

云函数 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

触发器相关接口

设置函数触发方式

删除触发器

更新触发器状态

获取函数触发器列表

更新云函数自定义域名

获取云函数自定义域名列表

查看云函数自定义域名详情

删除云函数自定义域名

创建云函数自定义域名

更新触发器

函数相关接口

创建函数版本别名

删除别名

删除函数指定版本

删除函数

删除预置并发

删除函数最大独占配额

获取别名详细信息

获取函数代码下载地址

获取函数异步重试配置

获取函数预置并发详情

获取函数单个请求运行状态

获取函数最大独占配额详情

同步Invoke调用接口

获取别名列表

获取函数列表

查询函数版本

发布新版本

设置预置并发

设置最大独占配额

修改账号并发限制配额

更新别名的配置

更新函数代码

更新函数异步重试配置

创建函数

获取函数详细信息

获取函数运行日志

运行函数

更新函数配置

复制函数

命名空间相关接口

删除命名空间

更新命名空间

创建命名空间
列出命名空间列表
层管理相关接口
删除层版本
获取层版本详细信息
获取层版本列表
返回层列表
发布层版本
异步事件管理相关接口
拉取函数异步事件列表
终止函数异步事件
获取函数异步事件状态
其他接口
获取账户信息
数据结构
错误码
附录
函数和层的状态说明

API 文档

更新历史

最近更新时间：2025-03-13 01:54:55

第 70 次发布

发布时间：2025-03-13 01:54:42

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateCustomDomain](#)
 - 新增入参：Tags
- [GetCustomDomain](#)
 - 新增出参：Tags

修改数据结构：

- [DomainInfo](#)
 - 新增成员：Tags
- [TriggerInfo](#)
 - 新增成员：BoundResources

第 69 次发布

发布时间：2025-01-03 01:26:06

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteFunctionVersion](#)

修改数据结构：

- [FunctionLog](#)
 - 废弃成员：Level, Source
- [IntranetConfigOut](#)
 - 修改成员：IpFixed, IpAddress
- [Trigger](#)
 - 修改成员：ResourceId, BindStatus, TriggerAttribute, Description
- [TriggerInfo](#)
 - 修改成员：ResourceId, BindStatus, TriggerAttribute, Description

第 68 次发布

发布时间：2024-12-06 01:27:18

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [NamespaceResourceEnv](#)
 - 新增成员：OFFLINE

第 67 次发布

发布时间：2024-10-25 17:22:38

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateCustomDomain](#)
- [DeleteCustomDomain](#)
- [GetCustomDomain](#)
- [ListCustomDomains](#)
- [UpdateCustomDomain](#)

新增数据结构：

- [CertConf](#)
- [DomainInfo](#)
- [EndpointsConf](#)
- [PathRewriteRule](#)
- [WafConf](#)

第 66 次发布

发布时间：2024-08-28 02:17:22

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [LayerVersionInfo](#)
 - 新增成员：Tags
 - 修改成员：CompatibleRuntimes, AddTime, Description, LicenseInfo, Status, Stamp

第 65 次发布

发布时间：2024-08-27 02:17:25

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [PublishLayerVersion](#)
 - 新增入参：Tags

第 64 次发布

发布时间：2023-12-29 01:22:39

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetFunction](#)
 - 修改入参：FunctionName
- [UpdateFunctionConfiguration](#)
 - 新增入参：IgnoreSysLog

第 63 次发布

发布时间：2023-11-30 05:41:26

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateFunctionConfiguration](#)
 - 新增入参：DnsCache, IntranetConfig

第 62 次发布

发布时间：2023-11-29 02:01:37

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：DnsCache, IntranetConfig

新增数据结构：

- [IntranetConfigIn](#)

第 61 次发布

发布时间：2023-11-24 01:21:32

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetFunction](#)
 - 新增出参：ImageConfig, DnsCache, IntranetConfig

新增数据结构：

- [IntranetConfigOut](#)

第 60 次发布

发布时间：2023-08-22 01:18:21

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [UpdateTrigger](#)

第 59 次发布

发布时间：2023-08-04 01:16:09

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [NamespaceResourceEnvTKE](#)
 - 新增成员：PodTemplatePatch

第 58 次发布

发布时间：2023-08-02 01:20:41

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [K8SToleration](#)
 - 修改成员：Key

第 57 次发布

发布时间：2023-07-20 01:39:23

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateNamespace](#)
 - 新增入参：ResourceEnv

新增数据结构：

- [K8SLabel](#)
- [K8SToleration](#)
- [NamespaceResourceEnv](#)
- [NamespaceResourceEnvTKE](#)

第 56 次发布

发布时间：2023-06-30 01:40:45

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：AutoDeployClsTopicIndex, AutoCreateClsTopic

第 55 次发布

发布时间：2023-06-08 01:51:22

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [FunctionLog](#)
 - 修改成员：StartTime

第 54 次发布

发布时间：2023-05-31 01:38:53

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Trigger](#)
 - 修改成员：ResourceId, BindStatus, TriggerAttribute
 - 废弃成员：ResourceId, BindStatus, TriggerAttribute

- [TriggerInfo](#)
 - 修改成员: ResourceId, BindStatus, TriggerAttribute
 - 废弃成员: ResourceId, BindStatus, TriggerAttribute

第 53 次发布

发布时间: 2023-05-15 01:55:08

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateTrigger](#)
 - 新增入参: Description

修改数据结构:

- [Trigger](#)
 - 新增成员: Description

第 52 次发布

发布时间: 2023-04-24 01:35:50

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [TriggerInfo](#)
 - 新增成员: Description

第 51 次发布

发布时间: 2023-04-20 01:34:05

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [LayerVersionInfo](#)
 - 新增成员: Stamp

第 50 次发布

发布时间: 2023-04-04 01:39:21

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [UpdateTriggerStatus](#)

第 49 次发布

发布时间: 2023-03-24 02:04:32

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [ImageConfig](#)
 - 新增成员: ImagePort

第 48 次发布

发布时间: 2023-02-21 01:58:57

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [TriggerCount](#)
 - 新增成员: Eb

第 47 次发布

发布时间: 2022-11-17 06:40:32

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateFunction](#)
 - 新增入参: InstanceConcurrencyConfig
- [UpdateFunctionConfiguration](#)
 - 新增入参: InstanceConcurrencyConfig

新增数据结构:

- [InstanceConcurrencyConfig](#)

修改数据结构:

- [ImageConfig](#)
 - 新增成员: ContainerImageAccelerate

第 46 次发布

发布时间: 2022-05-09 06:12:32

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [NamespaceUsage](#)
 - 新增成员: TotalAllocatedProvisionedMem

第 45 次发布

发布时间: 2022-03-21 08:15:19

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [PutProvisionedConcurrencyConfig](#)
 - 新增入参: ProvisionedType, TrackingTarget, MinCapacity, MaxCapacity

第 44 次发布

发布时间: 2022-03-16 08:13:22

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [NamespaceUsage](#)
 - 新增成员：TotalConcurrencyMem, TotalAllocatedConcurrencyMem

第 43 次发布

发布时间：2022-01-12 08:13:52

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [TriggerAction](#)
 - 新增成员：ProvisionedType

第 42 次发布

发布时间：2021-12-09 08:09:39

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [FunctionVersion](#)
 - 新增成员：Status

第 41 次发布

发布时间：2021-12-08 08:14:03

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetAsyncEventStatus](#)

新增数据结构：

- [AsyncEventStatus](#)

第 40 次发布

发布时间：2021-12-03 08:10:14

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateFunctionCode](#)
 - 新增入参：InstallDependency

第 39 次发布

发布时间：2021-12-02 08:10:45

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetRequestStatus](#)

新增数据结构：

- [RequestStatus](#)

第 38 次发布

发布时间：2021-11-17 08:14:54

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：InstallDependency
- [UpdateFunctionConfiguration](#)
 - 新增入参：InstallDependency

第 37 次发布

发布时间：2021-11-08 08:09:22

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：ProtocolType, ProtocolParams
- [GetFunction](#)
 - 新增出参：ProtocolType, ProtocolParams
- [UpdateFunctionConfiguration](#)
 - 新增入参：ProtocolParams

新增数据结构：

- [ProtocolParams](#)
- [WSPParams](#)

第 36 次发布

发布时间：2021-11-02 08:10:12

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [TerminateAsyncEvent](#)
 - 新增入参：GraceShutdown

第 35 次发布

发布时间：2021-10-25 08:09:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [PutProvisionedConcurrencyConfig](#)

- 新增入参: TriggerActions

新增数据结构:

- [TriggerAction](#)

修改数据结构:

- [VersionProvisionedConcurrencyInfo](#)
 - 新增成员: TriggerActions

第 34 次发布

发布时间: 2021-09-17 08:11:57

本次发布包含了以下内容:

改善已有的文档。

新增数据结构:

- [ImageConfig](#)

修改数据结构:

- [Code](#)
 - 新增成员: ImageConfig

第 33 次发布

发布时间: 2021-09-07 08:07:50

本次发布包含了以下内容:

改善已有的文档。

修改数据结构:

- [Function](#)
 - 新增成员: AsyncRunEnable, TraceEnable

第 32 次发布

发布时间: 2021-08-20 08:07:56

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [ListNamespaces](#)
 - 新增入参: SearchKey

新增数据结构:

- [SearchKey](#)

第 31 次发布

发布时间: 2021-08-03 08:08:15

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [GetFunction](#)
 - 新增出参: AsyncRunEnable, TraceEnable

第 30 次发布

发布时间：2021-07-09 08:07:55

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [InvokeFunction](#)

第 29 次发布

发布时间：2021-05-31 08:07:35

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateFunctionCode](#)
 - 修改入参：Handler

第 28 次发布

发布时间：2021-04-26 08:07:03

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：AsyncRunEnable, TraceEnable

第 27 次发布

发布时间：2021-03-16 08:07:54

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetFunctionEventInvokeConfig](#)
- [UpdateFunctionEventInvokeConfig](#)

新增数据结构：

- [AsyncTriggerConfig](#)
- [RetryConfig](#)

第 26 次发布

发布时间：2021-03-09 08:07:21

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [GetAccount](#)

新增数据结构：

- [LimitsInfo](#)

- [NamespaceLimit](#)
- [NamespaceUsage](#)
- [TriggerCount](#)
- [UsageInfo](#)

第 25 次发布

发布时间：2021-01-26 08:07:28

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ListAsyncEvents](#)
- [TerminateAsyncEvent](#)

新增数据结构：

- [AsyncEvent](#)
- [TimeInterval](#)

第 24 次发布

发布时间：2020-12-31 08:06:16

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Trigger](#)
 - 新增成员：Qualifier

第 23 次发布

发布时间：2020-12-21 08:06:44

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteProvisionedConcurrencyConfig](#)
- [DeleteReservedConcurrencyConfig](#)
- [GetProvisionedConcurrencyConfig](#)
- [GetReservedConcurrencyConfig](#)
- [PutProvisionedConcurrencyConfig](#)
- [PutReservedConcurrencyConfig](#)
- [PutTotalConcurrencyConfig](#)

新增数据结构：

- [VersionProvisionedConcurrencyInfo](#)

第 22 次发布

发布时间：2020-11-20 08:05:55

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Function](#)
 - 新增成员: StatusReasons, TotalProvisionedConcurrencyMem, ReservedConcurrencyMem
- [Trigger](#)
 - 新增成员: ResourceId, BindStatus, TriggerAttribute
- [TriggerInfo](#)
 - 新增成员: ResourceId, BindStatus, TriggerAttribute

第 21 次发布

发布时间: 2020-10-13 08:05:52

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateTrigger](#)
 - 新增入参: CustomArgument

第 20 次发布

发布时间: 2020-09-24 08:05:31

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [CreateFunction](#)
 - 新增入参: Tags

第 19 次发布

发布时间: 2020-09-16 08:05:21

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [GetFunction](#)
 - 新增出参: StatusReasons

新增数据结构:

- [StatusReason](#)

第 18 次发布

发布时间: 2020-08-20 08:18:57

本次发布包含了以下内容:

改善已有的文档。

修改接口:

- [DeleteFunction](#)
 - 新增入参: Qualifier

第 17 次发布

发布时间: 2020-08-10 08:17:16

本次发布包含了以下内容:

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：InitTimeout
- [GetFunction](#)
 - 新增出参：InitTimeout
- [UpdateFunctionConfiguration](#)
 - 新增入参：InitTimeout

第 16 次发布

发布时间：2020-07-16 08:16:53

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：CfsConfig
- [GetFunction](#)
 - 新增出参：CfsConfig, AvailableStatus, Qualifier
- [UpdateFunctionConfiguration](#)
 - 新增入参：CfsConfig

新增数据结构：

- [CfsConfig](#)
- [CfsInsInfo](#)

第 15 次发布

发布时间：2020-06-10 08:15:37

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [FunctionLog](#)
 - 新增成员：RetryNum

第 14 次发布

发布时间：2020-05-11 10:13:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateFunction](#)
 - 新增入参：PublicNetConfig
- [UpdateFunctionConfiguration](#)
 - 新增入参：PublicNetConfig

新增数据结构：

- [EipConfigIn](#)
- [PublicNetConfigIn](#)

第 13 次发布

发布时间：2020-05-08 08:15:22

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ListTriggers](#)

新增数据结构：

- [TriggerInfo](#)

第 12 次发布

发布时间：2020-04-30 08:14:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAlias](#)
- [GetAlias](#)
- [ListAliases](#)
- [UpdateAlias](#)

修改接口：

- [UpdateFunctionConfiguration](#)
 - 新增入参：L5Enable

新增数据结构：

- [Alias](#)
- [RoutingConfig](#)
- [VersionMatch](#)
- [VersionWeight](#)

第 11 次发布

发布时间：2020-04-29 08:15:20

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteAlias](#)

修改接口：

- [UpdateFunctionConfiguration](#)
 - 删除入参：L5Enable, OnsEnable

第 10 次发布

发布时间：2020-04-24 08:14:03

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Trigger](#)
 - 新增成员：AvailableStatus

第 9 次发布

发布时间：2020-04-16 08:14:01

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [Invoke](#)
 - 新增入参：RoutingKey

第 8 次发布

发布时间：2020-04-14 08:13:51

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ListVersionByFunction](#)
 - 新增入参：Offset, Limit, Order, OrderBy
 - 新增出参：TotalCount

第 7 次发布

发布时间：2020-04-02 08:14:22

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [FunctionVersion](#)
 - 新增成员：AddTime, ModTime

第 6 次发布

发布时间：2020-02-19 16:00:09

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteLayerVersion](#)
- [GetLayerVersion](#)
- [ListLayerVersions](#)
- [ListLayers](#)
- [PublishLayerVersion](#)

修改接口：

- [CreateFunction](#)
 - 新增入参：Layers, DeadLetterConfig
- [GetFunction](#)
 - 新增出参：Layers, DeadLetterConfig, AddTime, PublicNetConfig, OnsEnable
- [GetFunctionLogs](#)
 - 修改入参：FunctionName
- [UpdateFunctionConfiguration](#)
 - 新增入参：Layers, DeadLetterConfig, OnsEnable

新增数据结构：

- [DeadLetterConfig](#)
- [EipConfigOut](#)
- [LayerVersionInfo](#)
- [LayerVersionSimple](#)
- [PublicNetConfigOut](#)

