用途。

错误码列表

错误代码	模块错误代码	英文提示	描述
4490	10470	string of KeyUsage must be ENCRYPT/DECRYPT	KeyUsage 参数错误，目前只能是 ENCRYPT 或者是 DECRYPT
4500	10480	alias is already existed in your account, please input another unique one	别名已经存在，请换一个别名
4510	10610	number of CMK in your account has reached the limit	CMK 数量达到最大值。一个账户的 CMK 数量是有限（200）的，达到最大值后，无法创建新的 CMK

4520	10620	keySpec value error	keySpec 错误
------	-------	---------------------	------------

 Note:

上表所列错误码是接口特有错误码，如果您要查找的错误码不在其中，请查看 [公共错误码](#) 文档。

4. 示例

输入：

```
https://kms-sh.api.tencentyun.com/v2/index.php?Action=CreateKey
&alias=kms-test-1
&description=this key is for kms test, and don't delete it
&keyUsage=ENCRYPT/DECRYPT
&type=1
&<公共请求参数>
```

输出：

```
{
  "code": 0,
  "message": "",
  "requestId": "14534664555",
  "keyMetadata": {
    "keyId": "kms-94kalsdf",
    "alias": "kms-test-1",
    "createTime": 1483945424,
    "description": "this key is for kms test, and don't delete it",
    "keyState": "Enabled",
    "keyUsage": "ENCRYPT/DECRYPT"
  }
}
```

启用主密钥

最近更新時間：2020-02-18 14:50:50

1. 接口描述

本接口 (EnableKey) 用于启用一个主密钥，允许该 Key 被使用。

kms请求域名接口和说明可参考 [地域和域名说明](#)。

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有 code, message, requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
keyId	是	String	调用 CreateKey 生成的 CMK 全局唯一标识符。

3. 输出参数

参数名称	类型	描述
code	Int	0：表示成功，其他返回值的含义可以参考 错误码 文档
message	String	错误提示信息
requestId	String	服务器生成的请求 ID。出现服务器内部错误时，用户可提交此 ID 给后台定位问题

4. 示例

输入

```
https://kms-gz.api.tencentyun.com/v2/index.php?Action=EnableKey
&keyId=kms-13f32sxxx
&<公共请求参数>
```

输出

```
{
  "code" : 0,
  "message" : "",
  "requestId": "14534664xxx"
}
```

生成数据密钥

最近更新时间：2020-03-25 16:58:59

1. 接口描述

本接口 (GenerateDataKey) 生成一个密钥，您可以用这个密钥进行本地数据的加密。

kms 请求域名接口和说明可参考 [地域和域名说明](#)。

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有 code, message, requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
keyId	是	String	调用 CreateKey 生成的 CMK 全局唯一标识符。
keySpec	否	String	指定生成 Datakey 的加密算法以及 Datakey 大小，AES_128 或者 AES_256。
numberOfBytes	否	Int	生成的 DataKey 的长度，同时指定 NumberOfBytes 和 KeySpec 时，以 NumberOfBytes 为准。
encryptionContext	否	json字符串	key/value 对的 json 字符串，如果指定了该参数，则在调用 Decrypt API 时需提供同样的参数。

3. 输出参数

参数名称	类型	描述
code	Int	0：表示成功，others：错误，详细错误见下表。
message	String	错误提示信息。
requestId	String	服务器生成的请求 ID。出现服务器内部错误时，用户可提交此 ID 给后台定位问题。
keyId	String	CMK 的全局唯一标识。
plaintext	String	生成的 DataKey 的明文，该明文使用 base64 编码，用户需要使用 base64 解码得到明文。KMS SDK 生成数据密钥接口已经封装了 base64 解码操作，用户若使用 SDK 则不需要调用 base64 解码操作。
ciphertextBlob	String	DataKey 加密后的密文。

4. 示例

输入

```
https://kms-bj.api.tencentyun.com/v2/index.php?Action=GenerateDataKey
&keyId=kms-345sdfk
&keySpec=AES_256
&<公共请求参数>
```

输出

```
{
  "code": 0,
  "message": "",
  "requestId": "14534664555",
  "keyId": "kms-345sdfk",
  "plaintext": "string",
  "ciphertextBlob": "string"
}
```

禁用主密钥

最近更新时间：2019-12-05 18:39:54

1. 接口描述

本接口（DisableKey）用于禁用一个主密钥,处于禁用状态的 Key 无法用于加密、解密操作。直到 Key 恢复到启用状态之前，原来使用该 Key 加密的密文也无法解密。
kms请求域名接口和说明可参考 [地域和域名说明](#)。