第 5 次发布

发布时间：2019-09-26 20:40:24

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [GetFunction](#)
 - 新增出参：L5Enable
- [GetFunctionLogs](#)
 - 修改入参：FunctionName
- [UpdateFunctionCode](#)
 - 新增入参：Code, CodeSource
- [UpdateFunctionConfiguration](#)
 - 新增入参：L5Enable

第 4 次发布

发布时间：2019-08-15 20:51:37

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateNamespace](#)
- [DeleteNamespace](#)
- [GetFunctionAddress](#)
- [ListNamespaces](#)
- [ListVersionByFunction](#)
- [PublishVersion](#)
- [UpdateNamespace](#)

修改接口：

- [CopyFunction](#)
 - 新增入参：Override, CopyConfiguration
- [CreateFunction](#)
 - 新增入参：Namespace, Role, Type, CodeSource
- [CreateTrigger](#)
 - 新增入参：Namespace
- [DeleteFunction](#)
 - 新增入参：Namespace
- [DeleteTrigger](#)
 - 新增入参：Namespace
- [GetFunction](#)
 - 新增入参：Namespace
 - 新增出参：EipConfig, AccessInfo, Type
- [GetFunctionLogs](#)
 - 新增入参：Namespace, SearchContext
 - 新增出参：SearchContext
- [ListFunctions](#)
 - 新增入参：Namespace

- [UpdateFunctionCode](#)
 - 新增入参：Namespace, EnvId, Publish
- [UpdateFunctionConfiguration](#)
 - 新增入参：Namespace, Role, ClsLogsetId, ClsTopicId, Publish

新增数据结构：

- [AccessInfo](#)
- [EipOutConfig](#)
- [FunctionVersion](#)
- [LogSearchContext](#)
- [Namespace](#)

修改数据结构：

- [Code](#)
 - 新增成员：GitUrl, GitUserName, GitPassword, GitPasswordSecret, GitBranch, GitDirectory, GitCommitId, GitUserNameSecret
- [Function](#)
 - 新增成员：Type
- [FunctionLog](#)
 - 新增成员：Level, Source

第 3 次发布

发布时间：2019-03-21 19:50:07

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CopyFunction](#)

修改接口：

- [CreateFunction](#)
 - 新增入参：ClsLogsetId, ClsTopicId
- [CreateTrigger](#)
 - 新增入参：Enable
 - 新增出参：TriggerInfo
- [GetFunction](#)
 - 新增出参：InstallDependency, Status, StatusDesc, ClsLogsetId, ClsTopicId, FunctionId, Tags
- [GetFunctionLogs](#)
 - 修改入参：Filter
- [ListFunctions](#)
 - 新增入参：Description, Filters

新增数据结构：

- [LogFilter](#)
- [Tag](#)

修改数据结构：

- [Code](#)
 - 新增成员：Demold, TempCosObjectName
- [Filter](#)
 - 新增成员：Name, Values
 - 删除成员：RetCode
- [Function](#)
 - 新增成员：Status, StatusDesc, Description, Tags
- [Trigger](#)

- 新增成员：Enable, CustomArgument

第 2 次发布

发布时间：2018-07-26 19:15:23

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateFunction](#)
- [CreateTrigger](#)
- [DeleteFunction](#)
- [DeleteTrigger](#)
- [GetFunction](#)
- [GetFunctionLogs](#)
- [ListFunctions](#)
- [UpdateFunctionCode](#)
- [UpdateFunctionConfiguration](#)

修改接口：

- [Invoke](#)
 - 新增入参：Namespace

新增数据结构：

- [Code](#)
- [Environment](#)
- [Filter](#)
- [Function](#)
- [FunctionLog](#)
- [Trigger](#)
- [Variable](#)
- [VpcConfig](#)

第 1 次发布

发布时间：2018-05-17 15:58:40

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [Invoke](#)

新增数据结构：

- [Result](#)

简介

最近更新时间：2025-04-25 01:51:59

欢迎使用腾讯云云函数（Serverless Cloud Function）。云函数通过托管用户代码使用用户配置的触发器，根据触发器的触发事件自动执行用户应用，为用户提供 FaaS（Function as a Service）类型服务。

云函数（Serverless Cloud Function）是腾讯云提供的无服务器（Serverless）执行环境，您无需购买和管理服务器，而只需使用平台支持的语言编写核心代码并设置代码运行的条件，代码即可在腾讯云基础设施上弹性、安全地运行。腾讯云完全管理底层计算资源，包括服务器 CPU、内存、网络和其他配置/资源维护、代码部署、弹性伸缩、负载均衡等。代码按需运行，空闲时不收费。

注意：

- 本章节云函数 API 接口均为最新 API 3.0 接口，后续 SCF 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 API 3.0 接口。
- 现有旧版 API 接口功能依然保持，未来可能停止维护，如您仍需使用旧版接口可参考：[云函数 API（旧版）简介](#)。

术语表

术语	全称	中文名称	说明
SCF	Serverless Cloud Function	云函数	云函数通过托管用户代码，使用用户配置的触发器，根据触发器的触发事件，自动执行用户应用，为用户提供 FaaS（Function as a Service）类型服务。
Trigger	SCF Trigger	函数触发器	SCF 云函数由事件触发执行，事件来源即触发器，通常是由其他产品资源对象作为触发器，例如 COS Bucket，CMQ Topic队列，定时器等。

API 概览

最近更新時間：2025-01-03 01:26:28

其他接口

接口名称	接口功能	频率限制（次/秒）
GetAccount	获取账户信息	20

触发器相关接口

接口名称	接口功能	频率限制（次/秒）
CreateTrigger	设置函数触发方式	100
DeleteTrigger	删除触发器	100
UpdateTriggerStatus	更新触发器状态	20
ListTriggers	获取函数触发器列表	20
CreateCustomDomain	创建云函数自定义域名	20
DeleteCustomDomain	删除云函数自定义域名	20
GetCustomDomain	查看云函数自定义域名详情	20
ListCustomDomains	获取云函数自定义域名列表	20
UpdateCustomDomain	更新云函数自定义域名	20
UpdateTrigger	更新触发器	20

函数相关接口

接口名称	接口功能	频率限制（次/秒）
CreateAlias	创建函数版本别名	20
DeleteAlias	删除别名	20
DeleteFunction	删除函数	100
DeleteFunctionVersion	删除函数指定版本	20
DeleteProvisionedConcurrencyConfig	删除预置并发	20
DeleteReservedConcurrencyConfig	删除函数最大独占配额	20
GetAlias	获取别名详细信息	20
GetFunctionAddress	获取函数代码下载地址	20
GetFunctionEventInvokeConfig	获取函数异步重试配置	20
GetProvisionedConcurrencyConfig	获取函数预置并发详情	20
GetRequestStatus	获取函数单个请求运行状态	20
GetReservedConcurrencyConfig	获取函数最大独占配额详情	20
InvokeFunction	同步Invoke调用接口	10000
ListAliases	获取别名列表	20
ListFunctions	获取函数列表	100
ListVersionByFunction	查询函数版本	20

接口名称	接口功能	频率限制（次/秒）
PublishVersion	发布新版本	20
PutProvisionedConcurrencyConfig	设置预置并发	20
PutReservedConcurrencyConfig	设置最大独占配额	20
PutTotalConcurrencyConfig	修改账号并发限制配额	20
UpdateAlias	更新别名的配置	20
UpdateFunctionCode	更新函数代码	100
UpdateFunctionEventInvokeConfig	更新函数异步重试配置	20
CreateFunction	创建函数	100
GetFunction	获取函数详细信息	100
GetFunctionLogs	获取函数运行日志	5
Invoke	运行函数	100
UpdateFunctionConfiguration	更新函数配置	100
CopyFunction	复制函数	20

命名空间相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteNamespace	删除命名空间	20
UpdateNamespace	更新命名空间	20
CreateNamespace	创建命名空间	20
ListNamespaces	列出命名空间列表	20

层管理相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteLayerVersion	删除层版本	20
GetLayerVersion	获取层版本详细信息	20
ListLayerVersions	获取层版本列表	20
ListLayers	返回层列表	20
PublishLayerVersion	发布层版本	20

异步事件管理相关接口

接口名称	接口功能	频率限制（次/秒）
ListAsyncEvents	拉取函数异步事件列表	20
TerminateAsyncEvent	终止函数异步事件	20
GetAsyncEventStatus	获取函数异步事件状态	20

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 01:28:35

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 `scf.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `scf.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 `scf.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）	scf.tencentcloudapi.com
华南地区（广州）	scf.ap-guangzhou.tencentcloudapi.com
华东地区（上海）	scf.ap-shanghai.tencentcloudapi.com
华东地区（南京）	scf.ap-nanjing.tencentcloudapi.com
华北地区（北京）	scf.ap-beijing.tencentcloudapi.com
西南地区（成都）	scf.ap-chengdu.tencentcloudapi.com
西南地区（重庆）	scf.ap-chongqing.tencentcloudapi.com
港澳台地区（中国香港）	scf.ap-hongkong.tencentcloudapi.com
亚太东南（新加坡）	scf.ap-singapore.tencentcloudapi.com
亚太东南（雅加达）	scf.ap-jakarta.tencentcloudapi.com
亚太东南（曼谷）	scf.ap-bangkok.tencentcloudapi.com
亚太东北（首尔）	scf.ap-seoul.tencentcloudapi.com
亚太东北（东京）	scf.ap-tokyo.tencentcloudapi.com
美国东部（弗吉尼亚）	scf.na-ashburn.tencentcloudapi.com
美国西部（硅谷）	scf.na-siliconvalley.tencentcloudapi.com
南美地区（圣保罗）	scf.sa-saopaulo.tencentcloudapi.com
欧洲地区（法兰克福）	scf.eu-frankfurt.tencentcloudapi.com

注意：由于金融区和非金融区是隔离不通的，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）	scf.ap-shanghai-fsi.tencentcloudapi.com
华南地区（深圳金融）	scf.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：2025-03-28 01:56:27

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入参公共参数 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 scf；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
西南地区（成都）	ap-chengdu
西南地区（重庆）	ap-chongqing
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

地域	取值
亚太东北（东京）	ap-tokyo
欧洲地区（法兰克福）	eu-frankfurt
美国东部（弗吉尼亚）	na-ashburn
美国西部（硅谷）	na-siliconvalley
南美地区（圣保罗）	sa-saopaulo

签名方法 v3

最近更新時間：2024-12-25 01:56:19

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++)
    {
        sprintf(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* const 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

最近更新时间：2024-12-25 01:56:19

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：

- 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】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
InstanceId.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("cvm.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 01:46:14

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 06:45:10

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。

触发器相关接口

设置函数触发方式

最近更新时间：2025-05-23 01:27:22

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据参数输入设置新的触发方式。

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：CreateTrigger。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	新建触发器绑定的函数名称 示例值：FunctionName
TriggerName	是	String	新建触发器名称。如果是定时触发器，名称支持英文字母、数字、连接符和下划线，最长100个字符；如果是cos触发器，需要是对应cos存储桶适用于XML API的访问域名(例如:5401-5ff414-12345.cos.ap-shanghai.myqcloud.com);如果是其他触发器，见具体触发器绑定参数的说明 示例值：TriggerName
Type	是	String	触发器类型，目前支持 cos 、cls 、timer、ckafka、http类型。创建函数 URL 请使用 http 类型，参考 创建函数 URL 。创建 cls 触发器请参考 CLS 创建投递 SCF 任务 。 示例值：timer
TriggerDesc	否	String	触发器对应的参数，可见具体 触发器描述说明 示例值：/2***
Namespace	否	String	函数的命名空间 示例值：default
Qualifier	否	String	触发器所生效的版本或别名，建议填写 \$DEFAULT 方便后续进行版本的灰度发布，默认为 \$LATEST 。 示例值：\$DEFAULT
Enable	否	String	触发器的初始是能状态 OPEN表示开启 CLOSE表示关闭 示例值：1
CustomArgument	否	String	用户自定义参数，仅支持timer触发器 示例值：CustomArgument
Description	否	String	触发器描述 示例值：A trigger

3. 输出参数

参数名称	类型	描述
TriggerInfo	Trigger	触发器信息 示例值：{ "AddTime": "2021-06-09 16:07:18", "AvailableStatus": "Availa

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 创建新的触发器

用户使用该函数创建新的触发器

输入示例

```
https://scf.tencentcloudapi.com/?Action=CreateTrigger
&FunctionName=<FunctionName>
&TriggerName=<TriggerName>
&Type=timer
&TriggerDesc=*/2****
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TriggerInfo": {
      "AddTime": "2021-06-09 16:07:18",
      "AvailableStatus": "Available",
      "BindStatus": "on",
      "CustomArgument": "",
      "Description": "",
      "Enable": 0,
      "ModTime": "2021-08-31 21:45:32",
      "Qualifier": "$DEFAULT",
      "ResourceId": "XXX",
      "TriggerAttribute": "single",
      "TriggerDesc": "{\"cron\":\"0 */2 * * * *\"}",
      "TriggerName": "TriggerName",
      "Type": "timer"
    },
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.ApiGateway	API网关触发器创建失败。
FailedOperation.Apigw	创建触发器失败。
FailedOperation.Cos	操作COS资源失败。
FailedOperation.CreateTrigger	当前函数状态无法进行此操作。
InternalServerError.ApiGateway	创建apigw触发器内部错误。
InternalServerError.Ckafka	ckafka接口失败。
InternalServerError.Cmq	删除cmq触发器失败。
InternalServerError.Cos	更新触发器失败。
InternalServerError.System	内部系统错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ApiGateway	ApiGateway参数错误。
InvalidParameterValue.Cdn	Cdn传入错误。
InvalidParameterValue.Ckafka	Ckafka传入错误。
InvalidParameterValue.Cos	Cos传入错误。
InvalidParameterValue.CosNotifyRuleConflict	COS通知规则冲突。
InvalidParameterValue.CustomArgument	CustomArgument参数长度超限。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.EASConfig	EASConfig参数异常。
InvalidParameterValue.Enable	Enable取值与规范不符，请修正后再试。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.SecretInfo	SecretInfo错误。
InvalidParameterValue.ServiceName	ServiceName命名不规范。
InvalidParameterValue.TriggerDesc	TriggerDesc传入参数错误。
InvalidParameterValue.TriggerName	TriggerName传入错误。
InvalidParameterValue.Type	Type传入错误。
InvalidParameterValue.ZipFile	检测到不是标准的zip文件，请重新压缩后再试。
LimitExceeded.Cdn	Cdn使用超过最大限制。

错误码	描述
LimitExceeded.FunctionOnTopic	同一个主题下的函数超过最大限制。
LimitExceeded.Trigger	触发器数量超出最大限制，可通过 提交工单 申请提升限制。
ResourceInUse	资源被占用。
ResourceInUse.Cdn	Cdn已被占用。
ResourceInUse.Cmq	Cmq已被占用。
ResourceInUse.Cos	Cos已被占用。
ResourceInUse.Trigger	TriggerName已存在。
ResourceInUse.TriggerName	TriggerName已存在。
ResourceInsufficient.COS	COS资源不足。
ResourceNotFound	资源不存在。
ResourceNotFound.Cdn	Cdn不存在。
ResourceNotFound.Ckafka	Ckafka不存在。
ResourceNotFound.Cmq	Cmq不存在。
ResourceNotFound.Cos	Cos不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Qualifier	版本不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.CreateTrigger	没有权限。
UnsupportedOperation	操作不支持。
UnsupportedOperation.Cdn	Cdn不支持。
UnsupportedOperation.Cos	Cos操作不支持。
UnsupportedOperation.Trigger	Trigger操作不支持。

删除触发器

最近更新时间：2025-05-23 01:27:22

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据参数传入删除已有的触发方式。

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：DeleteTrigger。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数的名称 示例值：ledDummyAPITest
TriggerName	是	String	要删除的触发器名称 示例值：trigger_name
Type	是	String	要删除的触发器类型，目前只支持 timer、kafka、apigw、cls、cos、cmq、http 类型 示例值：timer
Namespace	否	String	函数所属命名空间 示例值：default
TriggerDesc	否	String	如果删除的触发器类型为 COS 触发器，该字段为必填值，存放 JSON 格式的数据 {"event": "cos:ObjectCreated:*"}，数据内容和 SetTrigger 接口中该字段的格式相同；如果删除的触发器类型为定时触发器或 CMQ 触发器，可以不指定该字段 示例值：{"event": "cos:ObjectCreated:*"}
Qualifier	否	String	要删除的触发器实际所指向的版本或别名，默认值为 \$LATEST 示例值：\$LATEST

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 删除触发器

用户使用该函数删除触发器。

输入示例

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

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

{
  "Type": "timer",
  "FunctionName": "ledDummyAPITest",
  "TriggerName": "test3"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.CreateTrigger	当前函数状态无法进行此操作。
FailedOperation.DeleteTrigger	删除触发器失败。
FailedOperation.DeleteTrigger.UrlUsed	Url被占用，无法删除。
InternalServerError	内部错误。
InternalServerError.System	内部系统错误。
InvalidParameterValue	参数取值错误。

错误码	描述
InvalidParameterValue.ApiGateway	ApiGateway参数错误。
InvalidParameterValue.Cdn	Cdn传入错误。
InvalidParameterValue.Cmq	Cmq传入错误。
InvalidParameterValue.Cos	Cos传入错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.TriggerDesc	TriggerDesc传入参数错误。
InvalidParameterValue.TriggerName	TriggerName传入错误。
InvalidParameterValue.Type	Type传入错误。
ResourceInUse.Cdn	Cdn已被占用。
ResourceInUse.Cmq	Cmq已被占用。
ResourceNotFound	资源不存在。
ResourceNotFound.Cdn	Cdn不存在。
ResourceNotFound.Cmq	Cmq不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Timer	Timer不存在。
ResourceNotFound.Trigger	触发器不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.DeleteTrigger	没有权限。
UnsupportedOperation.Cdn	Cdn不支持。

更新触发器状态

最近更新时间：2025-04-25 01:51:52

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

更新触发器状态的值

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：UpdateTriggerStatus。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Enable	是	String	触发器的初始是能状态OPEN表示开启 CLOSE表示关闭 示例值：CLOSE
FunctionName	是	String	函数名称 示例值：function-demo
TriggerName	是	String	触发器名称 示例值：SCF-timer-1735218857
Type	是	String	触发器类型，触发器类型，目前只支持 timer、cos、ckafka三种类型 示例值：timer
Qualifier	否	String	触发器在创建时所指向的触发别名或版本，默认值为\$LATEST 示例值：\$LATEST
Namespace	否	String	函数所在的命名空间，默认值为default 示例值：default
TriggerDesc	否	String	如果更新的触发器类型为 COS 触发器，该字段为必填值，存放 JSON 格式的数据 {"event": "cos:ObjectCreated:*"}，数据内容和 SetTrigger 接口中该字段的格式相同；如果更新的触发器类型为定时触发器或 CMQ 触发器，可以不指定该字段 示例值：0 0 0 */1 * * *

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 修改触发器状态

输入示例

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

```
{
  "Enable": "CLOSE",
  "Type": "timer",
  "FunctionName": "<FunctionName>",
  "TriggerName": "<TriggerName>"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "375d4028-2666-4cef-bcab-2331cb974fc3"
  }
}
```

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。
FailedOperation.Cos	操作COS资源失败。
FailedOperation.CreateTrigger	当前函数状态无法进行此操作。
FailedOperation.UpdateStatus	更新状态操作失败。

错误码	描述
FailedOperation.UpdateTriggerStatus	与原状态一致，无需更改。
InternalServerError.Cmq	删除cmq触发器失败。
InternalServerError.Cos	更新触发器失败。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.CosNotifyRuleConflict	COS通知规则冲突。
InvalidParameterValue.Enable	Enable取值与规范不符，请修正后再试。
InvalidParameterValue.TriggerDesc	TriggerDesc传入参数错误。
InvalidParameterValue.TriggerName	TriggerName传入错误。
InvalidParameterValue.Type	Type传入错误。
ResourceNotFound	资源不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Qualifier	版本不存在。
ResourceNotFound.Trigger	触发器不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.UpdateTriggerStatus	没有权限。

获取函数触发器列表

最近更新时间：2025-04-25 01:51:53

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

获取函数触发器列表

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：ListTriggers。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：function-demo
Namespace	否	String	命名空间，默认是default 示例值：default
Offset	否	Integer	数据偏移量，默认值为 0 示例值：0
Limit	否	Integer	返回数据长度，默认值为 20 示例值：1
OrderBy	否	String	根据哪个字段进行返回结果排序,支持以下字段：add_time，mod_time，默认mod_time 示例值：add_time
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC 和 DESC，默认DESC 示例值：ASC
Filters.N	否	Array of Filter	* Qualifier: 函数版本，别名 * TriggerName: 函数触发器名称 * Description: 函数触发器描述 示例值：Qualifier: \$LATEST

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	触发器总数 示例值：1
Triggers	Array of TriggerInfo	触发器列表 示例值：Triggers
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 获取函数触发器列表

获取函数触发器列表

输入示例

```
https://scf.tencentcloudapi.com/?Action=ListTriggers
&FunctionName=function-demo
&Limit=2
&Order=ASC
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "Triggers": [
      {
        "Enable": 1,
        "Qualifier": "$DEFAULT",
        "TriggerName": "hulrzbh9f",
        "Type": "http",
        "TriggerDesc": "{\n\"AuthType\": \"NONE\", \"NetConfig\": {\n\"EnableIntranet\": true, \"IntranetUrl\": \"https://1258888888-hulrzbh9f.in.ap-chongqing.tencentscf.com\", \"IntranetHTTPEndpoint\": \"http://1258888888-hulrzbh9f.in.ap-chongqing.tencentscf.com\", \"ExtranetUrl\": \"\", \"ExtranetHTTPEndpoint\": \"\", \"WssIntranetUrl\": \"\", \"WsIntranetUrl\": \"\", \"WssExtranetUrl\": \"\", \"WsExtranetUrl\": \"\", \"EnableExtranet\": false}}",
        "AvailableStatus": "Available",
        "CustomArgument": "custom",
        "Description": "Some Description",
        "AddTime": "2020-09-22 00:00:00",
        "ModTime": "2020-09-22 00:00:00",
        "ResourceId": "qcs:scf:cq:uin/888888:function/function-demo/hulrzbh9f",
        "BindStatus": "on",
        "TriggerAttribute": "single"
      }
    ],
    "RequestId": "52dd00b1-7197-4a11-97f5-2a67f9f49273"
  }
}
```

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.Apigw	创建触发器失败。
InvalidParameterValue.Filters	Filters参数错误。
InvalidParameterValue.Order	Order传入错误。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。可参考对应接口文档中的示例值： https://cloud.tencent.com/document/product/583/17235
InvalidParameterValue.Orderby	Orderby传入错误。

更新云函数自定义域名

最近更新时间：2025-05-23 01:27:21

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

更新自定义域名相关配置

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：UpdateCustomDomain。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Domain	是	String	自定义域名 示例值：www.demo.com
Protocol	否	String	协议，取值范围：HTTP, HTTPS, HTTP&HTTPS 示例值：HTTP
CertConfig	否	CertConf	证书配置信息，HTTPS协议必穿
WafConfig	否	WafConf	web 应用防火墙配置
EndpointsConfig.N	否	Array of EndpointsConf	路由配置

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 示例1

更新自定义域名的证书Id,并关闭waf

输入示例

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

{
  "Domain": "www.ccc.com",
```

```
"Protocol": "HTTP&HTTPS",
"CertConfig": {
  "CertificateId": "cvvxxJ3DK7"
},
"WafConfig": {
  "WafOpen": "CLOSE",
  "WafInstanceId": "waf_2kzh3vlq0um"
}
}
```

输出示例

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

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.Ssl	更新自定义域名失败，SSL异常。

获取云函数自定义域名列表

最近更新时间：2025-04-25 01:51:53

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

遍历域名列表信息

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：ListCustomDomains。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Offset	否	Integer	偏移量，默认0 示例值：0
Limit	否	Integer	容量，默认20 示例值：10
OrderBy	否	String	取值范围：AddTime，ModTime，默认AddTime 示例值：AddTime
Order	否	String	取值范围：DESC，ASC 默认DESC 示例值：DESC
Filters.N	否	Array of Filter	过滤条件

3. 输出参数

参数名称	类型	描述
Total	Integer	总数 示例值：1
Domains	Array of DomainInfo	域名列表信息 注意：此字段可能返回 null，表示取不到有效值。
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 示例1

输入示例

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

```
X-TC-Action: ListCustomDomains
```

<公共请求参数>

```
{}
```

输出示例

```
{
  "Response": {
    "Domains": [
      {
        "CertConfig": {
          "CertificateId": "ccccaas"
        },
        "Domain": "www.demo.com",
        "EndpointsConfig": [
          {
            "FunctionName": "test1",
            "Namespace": "default",
            "PathMatch": "/cc/",
            "PathRewrite": [
              {
                "Path": "/cc/",
                "Rewrite": "/$1",
                "Type": "ExactRules"
              }
            ],
            "Qualifier": "$LATEST"
          }
        ],
        "Protocol": "HTTP&HTTPS",
        "WafConfig": {
          "WafInstanceId": "",
          "WafOpen": "CLOSE"
        }
      }
    ],
    "RequestId": "6941ssdd162b-925f-4c3a-b90f-3558c0aebf97",
    "Total": 1
  }
}
```

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

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查看云函数自定义域名详情

最近更新时间：2025-05-23 01:27:22

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

查看云函数自定义域名详情

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：GetCustomDomain。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Domain	否	String	域名 示例值：www.demo.com

3. 输出参数

参数名称	类型	描述
Domain	String	域名 示例值：www.demo.com
Protocol	String	协议 示例值：HTTP
EndpointsConfig	Array of EndpointsConf	路由配置
CertConfig	CertConf	证书配置信息
WafConfig	WafConf	web 应用防火墙配置
Tags	Array of Tag	标签
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 示例1

输入示例

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

```
{
  "Domain": "www.demo.com"
}
```

输出示例

```
{
  "Response": {
    "CertConfig": {
      "CertificateId": "BfHJsadd7"
    },
    "Domain": "www.demo.com",
    "EndpointsConfig": [
      {
        "FunctionName": "test1",
        "Namespace": "default",
        "PathMatch": "/cc/",
        "PathRewrite": [
          {
            "Path": "/cc/",
            "Rewrite": "/$1",
            "Type": "ExactRules"
          }
        ],
        "Qualifier": "$LATEST"
      }
    ],
    "Protocol": "HTTP&HTTPS",
    "RequestId": "6941ssdd162b-925f-4c3a-b90f-3558c0aebf97",
    "WafConfig": {
      "WafInstanceId": "",
      "WafOpen": "CLOSE"
    }
  }
}
```

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.Domain.UnExist	自定义域名不存在

删除云函数自定义域名

最近更新时间：2025-04-25 01:51:53

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

删除自定义域名

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：DeleteCustomDomain。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Domain	否	String	域名 示例值：www.demo.com

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 示例1

输入示例

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

{
  "Domain": "www.xxx.com"
}
```

输出示例

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

```
}  
}
```

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

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建云函数自定义域名

最近更新时间：2025-05-23 01:27:22

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

创建自定义域名

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：CreateCustomDomain。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Domain	是	String	域名，不支持泛域名 示例值：www.xxx.com
Protocol	是	String	协议，取值范围：HTTP, HTTPS, HTTP&HTTPS 示例值：HTTP
EndpointsConfig.N	是	Array of EndpointsConf	路由配置
CertConfig	否	CertConf	证书配置信息，有使用HTTPS协议时候必须传
WafConfig	否	WafConf	web 应用防火墙配置
Tags.N	否	Array of Tag	标签

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 示例1

创建一个自定义域名，关闭了waf防护，路由配置开启重写策略，该策略是通配符匹配替换。

输入示例

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

{
```

```
"Domain": "www.ccc.com",
"Protocol": "HTTP&HTTPS",
"CertConfig": {
  "CertificateId": "BxxJ3DK7"
},
"EndpointsConfig": [
  {
    "PathMatch": "/aa/*",
    "Namespace": "default",
    "FunctionName": "test1",
    "Qualifier": "$LATEST",
    "PathRewrite": [
      {
        "Path": "/aa/bb/*",
        "Type": "WildcardRules",
        "Rewrite": "/$"
      }
    ]
  }
],
"WafConfig": {
  "WafOpen": "CLOSE",
  "WafInstanceId": "waf_2kzh3v1q0umkkha9"
}
```

输出示例

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

示例2 示例2

创建一个自定义域名，开启了waf 防护，路由配置开启重写策略，该策略是精确路径匹配替换。

输入示例

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

{
  "Domain": "www.ccc.com",
  "Protocol": "HTTP&HTTPS",
  "CertConfig": {
    "CertificateId": "BxxJ3DK7"
  },
  "EndpointsConfig": [
    {
      "PathMatch": "/aa/",
      "Namespace": "default",
      "FunctionName": "test1",
```



```
"Qualifier": "1",
"PathRewrite": [
{
"Path": "/aa/",
"Type": "WildcardRules",
"Rewrite": "/$"
}
],
"WafConfig": {
"WafOpen": "OPEN",
"WafInstanceId": "waf_2kzh3v1q0umkkha9"
}
}
```

输出示例

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

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
InvalidParameterValue.CertConfig.empty	CertConfig配置为空。

更新触发器

最近更新时间：2025-05-23 01:27:21

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

支持触发器配置更新。

默认接口请求频率限制：20次/秒

注意：目前只支持timer触发器和ckafka触发器更新！

timer触发器和ckafka触发器支持更新字段有：Enable、TriggerDesc、Description、CustomArgument。

timer触发器TriggerDesc支持5段式和7段式的更新。

ckafka触发器TriggerDesc支持Retry、MaxMsgNum、TimeOut参数更新，不传值表示原值不变，传值不能为空。

Enable 触发器开启或关闭，传参为OPEN为开启，CLOSE为关闭。不传值表示原值不变，传值不能为空。

Description 触发器描述，不传值保持原值不变，传值为空则为空。

CustomArgument 触发器用户附加信息（注意：只有timer触发器展示），不传值保持原值不变，传值为空则为空。

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：UpdateTrigger。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：FuncName
TriggerName	是	String	触发器名称 示例值：TriggerName
Type	是	String	触发器类型，目前只支持timer、ckafka、http三种类型 示例值：timer
Enable	否	String	触发器开启或关闭，传参为OPEN为开启，CLOSE为关闭 示例值：OPEN
Qualifier	否	String	触发器创建时所指向的函数版本或别名，默认为 \$LATEST 示例值：\$DEFAULT
Namespace	否	String	函数的命名空间，默认值为default 示例值：default
TriggerDesc	否	String	TriggerDesc参数 示例值：*/1*****
Description	否	String	触发器描述 示例值：some description
CustomArgument	否	String	用户附加信息 示例值：custom information

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 接口调用示例

接口调用示例

输入示例

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

{
  "Enable": "CLOSE",
  "FunctionName": "helloworld-xxxxxxx",
  "TriggerName": "\t\nSCF-timer-xxxxxxx",
  "Type": "timer",
  "Qualifier": "$DEFAULT",
  "Namespace": "default",
  "TriggerDesc": "0 0 0 */2 * * *",
  "Description": "some description",
  "CustomArgument": "中文测试"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "375d4028-xxxx-xxxxf-xxxx-2331cb974fc3"
  }
}
```