说明:

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有 code, message, requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
keyId	是	String	调用CreateKey生成的CMK全局唯一标识符。

3. 输出参数

参数名称	类型	描述
code	Int	—
message	String	错误提示信息。
requestId	String	服务器生成的请求 ID。出现服务器内部错误时，用户可提交此 ID 给后台定位问题。

4. 示例

输入

```
https://kms-gz.api.tencentyun.com/v2/index.php?Action=DisableKey
&keyId=kms-345sdfk
&<公共请求参数>
```

输出

```
{
  "code" : 0,
  "message" : "",
  "requestId": "14534664555"
}
```

获取主密钥列表

最近更新时间：2020-02-18 14:48:04

1. 接口描述

本接口 (ListKey)，用于列出账号下面的密钥列表。

kms请求域名接口和说明可参考 [地域和域名说明](#)。

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户；失败情况下，至少会有 code，message，requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 文档。

参数名称	是否必选	类型	描述
offset	否	Int	含义跟 SQL 查询的 Offset 一致，表示本次获取从按一定顺序排列数组的第 Offset 个元素开始。缺省为0
limit	否	Int	含义跟 SQL 查询的 Limit 一致，表示本次获最多获取 Limit 个元素。缺省值为10

3. 输出参数

参数名称	类型	描述
code	Int	0：表示成功，其他返回值的含义可以参考 错误码
message	String	错误提示信息
requestId	String	服务器生成的请求 ID。出现服务器内部错误时，用户可提交此 ID 给后台定位问题
keys	Array	key 数组
totalCount	Int	key 的总数

Keys 的每个元素定义如下：

参数名称	类型	描述
keyId	String	CMK 的全局唯一标识

4. 示例

输入

```
https://kms-bj.api.tencentyun.com/v2/index.php?Action=ListKey
&offset=20
&limit=10
&<公共请求参数>
```

输出

```
{
  "code": 0,
  "message": "",
  "requestId": "14534664555",
  "keys": [{
    "keyId": "kms-984ksdl"
  },
  {
    "keyId": "kms-84sjkla"
  }
],
  "totalCount": 100
}
```

```
}
```

获取主密钥属性

最近更新时间：2019-12-05 18:46:29

1. 接口描述

本接口（GetKeyAttributes）用于返回指定 KeyId 的属性信息。

kms请求域名接口和说明可参考 [地域和域名说明](#)。

说明:

任何时候，包括内测期间，如果使用外网域名产生公网下行流量，都会收取流量费用。所以强烈建议服务在腾讯云上的用户使用内网域名，内网不会产生流量费用。

- 输入参数有些是可选的，不填取默认值。
- 输出参数在成功情况下所有出参都会返回给用户。失败情况下，至少会有 code, message, requestId 返回。

2. 输入参数

以下请求参数列表仅列出了接口请求参数，其它参数见 [公共请求参数](#) 页面。

参数名称	是否必选	类型	描述
keyId	是	String	调用 CreateKey 生成的 CMK 全局唯一标识符。

3. 输出参数

参数名称	类型	描述
code	Int	请参考 错误码
message	String	错误提示信息。
requestId	String	服务器生成的请求 ID。出现服务器内部错误时，用户可提交此 ID 给后台定位问题。
keyMetadata	Struct	密钥信息。

keyMetadata定义如下：

参数名称	类型	描述	取值	备注
keyId	String	key 的全局唯一标识符	—	—
alias	String	作为密钥更容易辨识，更容易被人看懂的别名。	可为空。0-32个字符。	—
createTime	UInt32	密钥创建时间	时间戳	—
description	String	key 的描述	key 的描述	—
keyState	String	key 的状态	Enabled 或者 Disabled	—
keyUsage	String	key 的用途	ENCRYPT/DECRYPT	—
type	Int	指定key类型，1为普通类型，2为量子类型。	—	—

4. 示例

输入

```
https://kms-gz.api.tencentyun.com/v2/index.php?Action=GetKeyAttributes
&keyId=kms-345sdfk
&<公共请求参数>
```

输出

```
{
  "code" : 0,
  "message" : "",
  "requestId": "14534664555",
  "keyMetadata": {
    "keyId": "kms-345sdfk",
    "createTime": 1483945424,
    "description": "test",
    "keyState": "Enabled",
    "keyUsage": "ENCRYPT/DECRYPT",
    "type": 1
  }
}
```

```
}  
}
```