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。
FailedOperation.CreateTrigger	当前函数状态无法进行此操作。
FailedOperation.UpdateTrigger	更新触发器出错。
InternalServerError.Cmq	删除cmq触发器失败。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Action	Action取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17235 。
InvalidParameterValue.Cors	Cors 字段传值异常。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.Enable	Enable取值与规范不符，请修正后再试。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.TriggerDesc	TriggerDesc传入参数错误。
InvalidParameterValue.TriggerName	TriggerName传入错误。
InvalidParameterValue.Type	Type传入错误。
ResourceNotFound	资源不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Qualifier	版本不存在。
ResourceNotFound.Trigger	触发器不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

函数相关接口

创建函数版本别名

最近更新时间：2025-04-25 01:51:59

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

为某个函数版本创建一个别名，您可以使用别名来标记特定的函数版本，如DEV/RELEASE版本，也可以随时修改别名指向的版本。
一个别名必须指向一个主版本，此外还可以同时指向一个附加版本。调用函数时指定特定的别名，则请求会被发送到别名指向的版本上，您可以配置请求发送到主版本和附加版本的比例。
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：CreateAlias。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Name	是	String	别名的名称，在函数级别中唯一，只能包含字母、数字、'_'和 '-'，且必须以字母开头，长度限制为1-64 示例值：TestAlias
FunctionName	是	String	函数名称 示例值：TestFunction
FunctionVersion	是	String	别名指向的主版本 示例值：\$LATEST
Namespace	否	String	函数所在的命名空间 示例值：default
RoutingConfig	否	RoutingConfig	别名的路由信息，需要为别名指定附加版本时，必须提供此参数；附加版本指的是：除主版本 FunctionVersion 外，为此别名再指定一个函数可正常使用的版本；这里附加版本中的 Version 值 不能是别名指向的主版本；要注意的是：如果想要某个版本的流量全部指向这个别名，不需配置此参数；目前一个别名最多只能指定一个附加版本 示例值：{ "AdditionalVersionWeights": [{ "Version": "2", "Weight": 0.01 }] }
Description	否	String	别名的描述信息 示例值：a alias1

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 创建一个别名

输入示例

```
https://scf.tencentcloudapi.com/?Action=CreateAlias
&Name=TestAlias
&Namespace=default
&FunctionName=TestFunction
&FunctionVersion=$LATEST
&RoutingConfig.AdditionalVersionWeights.0.Version=1
&RoutingConfig.AdditionalVersionWeights.0.Weight=0.5
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.CreateAlias	创建别名失败。
InternalServerError.System	内部系统错误。
InvalidParameter.RoutingConfig	RoutingConfig参数传入错误。
InvalidParameterValue.AdditionalVersionWeights	AdditionalVersionWeights参数传入错误。
InvalidParameterValue.Description	Description传入错误。

错误码	描述
InvalidParameterValue.Name	Name参数传入错误。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.RoutingConfig	RoutingConfig取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#RoutingConfig 。
LimitExceeded.Alias	别名个数超过最大限制。
ResourceInUse	资源被占用。
ResourceInUse.Alias	Alias已被占用。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

删除别名

最近更新时间：2025-04-25 01:51:58

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

删除一个函数版本的别名

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：DeleteAlias。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：<FunctionName>
Name	是	String	别名的名称 示例值：<AliasName>
Namespace	否	String	函数所在的命名空间 示例值：<Namespace>

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 删除别名

输入示例

```
https://scf.tencentcloudapi.com/?Action=DeleteAlias
&Namespace=<Namespace>
&FunctionName=<FunctionName>
&Name=<AliasName>
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "d1b93f9c-ac3a-412a-a4f3-6f0697099f72"
```



```
}  
  
}
```

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.DeleteAlias	删除别名失败。
InternalServerError.System	内部系统错误。
InvalidParameterValue.Alias	不支持删除默认别名，请修正后重试。
InvalidParameterValue.Name	Name参数传入错误。
ResourceNotFound.Alias	别名不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。

删除函数指定版本

最近更新时间：2025-04-25 01:51:58

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入参数删除函数的指定版本。

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：DeleteFunctionVersion。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	要删除的函数名称 示例值：test_function
Qualifier	是	String	填写需要删除的版本号 示例值：1
Namespace	否	String	函数所属命名空间 示例值：default
ForceDelete	否	String	强制删除标记，传true会直接删除容器，并强制关闭还在执行中的函数 示例值：true

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 删除函数指定版本

删除函数 test 的版本 2

输入示例

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

{
  "FunctionName": "functionName1",
```

```
"Namespace": "default",
"Qualifier": "2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  }
}
```

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。
FailedOperation.DeleteFunction	当前函数状态无法进行此操作，请在函数状态正常时重试。
FailedOperation.FunctionNameStatusError	当前函数状态无法更新代码，请在状态为正常时更新。
FailedOperation.FunctionStatusError	函数在部署中,无法做此操作。
FailedOperation.ProvisionedInProgress	当前函数版本已有预置任务处于进行中，请稍后重试。
InternalServerError.Cmq	删除cmq触发器失败。
InternalServerError.GetStsTokenFailed	获取sts票据信息失败。
InternalServerError.System	内部系统错误。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。

错误码	描述
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
ResourceNotFound	资源不存在。
ResourceNotFound.AttachedTagKeyNotFound	函数需要关联的标签Key不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Qualifier	版本不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.DeleteFunction	没有权限的操作。
UnsupportedOperation	操作不支持。
UnsupportedOperation.AliasBind	资源还有别名绑定，不支持当前操作，请解绑别名后重试。

删除函数

最近更新时间：2025-04-25 01:51:58

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入参数删除函数。

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：DeleteFunction。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	要删除的函数名称 示例值：<FunctionName>
Namespace	否	String	函数所属命名空间 示例值：default
Qualifier	否	String	填写需要删除的版本号，不填默认删除函数下全部版本。 示例值：\$LATEST

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 删除函数

用户删除函数时需要使用，可指定版本与命名空间。

输入示例

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

{
  "FunctionName": "<FunctionName>"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.DeleteFunction	当前函数状态无法进行此操作，请在函数状态正常时重试。
FailedOperation.FunctionNameStatusError	当前函数状态无法更新代码，请在状态为正常时更新。
FailedOperation.FunctionStatusError	函数在部署中,无法做此操作。
FailedOperation.ProvisionedInProgress	当前函数版本已有预置任务处于进行中，请稍后重试。
InternalServerError.Cmq	删除cmq触发器失败。
InternalServerError.GetStsTokenFailed	获取sts票据信息失败。
InternalServerError.System	内部系统错误。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
ResourceNotFound	资源不存在。

错误码	描述
ResourceNotFound.AttachedTagKeyNotFound	函数需要关联的标签Key不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Qualifier	版本不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.DeleteFunction	没有权限的操作。
UnsupportedOperation	操作不支持。
UnsupportedOperation.AliasBind	资源还有别名绑定，不支持当前操作，请解绑别名后重试。

删除预置并发

最近更新时间：2025-04-25 01:51:58

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

删除函数版本的预置并发配置。

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：DeleteProvisionedConcurrencyConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	需要删除预置并发的函数的名称 示例值：functionName1
Qualifier	是	String	函数的版本号 示例值：1
Namespace	否	String	函数所属命名空间，默认为default 示例值：namespace1

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 删除预置并发

输入示例

```
https://scf.tencentcloudapi.com/?Action=DeleteProvisionedConcurrencyConfig
&FunctionName=functionName1
&Namespace=namespace1
&Qualifier=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```



```
}
}
```

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。
FailedOperation.ProvisionedInProgress	当前函数版本已有预置任务处于进行中，请稍后重试。
InternalError	内部错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Qualifier	不支持的函数版本。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation	未授权操作。

删除函数最大独占配额

最近更新时间：2025-04-25 01:51:58

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

删除函数的最大独占配额配置。

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：DeleteReservedConcurrencyConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	需要删除最大独占配额的函数的名称 示例值：functionName1
Namespace	否	String	函数所属命名空间，默认为default 示例值：default

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 删除函数最大独占配额

输入示例

```
https://scf.tencentcloudapi.com/?Action=DeleteReservedConcurrencyConfig
&FunctionName=functionName1
&Namespace=default
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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.DebugModeStatus	当前调试状态无法执行此操作。
FailedOperation.ReservedInProgress	当前函数已有保留并发设置任务处于进行中，请稍后重试。
InternalServerError	内部错误。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation	未授权操作。

获取别名详细信息

最近更新时间：2025-04-25 01:51:58

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

获取别名的详细信息，包括名称、描述、版本、路由信息等。

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：GetAlias。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：functionName1
Name	是	String	别名的名称 示例值：alias1
Namespace	否	String	函数所在的命名空间 示例值：ns1

3. 输出参数

参数名称	类型	描述
FunctionVersion	String	别名指向的主版本 示例值：1
Name	String	别名的名称 示例值：alias1
RoutingConfig	RoutingConfig	别名的路由信息
Description	String	别名的描述 示例值：alias1
AddTime	Timestamp	创建时间 示例值："2024-12-19 12:19:07"
ModTime	Timestamp	更新时间 示例值："2024-12-19 12:19:07"
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 查询别名详细信息

输入示例

```
https://scf.tencentcloudapi.com/?Action=GetAlias
&Namespace=<Namespace>
&FunctionName=<FunctionName>
&Name=<AliasName>
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3c140219-cfe9-470e-b241-907877d6fb03",
    "Name": "ddd",
    "Description": "",
    "FunctionVersion": "1",
    "RoutingConfig": {
      "AdditionalVersionWeights": [
        {
          "Version": "$LATEST",
          "Weight": 0.8
        }
      ],
      "AddtionVersionMatchs": []
    },
    "AddTime": "2024-12-19 14:53:51",
    "ModTime": "2024-12-19 14:53:51"
  }
}
```

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.GetAlias	获取别名信息失败。
InternalServerError.System	内部系统错误。
InvalidParameter.RoutingConfig	RoutingConfig参数传入错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Name	Name参数传入错误。
InvalidParameterValue.Namespace	Namespace参数传入错误。
ResourceNotFound.Alias	别名不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

获取函数代码下载地址

最近更新时间：2025-04-25 01:51:57

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口用于获取函数代码包的下载地址。

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：GetFunctionAddress。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数的名称 示例值：<FunctionName>
Qualifier	否	String	函数的版本 示例值：\$LATEST
Namespace	否	String	函数的命名空间 示例值：default

3. 输出参数

参数名称	类型	描述
Url	String	函数的Cos地址 示例值：https://xxxx.com/xxxx
CodeSha256	String	函数的SHA256编码 示例值：fdsfdgdfgdfghdfhdfgsfsrewreterytdx
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 获取函数代码下载地址

输入示例

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

{
```

```
"FunctionName": "<FunctionName>"
}
```

输出示例

```
{
  "Response": {
    "Url": "http://lambda-code-1253970226.cosgz.myqcloud.com/1253970226/ledDummyAPITest/ledDummyAPITest_LATEST.zip?sign=RwvyBS5VPXz75hYNCJKHrSUHBqBhPTEyNTM5NzAyMjYmaz1BS01EQ2dPUFdqUTZCQXh2SHR5Y2toV0FCS1ZZU0JqNTQ4cE4mZT0wJnQ9MTUyOTg1MTQxNSZyPTY5NzYyMTk3NCZmPS8xMjUzOTcwMjI2L2x1ZER1bW15QVBJVGZzdC9sZWREdW1teUFQSVRlc3RfTEFURVNULnppcCZiPWxhbWJkYWVWNvZGU=",
    "CodeSha256": "5077f5203547b225b532434dd6fcf7c9897025d7581a9f3d474a7afee88bc696",
    "RequestId": "a90cb10a-7ad4-4270-b92d-ba0319be6dbd"
  }
}
```

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.FunctionStatusError	函数在部署中,无法做此操作。
FailedOperation.GetFunctionAddress	获取函数代码地址失败。
InternalServerError	内部错误。
InternalServerError.System	内部系统错误。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Action	Action取值与规范不符,请修正后再试。可参考: https://cloud.tencent.com/document/product/583/17235 。
InvalidParameterValue.FunctionName	函数不存在。

错误码	描述
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.CodeSecret	无访问代码权限。

获取函数异步重试配置

最近更新时间：2025-04-25 01:51:57

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

获取函数异步重试配置，包括重试次数和消息保留时间

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：GetFunctionEventInvokeConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：Unzip-1609933864
Namespace	否	String	函数所属命名空间，默认为default 示例值：default
Qualifier	否	String	函数版本，默认为\$LATEST 示例值：\$LATEST

3. 输出参数

参数名称	类型	描述
AsyncTriggerConfig	AsyncTriggerConfig	异步重试配置信息
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 获取异步配置请求示例

输入示例

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

{
  "FunctionName": "Unzip-1609933864",
  "Namespace": "default",
```

```
"Qualifier": "$LATEST"
}
```

输出示例

```
{
  "Response": {
    "AsyncTriggerConfig": {
      "MsgTTL": 3600,
      "RetryConfig": [
        {
          "RetryNum": 1
        },
        {
          "RetryNum": -1
        }
      ]
    },
    "RequestId": "435fbf8e-d512-4f03-baf3-9cf1688f6c98"
  }
}
```

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。
FailedOperation.FunctionVersionStatusNotActive	当前函数版本状态无法进行此操作，请在版本状态为正常时重试。
FailedOperation.InsufficientBalance	当前账号或命名空间处于欠费状态，请在可用时重试。
InternalError	内部错误。

错误码	描述
InvalidParameterValue	参数取值错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
MissingParameter	缺少参数错误。
ResourceNotFound	资源不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation	未授权操作。

获取函数预置并发详情

最近更新时间：2025-04-25 01:51:57

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

获取函数或函数某一版本的预置并发详情。

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：GetProvisionedConcurrencyConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	需要获取预置并发详情的函数名称。 示例值：funcName1
Namespace	否	String	函数所在的命名空间，默认为default。 示例值：default
Qualifier	否	String	函数版本号，不传则返回函数所有版本的预置并发信息。 示例值：1

3. 输出参数

参数名称	类型	描述
UnallocatedConcurrencyNum	Integer	该函数剩余可配置的预置并发数。 示例值：1
Allocated	Array of VersionProvisionedConcurrencyInfo	函数已预置的并发配置详情。
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 获取函数预置并发详情

输入示例

```
https://scf.tencentcloudapi.com/?Action=GetProvisionedConcurrencyConfig
&FunctionName=functionName1
&Namespace=default
&Qualifier=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "UnallocatedConcurrencyNum": 2343,
    "Allocated": [
      {
        "Qualifier": "1",
        "AvailableProvisionedConcurrencyNum": 1,
        "AllocatedProvisionedConcurrencyNum": 1,
        "Status": "Done",
        "StatusReason": "",
        "TriggerActions": []
      },
      {
        "Qualifier": "2",
        "AvailableProvisionedConcurrencyNum": 1,
        "AllocatedProvisionedConcurrencyNum": 1,
        "Status": "Done",
        "StatusReason": "",
        "TriggerActions": [
          {
            "TriggerName": "timer-1",
            "ProvisionedType": "Default",
            "TriggerProvisionedConcurrencyNum": 1,
            "TriggerCronConfig": "30 42 15 20 12 * 2024"
          }
        ]
      }
    ],
    "RequestId": "dsdsfsdf-b246-4fe5-812a-f77f1275471a"
  }
}
```

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.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
MissingParameter	缺少参数错误。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation	未授权操作。

获取函数单个请求运行状态

最近更新时间：2025-04-25 01:51:57

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据指定的查询条件返回函数单个请求运行状态。

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：GetRequestStatus。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：hello
FunctionRequestId	是	String	需要查询状态的请求 id 示例值：a63607c3-581f-4be9-8c4b-631c75909f07
Namespace	否	String	函数的所在的命名空间 示例值：default
StartTime	否	String	查询的开始时间，例如：2017-05-16 20:00:00，不填默认为当前时间 - 15min 示例值：2017-05-16 20:00:00
EndTime	否	String	查询的结束时间，例如：2017-05-16 20:59:59。StartTime 为空时，EndTime 默认为当前时间；StartTime 有值时，需要同时传 EndTime。EndTime 需要晚于 StartTime。 示例值：2017-05-16 20:59:59

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	函数运行状态的总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：10
Data	Array of RequestStatus	函数运行状态数组 注意：此字段可能返回 null，表示取不到有效值。
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 获取函数运行状态

输入示例


```
https://scf.tencentcloudapi.com/?Action=GetRequestStatus
&FunctionName=<FunctionName>
&FunctionRequestId=<FunctionRequestId>
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 0,
    "Data": [
      {
        "FunctionName": "function-demo",
        "RetMsg": "Hello",
        "RequestId": "96d91886-f49b-4962-825e-1b4b8a2585f7",
        "StartTime": "2024-12-25 19:50:07",
        "RetCode": 0,
        "Duration": 0,
        "MemUsage": 0,
        "RetryNum": 0
      }
    ],
    "RequestId": "a63607c3-581f-4be9-8c4b-631c75909f07"
  }
}
```

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。

错误码	描述
FailedOperation.QueryError	查询失败。
FailedOperation.TopicNotExist	Topic不存在。
InternalServerError	内部错误。
InternalServerError.System	内部系统错误。
InvalidParameter.Cls	日志参数有误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.DateTime	DateTime传入错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.FunctionRequestId	请求 id 传参错误。
InvalidParameterValue.RetCode	RetCode不合法。
InvalidParameterValue.StartTime	起始时间传入错误。
InvalidParameterValue.StartTimeOrEndTime	需要同时指定开始日期与结束日期。
OperationDenied.AccountIsolate	账号已被隔离。
OperationDenied.AccountNotExists	账号不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。

获取函数最大独占配额详情

最近更新时间：2025-04-25 01:51:57

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

获取函数的最大独占配额详情。

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：GetReservedConcurrencyConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	需要获取最大独占配额详情的函数名称。 示例值：functionName1
Namespace	否	String	函数所在的命名空间，默认为default。 示例值：default

3. 输出参数

参数名称	类型	描述
ReservedMem	Integer	该函数的最大独占配额。 注意：此字段可能返回 null，表示取不到有效值。 示例值：128
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 获取函数最大独占配额详情

输入示例

```
https://scf.tencentcloudapi.com/?Action=GetReservedConcurrencyConfig
&FunctionName=functionName1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397",
    "ReservedMem": 128
  }
}
```

```
}  
}
```

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.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
MissingParameter	缺少参数错误。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation	未授权操作。

同步Invoke调用接口

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

SCF同步调用函数接口。

默认接口请求频率限制：10000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：InvokeFunction。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
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-chengdu, ap-chongqing, ap-guangzhou, ap-hongkong, ap-jakarta, ap-nanjing, ap-seoul, ap-shanghai, ap-shanghai-fsi, ap-shenzhen-fsi, ap-singapore, ap-tokyo, eu-frankfurt, na-ashburn, na-siliconvalley, sa-saopaulo。
FunctionName	是	String	函数名称 示例值：helloworld
Qualifier	否	String	触发函数的版本号或别名，默认值为\$DEFAULT 示例值：\$LATEST
Event	否	String	运行函数时的参数，以json格式传入，最大支持的参数长度是 6MB。该字段信息对应函数 event 入参。 示例值：{"key":"value"}
LogType	否	String	返回值会包含4KB的日志，可选值为None和Tail，默认值为None。当该值为Tail时，返回参数中的Log字段会包含对应的函数执行日志 示例值：Tail
Namespace	否	String	命名空间，不填默认为 default 示例值：default
RoutingKey	否	String	函数灰度流量控制调用，以json格式传入，例如{"k":"v"}，注意kv都需要是字符串类型，最大支持的参数长度是1024字节 示例值：{"key":"value"}

3. 输出参数

参数名称	类型	描述
Result	Result	函数执行结果
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 运行函数

输入示例

```
https://scf.tencentcloudapi.com/?Action=InvokeFunction
&FunctionName=function-demo
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Result": {
      "MemUsage": 3207168,
      "Log": "",
      "RetMsg": "hello from scf",
      "BillDuration": 100,
      "FunctionRequestId": "6add56fa-58f1-11e8-89a9-5254005d5fdb",
      "Duration": 0.826,
      "ErrMsg": "",
      "InvokeResult": 0
    },
    "RequestId": "c2af8a64-c922-4d55-ae0-bd86a5c2cd12"
  }
}
```

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.FunctionStatusError	函数在部署中,无法做此操作。
FailedOperation.InvokeFunction	调用函数失败。
InternalError	内部错误。

错误码	描述
InternalServerError.System	内部系统错误。
InvalidParameter.FunctionName	FunctionName取值与规范不符，请修正后再试。
InvalidParameter.RequestTooLarge	请求大小超限。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.ClientContext	运行函数时的参数传入有误。
InvalidParameterValue.FunctionType	FunctionType参数错误。
InvalidParameterValue.Param	入参不是标准的json。
LimitExceeded.Quota	配额超限。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Qualifier	版本不存在。
ResourceUnavailable.InsufficientBalance	余额不足，请先充值。
ResourceUnavailable.ServiceSuspended	服务已冻结。
UnauthorizedOperation.CAM	CAM鉴权失败。

获取别名列表

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

返回一个函数下的全部别名，可以根据特定函数版本过滤。

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：ListAliases。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：functionName1
Namespace	否	String	函数所在的命名空间 示例值：default
FunctionVersion	否	String	如果提供此参数，则只返回与该函数版本有关联的别名 示例值：1
Offset	否	String	数据偏移量，默认值为 0 示例值：0
Limit	否	String	返回数据长度，默认值为 20 示例值：10

3. 输出参数

参数名称	类型	描述
Aliases	Array of Alias	别名列表
TotalCount	Integer	别名总数 示例值：1
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 查询函数下的全部别名

输入示例

```
https://scf.tencentcloudapi.com/?Action=ListAliases
&Namespace=default
```



```
&FunctionName=functionName1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Aliases": [
      {
        "ModTime": "2023-07-12 10:53:52",
        "AddTime": "2023-07-12 10:53:52",
        "Name": "$DEFAULT",
        "Description": "",
        "FunctionVersion": "$LATEST",
        "RoutingConfig": {
          "AdditionalVersionWeights": [],
          "AddtionVersionMatches": []
        }
      }
    ],
    "TotalCount": 1,
    "RequestId": "dfdfdfd-e3ca-4756-bc18-09a22ec9048f"
  }
}
```

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	内部系统错误。
InvalidParameterValue.FunctionName	函数不存在。

错误码	描述
InvalidParameterValue.Namespace	Namespace参数传入错误。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

获取函数列表

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入的查询参数返回相关函数信息。

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：ListFunctions。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC 和 DESC 示例值：ASC
Orderby	否	String	根据哪个字段进行返回结果排序,支持以下字段：AddTime, ModTime, FunctionName 示例值：ModTime
Offset	否	Integer	数据偏移量，默认为 0 示例值：0
Limit	否	Integer	返回数据长度，默认为 20 示例值：2
SearchKey	否	String	支持FunctionName模糊匹配 示例值：hello-world
Namespace	否	String	命名空间 示例值：default
Description	否	String	函数描述，支持模糊搜索 示例值：function1
Filters.N	否	Array of Filter	过滤特定属性或者有特定标签的函数。- 传值方式key-value 进行传值 例如："Filters": [{ "Name": "Status", "Values": ["CreateFailed","Creating"]}, {"Name": "Type", "Values": ["HTTP"]}]]上述条件的函数是，函数状态为创建失败或者创建中，且函数类型为 HTTP 函数如果通过标签进行过滤：- tag:tag-key - String - 是否必填：否 - （过滤条件）按照标签键值对进行过滤。tag-key使用具体的标签键进行替换。示例值："Filters": [{"Name":"tag-dmtest", "Values": ["dmtest"]}]]入参限制：1.每次请求的Filters的上限为10，Filter.Values的上限为5。2.[VpcId, 'SubnetId', 'ClsTopicId', 'ClsLogsetId', 'Role', 'CfsId', 'CfsMountInsId', 'Eip'] 过滤的Name 为这些属性时， values 只能传一个值3.['Status', 'Runtime', 'Type', 'PublicNetStatus', 'AsyncRunEnable', 'TraceEnable', 'Stamp'] 过滤的 Name 为这些属性时， values 可以传多个值 示例值：[{"Name":"Status", "Values":["CreateFailed", "Creating"]}]]

3. 输出参数

参数名称	类型	描述
Functions	Array of Function	函数列表

参数名称	类型	描述
TotalCount	Integer	总数 示例值：1
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 获取函数列表

输入示例

```
https://scf.tencentcloudapi.com/?Action=ListFunctions
&Limit=2
&Order=ASC
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "Functions": [
      {
        "Namespace": "default",
        "FunctionName": "functionName1",
        "FunctionId": "sdFdsgfd g",
        "Description": "",
        "Tags": [],
        "Runtime": "Nodejs8.9",
        "Type": "Event",
        "AsyncRunEnable": "FALSE",
        "TraceEnable": "FALSE",
        "ReservedConcurrencyMem": null,
        "TotalProvisionedConcurrencyMem": 0,
        "Status": "Active",
        "StatusDesc": "",
        "StatusReasons": [],
        "AddTime": "2024-08-27 15:27:43",
        "ModTime": "2024-08-27 15:27:43"
      }
    ],
    "RequestId": "sdfsdfdg-7d05-4d3e-91f8-e62e6695c8c8"
  }
}
```

示例2 根据标签查询函数

查询status标签的值是dev，且owner标签的值是me的函数

输入示例

```
https://scf.tencentcloudapi.com/?Action=ListFunctions
&Filters.0.Name=tag-status
&Filters.0.Values.0=dev
&Filters.1.Name=tag-owner
```

```
&Filters.1.Values.0=me
```

```
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "Functions": [
      {
        "Status": "Active",
        "FunctionName": "functionName1",
        "TotalProvisionedConcurrencyMem": 1,
        "Namespace": "default",
        "Runtime": "Python3.6",
        "Type": "HTTP",
        "ReservedConcurrencyMem": 1,
        "Description": "just for test",
        "TraceEnable": "FALSE",
        "AsyncRunEnable": "FALSE",
        "FunctionId": "lam-xxxxxxx",
        "StatusReasons": [],
        "StatusDesc": "",
        "AddTime": "2000-01-01 00:00:00",
        "ModTime": "2000-01-01 00:00:00",
        "Tags": [
          {
            "Value": "dev",
            "Key": "status"
          },
          {
            "Value": "me",
            "Key": "owner"
          }
        ],
        "RequestId": "77cce5f58-0ebb-4523-ab88-a37063443524"
      }
    ]
  }
}
```

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.Exception	内部服务异常。
InternalServerError.System	内部系统错误。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Limit	Limit传入错误。
InvalidParameterValue.LimitExceeded	参数超出长度限制。
InvalidParameterValue.Name	Name参数传入错误。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Offset	偏移量不合法。
InvalidParameterValue.Order	Order传入错误。
InvalidParameterValue.Orderby	Orderby传入错误。
InvalidParameterValue.QueryVersion	查询版本详情，版本参数传入错误。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.Region	Region错误。

查询函数版本

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入的参数查询函数的版本。

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：ListVersionByFunction。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名 示例值：functionName1
Namespace	否	String	函数所在命名空间 示例值：default
Offset	否	Integer	数据偏移量，默认值为 0 示例值：0
Limit	否	Integer	返回数据长度，默认值为 20 示例值：10
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC 和 DESC 示例值：DESC
OrderBy	否	String	根据哪个字段进行返回结果排序,支持以下字段：AddTime, ModTime 示例值：ModTime

3. 输出参数

参数名称	类型	描述
FunctionVersion	Array of String	函数版本。 示例值：["\$LATEST","3","2","1"]
Versions	Array of FunctionVersion	函数版本列表。
TotalCount	Integer	函数版本总数。 示例值：1
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 查询函数版本

输入示例

```
https://scf.tencentcloudapi.com/?Action=ListVersionByFunction
&FunctionName=functionName1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Versions": [
      {
        "Version": "$LATEST",
        "Description": "",
        "AddTime": "2024-12-18 10:11:51",
        "ModTime": "2024-12-18 17:04:41",
        "Status": "Active"
      }
    ],
    "FunctionVersion": [
      "$LATEST"
    ],
    "TotalCount": 1,
    "RequestId": "sdssd-a36d-42d3-9c24-c276409592a7"
  }
}
```

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
InvalidParameterValue	参数取值错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Order	Order传入错误。
InvalidParameterValue.Orderby	Orderby传入错误。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

发布新版本

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口用于用户发布新版本函数。

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：PublishVersion。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	发布函数的名称 示例值：<FunctionName>
Description	否	String	函数的描述 示例值：<Description>
Namespace	否	String	函数的命名空间 示例值：default

3. 输出参数

参数名称	类型	描述
FunctionVersion	String	函数的版本 示例值：4
CodeSize	Integer	代码大小 示例值：45686
MemorySize	Integer	最大可用内存 示例值：256
Description	String	函数的描述 示例值：dummytut
Handler	String	函数的入口 示例值：scfredis.main_handler
Timeout	Integer	函数的超时时间 示例值：3
Runtime	String	函数的运行环境 注意：此字段可能返回 null，表示取不到有效值。 示例值：Python2.7
Namespace	String	函数的命名空间 示例值：default

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 发布新版本

发布新版本

输入示例

```
https://scf.tencentcloudapi.com/?Action=PublishVersion
&FunctionName=<FunctionName>
&Description=<Description>
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Namespace": "default",
    "Description": "dummytut",
    "CodeSize": 45686,
    "MemorySize": 256,
    "Handler": "scfredis.main_handler",
    "RequestId": "68c9669c-9e2c-47b0-9a01-e8406a41be59",
    "Timeout": 3,
    "Runtime": "Python2.7",
    "FunctionVersion": "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
FailedOperation.PublishVersion	当前函数状态无法发布版本，请在状态为正常时发布。
InternalServerError	内部错误。
InternalServerError.System	内部系统错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
LimitExceeded.ContainerImageAccelerate	用户开启镜像加速函数版本超限。
LimitExceeded.ContainerImageAccelerateQuota	用户开启镜像加速函数版本超限。
ResourceInUse	资源被占用。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnsupportedOperation	操作不支持。
UnsupportedOperation.PublishVersion	当前状态无法发布版本。

设置预置并发

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

设置函数某一非\$LATEST版本的预置并发。

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：PutProvisionedConcurrencyConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	需要设置预置并发的函数的名称 示例值：functionName1
Qualifier	是	String	函数的版本号，注：\$LATEST 版本不支持预置并发 示例值：1
VersionProvisionedConcurrencyNum	是	Integer	预置并发数量，注：所有版本的预置并发数总和存在上限限制，当前的上限是：函数最大并 示例值：10
Namespace	否	String	函数所属命名空间，默认为default 示例值：default
TriggerActions.N	否	Array of TriggerAction	定时预置任务 示例值：[{"TriggerName":"timer-1","ProvisionedType":"Default","TriggerProvisionedConcurrencyNum":1,"T42 15 20 12 * 2024"}]
ProvisionedType	否	String	预置类型， 静态预置：Default 动态追踪并发利用率指标预置：ConcurrencyUtilizationTracking 预置类型二选一，设置静态预置时可以设置VersionProvisionedConcurrencyNum。 动态利用率预置可以设置TrackingTarget，MinCapacity，MaxCapacity，保持向后VersionProvisionedConcurrencyNum设置为0。 示例值：ConcurrencyUtilizationTracking
TrackingTarget	否	Float	指标追踪的并发利用率。设置范围(0,1) 示例值：0
MinCapacity	否	Integer	缩容时的最小值, 最小值为1 示例值：0
MaxCapacity	否	Integer	扩容时的最大值 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 设置预置并发

输入示例

```
https://scf.tencentcloudapi.com/?Action=PutProvisionedConcurrencyConfig
&FunctionName=functionName1
&Qualifier=1
&VersionProvisionedConcurrencyNum=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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.DebugModeStatus	当前调试状态无法执行此操作。
FailedOperation.FunctionVersionStatusNotActive	当前函数版本状态无法进行此操作，请在版本状态为正常时重试。

错误码	描述
FailedOperation.InsufficientBalance	当前账号或命名空间处于欠费状态，请在可用时重试。
FailedOperation.ProvisionCreateTimer	创建定时预置任务失败。
FailedOperation.ProvisionDeleteTimer	删除定时预置任务失败。
FailedOperation.ProvisionedExceedAvailable	预置超过可用。
FailedOperation.ProvisionedExceedReserved	预置超限。
FailedOperation.ProvisionedInProgress	当前函数版本已有预置任务处于进行中，请稍后重试。
FailedOperation.ServiceClosed	ServiceClosed 请确认后再操作。
FailedOperation.UnOpenedService	指定的服务未开通，可以提交工单申请开通服务。
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.MinCapacity	MinCapacity 参数传入错误。
InvalidParameterValue.ProvisionTriggerCronConfigDuplicate	定时预置的cron配置重复。
InvalidParameterValue.ProvisionTriggerName	TriggerName参数传入错误。
InvalidParameterValue.ProvisionTriggerNameDuplicate	TriggerName重复。
InvalidParameterValue.ProvisionType	ProvisionType 参数传入错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
InvalidParameterValue.TrackingTarget	TrackingTarget 参数输入错误。
InvalidParameterValue.TriggerCronConfig	TriggerCronConfig参数传入错误。
InvalidParameterValue.TriggerCronConfigTimeInterval	TriggerCronConfig参数定时触发间隔小于指定值。
InvalidParameterValue.TriggerProvisionedConcurrencyNum	TriggerProvisionedConcurrencyNum参数传入错误。
LimitExceeded.FunctionProvisionedConcurrencyMemory	FunctionProvisionedConcurrencyMemory数量达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
LimitExceeded.FunctionTotalProvisionedConcurrencyMemory	FunctionTotalProvisionedConcurrencyMemory达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
LimitExceeded.FunctionTotalProvisionedConcurrencyNum	函数预置并发总数达到限制。
LimitExceeded.MaxCapacity	动态扩容最大值超限。
LimitExceeded.ProvisionTriggerAction	定时预置数量超过最大限制。
LimitExceeded.ProvisionTriggerInterval	定时触发间隔小于最大限制。
MissingParameter	缺少参数错误。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation	未授权操作。

设置最大独占配额

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

设置函数最大独占配额

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：PutReservedConcurrencyConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	需要设置最大独占配额的函数的名称 示例值：functionName1
ReservedConcurrencyMem	是	Integer	函数最大独占配额，注：函数的最大独占配额内存总和上限：用户总并发内存配额 - 12800 示例值：12800
Namespace	否	String	函数所属命名空间，默认为default 示例值：default

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 设置最大独占配额

输入示例

```
https://scf.tencentcloudapi.com/?Action=PutReservedConcurrencyConfig
&FunctionName=functionName1
&ReservedConcurrencyMem=12800
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```



```
}
}
```

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。
FailedOperation.DebugModeStatus	当前调试状态无法执行此操作。
FailedOperation.FunctionVersionStatusNotActive	当前函数版本状态无法进行此操作，请在版本状态为正常时重试。
FailedOperation.InsufficientBalance	当前账号或命名空间处于欠费状态，请在可用时重试。
FailedOperation.ProvisionedExceedReserved	预置超限。
FailedOperation.ReservedInProgress	当前函数已有保留并发设置任务处于进行中，请稍后重试。
FailedOperation.ServiceClosed	ServiceClosed 请确认后再操作。
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
LimitExceeded.FunctionReservedConcurrencyMemory	函数保留并发内存超限。
MissingParameter	缺少参数错误。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation	未授权操作。

修改账号并发限制配额

最近更新时间：2025-04-25 01:51:56

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

修改账号并发限制配额

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：PutTotalConcurrencyConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TotalConcurrencyMem	是	Integer	账号并发内存配额，注：账号并发内存配额下限：用户已用并发内存总额 + 12800 示例值：128000
Namespace	否	String	命名空间，默认为default 示例值：default

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 设置用户并发内存配额

输入示例

```
https://scf.tencentcloudapi.com/?Action=PutTotalConcurrencyConfig
&TotalConcurrencyMem=128000
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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.InsufficientBalance	当前账号或命名空间处于欠费状态，请在可用时重试。
FailedOperation.ReservedExceedTotal	ReservedExceedTotal 总保留超限。
FailedOperation.TotalConcurrencyMemoryInProgress	用户并发内存配额设置任务处于进行中，请稍后重试。
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
LimitExceeded.TotalConcurrencyMemory	用户并发内存配额超限。
LimitExceeded.UserTotalConcurrencyMemory	UserTotalConcurrencyMemory达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
MissingParameter	缺少参数错误。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.TotalConcurrencyMemory	并发内存配额资源未找到。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

更新别名的配置

最近更新时间：2025-04-25 01:51:55

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

更新别名的配置

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：UpdateAlias。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：TestFunction
Name	是	String	别名的名称 示例值：TestAlias
FunctionVersion	是	String	别名指向的主版本 示例值：\$LATEST
Namespace	否	String	函数所在的命名空间 示例值：default
RoutingConfig	否	RoutingConfig	别名的路由信息，需要为别名指定附加版本时，必须提供此参数；附加版本指的是：除主版本 FunctionVersion 外，为此别名再指定一个函数可正常使用的版本；这里附加版本中的 Version 值 不能是别名指向的主版本；要注意的是：如果想要某个版本的流量全部指向这个别名，不需配置此参数；目前一个别名最多只能指定一个附加版本
Description	否	String	别名的描述 示例值：a alias desc

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 更新别名的配置

输入示例

```
https://scf.tencentcloudapi.com/?Action=UpdateAlias
&Name=abcAlias
&Namespace=default
```

```
&FunctionName=functionName1
&FunctionVersion=1
&RoutingConfig.AdditionalVersionWeights.0.Version=2
&RoutingConfig.AdditionalVersionWeights.0.Weight=0.5
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.UpdateAlias	更新别名失败。
InternalServerError.System	内部系统错误。
InvalidParameter.RoutingConfig	RoutingConfig参数传入错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AdditionalVersionWeights	AdditionalVersionWeights参数传入错误。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Name	Name参数传入错误。

错误码	描述
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.RoutingConfig	RoutingConfig取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#RoutingConfig 。
ResourceInUse	资源被占用。
ResourceNotFound.Alias	别名不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

更新函数代码

最近更新时间：2025-04-25 01:51:55

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入参数更新函数代码。

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：UpdateFunctionCode。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	要修改的函数名称 示例值：functionName1
Handler	否	String	函数处理方法名称。名称格式支持“文件名称.函数名称”形式（java 名称格式 包名.类名::方法名），文件名称和函数名称之间以"."隔开，文件名称和函数名称要求以字母开始和结尾，中间允许插入字母、数字、下划线和连接符，文件名称和函数名字的长度要求 2-60 个字符 示例值：index.main
CosBucketName	否	String	对象存储桶名称 示例值：<CosBucketName>
CosObjectName	否	String	对象存储对象路径 示例值：<CosObjectName>
ZipFile	否	String	包含函数代码文件及其依赖项的 zip 格式文件，使用该接口时要求将 zip 文件的内容转成 base64 编码，最大支持20M 示例值：<base64>
Namespace	否	String	函数所属命名空间 示例值：default
CosBucketRegion	否	String	对象存储的地域，注：北京分为ap-beijing和ap-beijing-1 示例值：ap-beijing
InstallDependency	否	String	是否自动安装依赖 示例值：FALSE
EnvId	否	String	函数所属环境 示例值：<函数所属环境>
Publish	否	String	在更新时是否同步发布新版本，默认为：FALSE，不发布 示例值：FALSE
Code	否	Code	函数代码
CodeSource	否	String	代码来源方式，支持 ZipFile, Cos, Inline 之一 示例值：Cos

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 更新函数代码

输入示例

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

{
  "CosObjectName": "<CosObjectName>",
  "Handler": "index.main",
  "CosBucketName": "<CosBucketName>",
  "FunctionName": "functionName1"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.CallRoleFailed	请求role信息失败。
FailedOperation.FunctionStatusError	函数在部署中,无法做此操作。
FailedOperation.OperationConflict	操作冲突。
FailedOperation.UpdateFunctionCode	当前函数状态无法更新代码，请在状态为正常时更新。
InternalServerError.System	内部系统错误。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Args	Args 参数值有误。
InvalidParameterValue.ArgsList	容器启动参数取值与规范不符。
InvalidParameterValue.Code	Code传入错误。
InvalidParameterValue.CodeSecret	CodeSecret传入错误。
InvalidParameterValue.CodeSource	CodeSource传入错误。
InvalidParameterValue.Command	Command[Entrypoint] 参数值有误。
InvalidParameterValue.CommandList	容器启动命令参数值与规范不符。
InvalidParameterValue.Cos	Cos传入错误。
InvalidParameterValue.CosBucketName	CosBucketName不符合规范。
InvalidParameterValue.CosBucketRegion	CosBucketRegion取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#Code 。
InvalidParameterValue.CosObjectName	CosObjectName不符合规范。
InvalidParameterValue.Demo	DemoID 对应的函数模板，code 参数值有误，请确认后重试。
InvalidParameterValue.Demold	Demold 不存在。
InvalidParameterValue.DnsInfo	环境变量DNS[OS_NAMESERVER]配置有误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.GitBranch	GitBranch不符合规范。
InvalidParameterValue.GitDirectory	GitDirectory不符合规范。
InvalidParameterValue.GitPassword	GitPassword不符合规范。
InvalidParameterValue.GitUrl	GitUrl不符合规范。
InvalidParameterValue.GitUserName	GitUserName不符合规范。
InvalidParameterValue.Handler	Handler传入错误。
InvalidParameterValue.ImageType	ImageType 参数值有误。
InvalidParameterValue.ImageUri	imageUri 传入有误。
InvalidParameterValue.InlineZipFile	InlineZipFile非法。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.RegistryId	企业版镜像实例ID[RegistryId]传值错误。
InvalidParameterValue.TempCosObjectName	非法的TempCosObjectName。

错误码	描述
InvalidParameterValue.ZipFile	检测到不是标准的zip文件，请重新压缩后再试。
InvalidParameterValue.ZipFileBase64BinasciiError	压缩文件base64解码失败: <code>Incorrect padding</code> ，请修正后再试。
LimitExceeded.ContainerImageAccelerate	用户开启镜像加速函数版本超限。
LimitExceeded.ContainerImageAccelerateQuota	用户开启镜像加速函数版本超限。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.ImageConfig	未找到指定的ImageConfig，请创建后再试。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.TempCosAppid	TempCos的Appid和请求账户的APPID不一致。
UnauthorizedOperation.UpdateFunctionCode	无法进行此操作。
UnsupportedOperation	操作不支持。
UnsupportedOperation.UpdateFunctionCode	更新函数代码失败。

更新函数异步重试配置

最近更新时间：2025-04-25 01:51:55

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

更新函数的异步重试配置，包括重试次数和消息保留时间

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：UpdateFunctionEventInvokeConfig。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AsyncTriggerConfig	是	AsyncTriggerConfig	异步重试配置信息
FunctionName	是	String	函数名称 示例值：FunctionName
Namespace	否	String	函数所属命名空间，默认为default 示例值：default

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 异步重试配置请求示例

输入示例

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

{
  "FunctionName": "FunctionName",
  "Namespace": "default",
  "AsyncTriggerConfig": {
    "MsgTTL": 3600,
    "RetryConfig": [
      {
```

```
"RetryNum": 1
}
]
}
}
```

输出示例

```
{
  "Response": {
    "RequestId": "XXXXXX"
  }
}
```

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。
FailedOperation.FunctionVersionStatusNotActive	当前函数版本状态无法进行此操作，请在版本状态为正常时重试。
FailedOperation.InsufficientBalance	当前账号或命名空间处于欠费状态，请在可用时重试。
InternalServerError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AsyncTriggerConfig	函数异步重试配置参数无效。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
LimitExceeded.MsgTTL	函数异步重试配置消息保留时间超过限制。

错误码	描述
LimitExceeded.RetryNum	函数异步重试配置异步重试次数超过限制。
MissingParameter	缺少参数错误。
ResourceNotFound	资源不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation.EipFixed	指定的配置EipFixed暂不支持。
UnsupportedOperation.UpdateFunctionEventInvokeConfig	指定的配置暂不支持，请修正后再试。
UnsupportedOperation.VpcConfig	指定的配置VpcConfig暂不支持。

创建函数

最近更新时间：2025-05-23 01:27:23

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入参数创建新的函数。

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：CreateFunction。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	创建的函数名称，函数名称支持26个英文字母大小写、数字、连接符和下划线，第一个字符只能以字母开头，最后一个字符不能为连接符或者下划线，名称长度 2-60 示例值：helloworld
Code	是	Code	函数代码. 注意：不能同时指定Cos、ZipFile或 Demold。
Handler	否	String	函数处理方法名称，名称格式支持 "文件名称.方法名称" 形式（java 名称格式 包名.类名::方法名），文件名称和函数名称之间以"."隔开，文件名称和函数名称要求以字母开始和结尾，中间允许插入字母、数字、下划线和连接符，文件名称和函数名字的长度要求是 2-60 个字符 示例值：index.main_handler
Description	否	String	函数描述,最大支持 1000 个英文字母、数字、空格、逗号、换行符和英文句号，支持中文 示例值：hello world!
MemorySize	否	Integer	函数运行时内存大小，默认为 128M，可选范围 64、128MB-3072MB，并且以 128MB 为阶梯 示例值：128
Timeout	否	Integer	函数最长执行时间，单位为秒，可选值范围 1-900 秒，默认为 3 秒 示例值：3
Environment	否	Environment	函数的环境变量
Runtime	否	String	函数运行环境，默认Python2.7 目前支持的运行环境： - Python2.7 - Python3.6 - Python3.7 - Python3.9 - Python3.10 - Nodejs6.10 - Nodejs8.9 - Nodejs10.15 - Nodejs12.16 - Nodejs14.18 - Nodejs16.13 - Nodejs18.15

参数名称	必选	类型	描述
			- Php5.6 - Php7(7.2版本) - Php7.4 - Php8.0 - Go1 - Java8 - CustomRuntime 示例值：Python3.6
VpcConfig	否	VpcConfig	函数的私有网络配置
Namespace	否	String	函数所属命名空间 示例值：default
Role	否	String	函数绑定的角色 示例值：SCF_ExecuteRole
InstallDependency	否	String	在线依赖安装 ，TRUE 表示安装，默认值为 FALSE。仅支持 Node.js 函数。 示例值：FALSE
ClsLogsetId	否	String	函数日志投递到的CLS LogsetId 示例值：abcd-xcxcf-xcxvv
ClsTopicId	否	String	函数日志投递到的CLS TopicId 示例值：cvfd-dfsf-xcvcx
Type	否	String	函数类型，默认值为Event，创建触发器函数请填写Event，创建HTTP函数级服务请填写HTTP 示例值：Event
CodeSource	否	String	CodeSource 代码来源，支持ZipFile, Cos, Demo 其中之一 示例值：ZipFile
Layers.N	否	Array of LayerVersionSimple	函数要关联的Layer版本列表，Layer会按照在列表中顺序依次覆盖。
DeadLetterConfig	否	DeadLetterConfig	死信队列参数
PublicNetConfig	否	PublicNetConfigIn	公网访问配置
CfsConfig	否	CfsConfig	文件系统配置参数，用于云函数挂载文件系统
InitTimeout	否	Integer	函数初始化超时时间，默认 65s，镜像部署函数默认 90s。 示例值：3
Tags.N	否	Array of Tag	函数 Tag 参数，以键值对数组形式传入
AsyncRunEnable	否	String	是否开启异步属性，TRUE 为开启，FALSE为关闭 示例值：FALSE
TraceEnable	否	String	是否开启事件追踪，TRUE 为开启，FALSE为关闭 示例值：FALSE
AutoDeployClsTopicIndex	否	String	是否自动创建cls索引，TRUE 为开启，FALSE为关闭 示例值：FALSE
AutoCreateClsTopic	否	String	是否自动创建cls主题，TRUE 为开启，FALSE为关闭 示例值：FALSE
ProtocolType	否	String	HTTP函数支持的访问协议。当前支持WebSockets协议，值为WS 示例值：WS
ProtocolParams	否	ProtocolParams	HTTP函数配置ProtocolType访问协议，当前协议可配置的参数
InstanceConcurrencyConfig	否	InstanceConcurrencyConfig	单实例多并发配置。只支持Web函数。
DnsCache	否	String	是否开启Dns缓存能力。只支持EVENT函数。默认为FALSE，TRUE 为开启，FALSE为关闭 示例值：FALSE
IntranetConfig	否	IntranetConfigIn	内网访问配置

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 创建函数

创建函数

输入示例

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

{
  "Code": {
    "CosBucketName": "<CosBucketName>",
    "CosObjectName": "<CosObjectName>"
  },
  "Handler": "<function.handler>",
  "FunctionName": "<FunctionName>"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.AccountInsufficient	账户余额不足。
FailedOperation.ApmConfigInstancelId	获取Apm InstancelId失败。
FailedOperation.BindPlugin	绑定插件失败，请检查参数。
FailedOperation.CallNetDeployFailed	调用 NetDeploy 失败。
FailedOperation.CallRoleFailed	请求role信息失败。
FailedOperation.ClsServiceUnregistered	CLS服务未注册。
FailedOperation.ClusterNotFound	函数集群不存在。
FailedOperation.CreateFunction	操作失败。
FailedOperation.InstanceNotFound	InstanceNotFound 实例不存在。
FailedOperation.InsufficientResources	本地域资源售罄。
FailedOperation.Namespace	命名空间已存在，请勿重复创建。
FailedOperation.OpenService	服务开通失败。
FailedOperation.QcsRoleNotFound	角色不存在。
FailedOperation.TotalConcurrencyMemoryInProgress	用户并发内存配额设置任务处于进行中，请稍后重试。
FailedOperation.UnOpenedService	指定的服务未开通，可以提交工单申请开通服务。
InternalError	内部错误。
InternalError.GetRoleError	内部服务错误。
InternalError.GetStsTokenFailed	获取sts票据信息失败。
InternalError.System	内部系统错误。
InternalError.SystemError	内部服务错误。
InvalidParameter.ParamError	创建函数传参异常。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Action	Action取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17235 。
InvalidParameterValue.ApmConfig	ApmConfig参数传入错误。
InvalidParameterValue.ApmConfigInstancelId	ApmConfigInstancelId参数传入错误。
InvalidParameterValue.ApmConfigRegion	ApmConfigRegion参数传入错误。
InvalidParameterValue.Args	Args 参数值有误。
InvalidParameterValue.ArgsList	容器启动参数取值与规范不符。

错误码	描述
InvalidParameterValue.AsyncRunEnable	AsyncRunEnable 取值不正确。
InvalidParameterValue.CfsParameterDuplicate	cfs配置项重复。
InvalidParameterValue.CfsParameterError	cfs配置项取值与规范不符。
InvalidParameterValue.CfsStructionError	cfs参数格式与规范不符。
InvalidParameterValue.Cls	Cls传入错误。
InvalidParameterValue.Code	Code传入错误。
InvalidParameterValue.CodeSecret	CodeSecret传入错误。
InvalidParameterValue.CodeSource	CodeSource传入错误。
InvalidParameterValue.Command	Command[Entrypoint] 参数值有误。
InvalidParameterValue.CommandList	容器启动命令参数值与规范不符。
InvalidParameterValue.Cos	Cos传入错误。
InvalidParameterValue.CosBucketName	CosBucketName不符合规范。
InvalidParameterValue.CosBucketRegion	CosBucketRegion取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#Code 。
InvalidParameterValue.CosObjectName	CosObjectName不符合规范。
InvalidParameterValue.DeadLetterConfig	DeadLetterConfig取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#DeadLetterConfig 。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.DiskSize	DiskSize 参数取值不规范。
InvalidParameterValue.DnsInfo	环境变量DNS[OS_NAMESERVER]配置有误。
InvalidParameterValue.DynamicEnabled	DynamicEnabled 参数传入错误。
InvalidParameterValue.EipConfig	EipConfig参数错误。
InvalidParameterValue.Environment	Environment传入错误。
InvalidParameterValue.EnvironmentExceededLimit	环境变量大小超限，请保持在 4KB 以内。
InvalidParameterValue.EnvironmentSystemProtect	不支持修改函数系统环境变量和运行环境变量。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.FunctionType	FunctionType参数错误。
InvalidParameterValue.GitCommitId	GitCommitId取值与规范不符，请修正后再试。
InvalidParameterValue.GitPasswordSecret	GitPasswordSecret 传参有误。
InvalidParameterValue.GitUrl	GitUrl不符合规范。
InvalidParameterValue.GitUserNameSecret	GitUserNameSecret 传参有误。
InvalidParameterValue.Handler	Handler传入错误。
InvalidParameterValue.IdleTimeOut	IdleTimeOut参数传入错误。
InvalidParameterValue.ImageType	ImageType 参数值有误。
InvalidParameterValue.ImageUri	imageUri 传入有误。
InvalidParameterValue.InstanceConcurrencyConfig	InstanceConcurrencyConfig 参数传入错误。
InvalidParameterValue.L5Enable	L5Enable取值与规范不符，请修正后再试。

错误码	描述
InvalidParameterValue.Layers	Layers参数传入错误。
InvalidParameterValue.LimitExceeded	参数超出长度限制。
InvalidParameterValue.LogFormat	日志格式不正确。
InvalidParameterValue.MaxConcurrency	MaxConcurrency 参数传入错误。
InvalidParameterValue.Memory	Memory 取值与规范不符，函数运行时内存大小默认为128MB，可选范围64MB、128MB – 3072MB，以128MB为阶梯，请修正后重试。
InvalidParameterValue.MemorySize	MemorySize错误。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.NodeSpec	NodeSpec 参数传入错误。
InvalidParameterValue.NodeType	NodeType 参数传入错误。
InvalidParameterValue.ProtocolType	ProtocolType参数传入错误。
InvalidParameterValue.PublicNetConfig	PublicNetConfig参数错误。
InvalidParameterValue.RegistryId	企业版镜像实例ID[RegistryId]传值错误。
InvalidParameterValue.Runtime	Runtime传入错误。
InvalidParameterValue.Stamp	Stamp取值与规范不符，请修正后再试。
InvalidParameterValue.TempCosObjectName	非法的TempCosObjectName。
InvalidParameterValue.TraceEnable	TraceEnable取值与规范不符，请修正后再试。
InvalidParameterValue.Type	Type传入错误。
InvalidParameterValue.Vpc	VPC不正确。
InvalidParameterValue.VpcNotSetWhenOpenCfs	开启cfs配置的同时必须开启vpc。
InvalidParameterValue.WebSocketsParams	WebSocketsParams参数传入错误。
InvalidParameterValue.ZipFile	检测到不是标准的zip文件，请重新压缩后再试。
InvalidParameterValue.ZipFileBase64BinasciiError	压缩文件base64解码失败: <code>Incorrect padding</code> ，请修正后再试。
LimitExceeded.ContainerImageAccelerate	用户开启镜像加速函数版本超限。
LimitExceeded.ContainerImageAccelerateQuota	用户开启镜像加速函数版本超限。
LimitExceeded.Eip	eip资源超限。
LimitExceeded.Function	函数数量超出最大限制，可通过 提交工单 申请提升限制。
LimitExceeded.InitTimeout	InitTimeout达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
LimitExceeded.Intralp	内网固定IP个数超限。
LimitExceeded.Memory	内存超出最大限制。
LimitExceeded.PrivilegeContainer	特权容器属性开启失败
LimitExceeded.Timeout	Timeout超出最大限制。
MissingParameter.Code	Code没有传入。
MissingParameter.Runtime	缺失 Runtime 字段。
ResourceInUse	资源被占用。
ResourceInUse.Function	函数已存在。
ResourceInUse.FunctionName	FunctionName已存在。

错误码	描述
ResourceNotFound	资源不存在。
ResourceNotFound.CfsMountInsNotMatch	指定的cfs下未找到您所指定的挂载点。
ResourceNotFound.CfsProtocolError	CfsProtocolError 参数异常。
ResourceNotFound.CfsStatusError	检测cfs状态为不可用。
ResourceNotFound.CfsVpcNotMatch	cfs与云函数所处vpc不一致。
ResourceNotFound.Cmq	Cmq不存在。
ResourceNotFound.Demo	不存在的Demo。
ResourceNotFound.GetCfsMountInsError	获取cfs挂载点信息错误。
ResourceNotFound.GetCfsNotMatch	获取cfs信息错误。
ResourceNotFound.ImageConfig	未找到指定的ImageConfig，请创建后再试。
ResourceNotFound.Layer	layer不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Role	角色不存在。
ResourceNotFound.Vpc	VPC或子网不存在。
ResourceUnavailable.Namespace	Namespace不可用。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.Region	Region错误。
UnauthorizedOperation.Role	没有权限访问您的Cos资源。
UnauthorizedOperation.TempCosAppid	TempCos的Appid和请求账户的APPID不一致。
UnsupportedOperation	操作不支持。
UnsupportedOperation.AsyncRunEnable	指定的配置AsyncRunEnable暂不支持，请修正后再试。
UnsupportedOperation.DiskSize	DiskSize 配置内测中，需要申请开启。如需使用可联系我们。
UnsupportedOperation.EipFixed	指定的配置EipFixed暂不支持。
UnsupportedOperation.VpcConfig	指定的配置VpcConfig暂不支持。

获取函数详细信息

最近更新时间：2025-05-23 01:27:23

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口获取某个函数的详细信息，包括名称、代码、处理方法、关联触发器和超时时间等字段。

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：GetFunction。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	否	String	需要获取详情的函数名称，ResourceId和FunctionName只能传一个 示例值：<FunctionName>
Qualifier	否	String	函数的版本号 默认值：\$LATEST 示例值：\$LATEST
Namespace	否	String	函数所属命名空间 默认值：default 示例值：default
ShowCode	否	String	是否显示代码，TRUE表示显示代码，FALSE表示不显示代码,大于1M的入口文件不会显示 示例值：TRUE

3. 输出参数

参数名称	类型	描述
ModTime	Timestamp	函数的最后修改时间 示例值：2018-06-07 09:52:23
CodeInfo	String	函数的代码 示例值："helloworld"
Description	String	函数的描述信息 示例值：这是一个云函数
Triggers	Array of Trigger	函数的触发器列表
Handler	String	函数的入口 示例值：scfredis.main_handler
CodeSize	Integer	函数代码大小 示例值：0
Timeout	Integer	函数的超时时间 示例值：3

参数名称	类型	描述
FunctionVersion	String	函数的版本 示例值：\$LATEST
MemorySize	Integer	函数的最大可用内存 示例值：128
Runtime	String	函数的运行环境 注意：此字段可能返回 null，表示取不到有效值。 示例值：Python2.7
FunctionName	String	函数的名称 示例值：ledDummyAPITest
VpcConfig	VpcConfig	函数的私有网络
UseGpu	String	是否使用GPU 示例值：FALSE
Environment	Environment	函数的环境变量
CodeResult	String	代码是否正确 示例值：failed
CodeError	String	代码错误信息 示例值："
ErrNo	Integer	代码错误码 示例值：0
Namespace	String	函数的命名空间 示例值：default
Role	String	函数绑定的角色 示例值：scf-role1
InstallDependency	String	是否自动安装依赖 示例值：FALSE
Status	String	函数状态，状态值及流转 参考说明 示例值：Active
StatusDesc	String	状态描述 示例值："
ClsLogsetId	String	日志投递到的Cls日志集 示例值：5354061a-dcaa-46be-8359-fe5gt56075c0
ClsTopicId	String	日志投递到的Cls Topic 示例值：5354061a-dcaa-46be-jydw-fe51e56075c0
FunctionId	String	函数ID 示例值：lam-bcrxav2w
Tags	Array of Tag	函数的标签列表
EipConfig	EipOutConfig	EipConfig配置
AccessInfo	AccessInfo	域名信息
Type	String	函数类型，取值为HTTP或者Event 示例值：HTTP
L5Enable	String	是否启用L5 示例值：FALSE
Layers	Array of LayerVersionInfo	函数关联的Layer版本信息
DeadLetterConfig	DeadLetterConfig	函数关联的死信队列信息

参数名称	类型	描述
AddTime	Timestamp	函数创建回见 示例值：2024-12-20 10:00:00
PublicNetConfig	PublicNetConfigOut	公网访问配置 注意：此字段可能返回 null，表示取不到有效值。
OnsEnable	String	是否启用Ons 注意：此字段可能返回 null，表示取不到有效值。 示例值：FALSE
CfsConfig	CfsConfig	文件系统配置参数，用于云函数挂载文件系统
AvailableStatus	String	函数的计费状态，状态值 参考此处 示例值：Available
Qualifier	String	函数版本 示例值：\$LATEST
InitTimeout	Integer	函数初始化超时时间 示例值：3
StatusReasons	Array of StatusReason	函数状态失败原因 示例值：[]
AsyncRunEnable	String	是否开启异步属性 示例值：FALSE
TraceEnable	String	是否开启事件追踪 示例值：FALSE
ImageConfig	ImageConfig	镜像配置 注意：此字段可能返回 null，表示取不到有效值。
ProtocolType	String	HTTP函数支持的访问协议。当前支持WebSockets协议。 注意：此字段可能返回 null，表示取不到有效值。 示例值："
ProtocolParams	ProtocolParams	HTTP函数配置ProtocolType访问协议，当前协议配置的参数 注意：此字段可能返回 null，表示取不到有效值。
DnsCache	String	是否开启DNS缓存 示例值：FALSE
IntranetConfig	IntranetConfigOut	内网访问配置 注意：此字段可能返回 null，表示取不到有效值。
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 获取函数详细信息

用户用此函数获取对应的函数信息，可指定版本与命名空间。

输入示例

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

{
  "FunctionName": "<FunctionName>"
}
```


输出示例

```
{
  "Response": {
    "Qualifier": "$LATEST",
    "Description": "Created by Serverless",
    "Timeout": 3,
    "InitTimeout": 90,
    "MemorySize": 128,
    "Runtime": "CustomImage",
    "VpcConfig": {
      "VpcId": "",
      "SubnetId": ""
    },
    "Environment": {
      "Variables": []
    },
    "Handler": "",
    "UseGpu": "FALSE",
    "Role": "",
    "CodeSize": 0,
    "FunctionVersion": "$LATEST",
    "FunctionName": "function-index1",
    "FunctionId": "xccvxcv",
    "Namespace": "default",
    "InstallDependency": "FALSE",
    "Status": "Active",
    "AvailableStatus": "Available",
    "StatusDesc": "",
    "L5Enable": "FALSE",
    "DnsCache": "FALSE",
    "EipConfig": {
      "EipFixed": "FALSE",
      "Eips": []
    },
    "ModTime": "2024-12-17 17:52:48",
    "AddTime": "2024-12-17 17:47:08",
    "Layers": [],
    "DeadLetterConfig": {
      "Type": "",
      "Name": "",
      "FilterType": ""
    },
    "OnsEnable": "FALSE",
    "PublicNetConfig": {
      "PublicNetStatus": "ENABLE",
      "EipConfig": {
        "EipStatus": "DISABLE",
        "EipAddress": []
      }
    },
    "ImageConfig": {
      "RegistryId": "",
      "ImageType": "personal",
      "ImageUri": "ccr.ccs.tencentyun.com/xxx/xxx:xx@sha256:xxxxxxxxxx",
      "ContainerImageAccelerate": false,
      "Command": "node",
```

```
"Args": " index.js ",
"ImagePort": 9000,
"EntryPoint": null
},
"ProtocolType": "",
"Triggers": [],
"ClsLogsetId": "b91e2c12-cccccccccccc",
"ClsTopicId": "a3b039f5-bbbbbbbbbb",
"CodeInfo": "",
"CodeResult": "success",
"CodeError": "",
"ErrNo": 0,
"Tags": [],
"AccessInfo": {
  "Host": "",
  "Vip": ""
},
"Type": "Event",
"CfsConfig": {
  "CfsInsList": []
},
"StatusReasons": [],
"AsyncRunEnable": "FALSE",
"TraceEnable": "FALSE",
"ProtocolParams": null,
"IntranetConfig": {
  "IpFixed": "DISABLE"
},
"RequestId": "08b4c93b-b515-4cf4-a7a3-72b5db4e21dc"
}
```

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.Apigw	创建触发器失败。
FailedOperation.CallNetDeployFailed	调用 NetDeploy 失败。
InternalError	内部错误。
InternalError.Exception	内部服务异常。
InternalError.System	内部系统错误。
InvalidParameter.FunctionNameOrResourceIdRequired	资源ID和函数名参数不能同时为空。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.CodeSecret	CodeSecret传入错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Version	版本不存在。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.CodeSecret	无访问代码权限。

获取函数运行日志

最近更新时间：2025-04-25 01:51:57

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据指定的日志查询条件返回函数运行日志。该接口已下线，查询函数请求运行的返回信息，请使用 [GetRequestStatus](#)。查询函数运行日志，请参考[日志检索教程](#)。

默认接口请求频率限制：5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：GetFunctionLogs。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	否	String	函数的名称。 - 为保证 获取函数运行日志 接口GetFunctionLogs兼容性，输入参数FunctionName仍为非必填项，但建议填写该参数，否则可能导致日志获取失败。 - 函数关联日志服务后，建议使用 日志服务 相关接口以获得最佳日志检索体验。 示例值：helloworld-demo
Offset	否	Integer	数据的偏移量，Offset+Limit不能大于10000 示例值：0
Limit	否	Integer	返回数据的长度，Offset+Limit不能大于10000 示例值：20
Order	否	String	以升序还是降序的方式对日志进行排序，可选值 desc和 asc 示例值：desc
OrderBy	否	String	根据某个字段排序日志,支持以下字段：function_name, duration, mem_usage, start_time 示例值：function_name
Filter	否	LogFilter	日志过滤条件。可用于区分正确和错误日志，filter.RetCode=not0 表示只返回错误日志，filter.RetCode=is0 表示只返回正确日志，不传，则返回所有日志 示例值：not0
Namespace	否	String	函数的命名空间 示例值：default
Qualifier	否	String	函数的版本 示例值：\$LATEST
FunctionRequestId	否	String	执行该函数对应的requestId 示例值：d1922633-55cf-443a-a6cc-0788ce93df06
StartTime	否	Timestamp	查询的具体日期，例如：2017-05-16 20:00:00，只能与endtime相差一天之内 示例值：2017-05-16 20:00:00
EndTime	否	Timestamp	查询的具体日期，例如：2017-05-16 20:59:59，只能与startTime相差一天之内 示例值：2017-05-16 20:59:59
SearchContext	否	LogSearchContext	该字段已下线

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	函数日志的总数 示例值：100
Data	Array of FunctionLog	函数日志信息
SearchContext	LogSearchContext	该字段已下线
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 获取函数运行日志

获取函数运行日志

输入示例

```
https://scf.tencentcloudapi.com/?Action=GetFunctionLogs
&FunctionName=<FunctionName>
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "Data": [
      {
        "MemUsage": 3174400,
        "RetCode": 1,
        "RetMsg": "Success",
        "Log": "",
        "BillDuration": 100,
        "InvokeFinished": 1,
        "RequestId": "bc309eaa-6d64-11e8-a7fe-5254000b4175",
        "StartTime": "2018-06-11 18:46:45",
        "Duration": 0.532,
        "FunctionName": "APITest",
        "Level": "",
        "Source": "",
        "RetryNum": 1
      }
    ],
    "SearchContext": {
      "Offset": "",
      "Limit": 0,
      "Keyword": "",
      "Type": ""
    },
    "RequestId": "e2571ff3-da04-4c53-8438-f58bf057ce4a"
  }
}
```

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。
FailedOperation.AuthFailure	登录信息验证失败，token 验证失败。
FailedOperation.QueryError	查询失败。
FailedOperation.TopicNotExist	Topic不存在。
InternalServerError	内部错误。
InternalServerError.ES	ES错误。
InternalServerError.Exception	内部服务异常。
InternalServerError.System	内部系统错误。
InvalidParameter.Cls	日志参数有误。
InvalidParameter.ParamError	创建函数传参异常。
InvalidParameter.Payload	请求参数不合法。
InvalidParameter.RoleCheck	RoleCheck 传参有误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.DateTime	DateTime传入错误。
InvalidParameterValue.FunctionType	FunctionType参数错误。
InvalidParameterValue.ImageType	ImageType 参数值有误。
InvalidParameterValue.Offset	偏移量不合法。
InvalidParameterValue.Order	Order传入错误。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。可参考对应接口文档中的示例值： https://cloud.tencent.com/document/product/583/17235

错误码	描述
InvalidParameterValue.RetCode	RetCode不合法。
InvalidParameterValue.StartTimeOrEndTime	需要同时指定开始日期与结束日期。
LimitExceeded.Offset	Offset超出限制。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.RoleCheck	Role不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnsupportedOperation	操作不支持。
UnsupportedOperation.NotSupportRegion	不支持的地域。

运行函数

最近更新时间：2025-04-25 01:51:57

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口用于运行函数。

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：Invoke。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：helloworld
InvocationType	否	String	同步调用请使用 同步 Invoke 调用接口 或填写同步调用参数 RequestResponse，建议使用同步调用接口以获取最佳性能；异步调用填写 Event；默认为同步。接口超时时间为 300s，更长超时时间请使用异步调用。 示例值：RequestResponse
Qualifier	否	String	触发函数的版本号或别名，默认值为 \$LATEST 示例值：\$LATEST
ClientContext	否	String	运行函数时的参数，以json格式传入，同步调用最大支持 6MB，异步调用最大支持 128 KB。该字段信息对应函数 event 入参 。 示例值：{"key":"value"}
LogType	否	String	异步调用该字段返回为空。 示例值：Tail
Namespace	否	String	命名空间 示例值：default
RoutingKey	否	String	函数灰度流量控制调用，以json格式传入，例如{"k":"v"}，注意kv都需要是字符串类型，最大支持的参数长度是1024字节 示例值：{"key":"value"}

3. 输出参数

参数名称	类型	描述
Result	Result	函数执行结果
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 运行函数

运行函数

输入示例

```
https://scf.tencentcloudapi.com/?Action=Invoke
&FunctionName=function-demo
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Result": {
      "MemUsage": 3207168,
      "Log": "",
      "RetMsg": "hello from scf",
      "BillDuration": 100,
      "FunctionRequestId": "6add56fa-58f1-11e8-89a9-5254005d5fdb",
      "Duration": 0.826,
      "ErrMsg": "",
      "InvokeResult": 0
    },
    "RequestId": "c2af8a64-c922-4d55-ae0-bd86a5c2cd12"
  }
}
```

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.FunctionStatusError	函数在部署中,无法做此操作。
FailedOperation.InvokeFunction	调用函数失败。
InternalServerError	内部错误。

错误码	描述
InternalServerError.System	内部系统错误。
InvalidParameter.FunctionName	FunctionName取值与规范不符，请修正后再试。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Param	入参不是标准的json。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceUnavailable.InsufficientBalance	余额不足，请先充值。
UnauthorizedOperation.CAM	CAM鉴权失败。

更新函数配置

最近更新时间：2025-05-23 01:27:23

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入参数更新函数配置。

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：UpdateFunctionConfiguration。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	要修改的函数名称 示例值：<FunctionName>
Description	否	String	函数描述。最大支持 1000 个英文字母、数字、空格、逗号和英文句号，支持中文 示例值：<Description>
MemorySize	否	Integer	函数运行时内存大小，默认为 128 M，可选范64M、128 M-3072 M，以 128MB 为阶梯。 示例值：128
Timeout	否	Integer	函数最长执行时间，单位为秒，可选值范 1-900 秒，默认为 3 秒 示例值：3
Runtime	否	String	函数运行环境，创建时指定，目前不支持修改。 示例值：Python3.6
Environment	否	Environment	函数的环境变量
Namespace	否	String	函数所属命名空间 默认值：default 示例值：default
VpcConfig	否	VpcConfig	函数的私有网络配置
Role	否	String	函数绑定的角色 示例值：SCF_ExecuteRole
InstallDependency	否	String	在线依赖安装 ，TRUE 表示安装，默认值为 FALSE。仅支持 Node.js 函数。 示例值：FALSE
ClsLogsetId	否	String	日志投递到的cls日志集ID 示例值：fsdfs-sfdsfsd-sdfsdf
ClsTopicId	否	String	日志投递到的cls Topic ID 示例值：fsdfs-sfdsfsd-sdfsdf
Publish	否	String	在更新时是否同步发布新版本，默认为：FALSE，不发布新版本 示例值：FALSE

参数名称	必选	类型	描述
L5Enable	否	String	是否开启L5访问能力，TRUE 为开启，FALSE为关闭 示例值：FALSE
Layers.N	否	Array of LayerVersionSimple	函数要关联的层版本列表，层的版本会按照在列表中顺序依次覆盖。
DeadLetterConfig	否	DeadLetterConfig	函数关联的死信队列信息
PublicNetConfig	否	PublicNetConfigIn	公网访问配置
CfsConfig	否	CfsConfig	文件系统配置入参，用于云函数绑定CFS文件系统
InitTimeout	否	Integer	函数初始化执行超时时间 示例值：15
ProtocolParams	否	ProtocolParams	HTTP函数配置ProtocolType访问协议，当前协议可配置的参数
InstanceConcurrencyConfig	否	InstanceConcurrencyConfig	单实例多并发配置。只支持Web函数。
DnsCache	否	String	是否开启Dns缓存能力。只支持EVENT函数。默认为FALSE，TRUE 为开启，FALSE为关闭 示例值：FALSE
IntranetConfig	否	IntranetConfigIn	内网访问配置
IgnoreSysLog	否	Boolean	忽略系统日志上报 示例值：false

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 更新函数配置

更新函数配置

输入示例

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

{
  "FunctionName": "functionName1",
  "Namespace": "default",
  "Timeout": 100,
  "VpcConfig": {
    "SubnetId": "subnet-xxxx",
    "VpcId": "vpc-xxxxx"
  },
  "MemorySize": 512
}
```

输出示例

```
{
  "Response": {
    "RequestId": "eac6b301-a322-493a-8e36-83b295459397"
  }
}
```

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。
FailedOperation.AccountInsufficient	账户余额不足。
FailedOperation.ApmConfigInstancelId	获取Apm InstancelId失败。
FailedOperation.BindPlugin	绑定插件失败，请检查参数。
FailedOperation.CallNetDeployFailed	调用 NetDeploy 失败。
FailedOperation.ClsServiceUnregistered	CLS服务未注册。
FailedOperation.DebugModeUpdateTimeOutFail	调试状态下无法更新执行超时时间。
FailedOperation.InstanceNotFound	InstanceNotFound 实例不存在。
FailedOperation.InsufficientBalance	当前账号或命名空间处于欠费状态，请在可用时重试。
FailedOperation.InsufficientResources	本地域资源售罄。
FailedOperation.OpenService	服务开通失败。
FailedOperation.QcsRoleNotFound	角色不存在。
FailedOperation.ReservedInProgress	当前函数已有保留并发设置任务处于进行中，请稍后重试。
FailedOperation.UpdateFunctionConfiguration	UpdateFunctionConfiguration操作失败。

错误码	描述
InternalError.System	内部系统错误。
InternalError.SystemError	内部服务错误。
InvalidParameter.ParamError	创建函数传参异常。
InvalidParameter.Payload	请求参数不合法。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Action	Action取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17235 。
InvalidParameterValue.ApmConfig	ApmConfig参数传入错误。
InvalidParameterValue.ApmConfigInstanceId	ApmConfigInstanceId参数传入错误。
InvalidParameterValue.ApmConfigRegion	ApmConfigRegion参数传入错误。
InvalidParameterValue.CfsParameterDuplicate	cfs配置项重复。
InvalidParameterValue.CfsParameterError	cfs配置项取值与规范不符。
InvalidParameterValue.Cls	Cls传入错误。
InvalidParameterValue.ClsRole	修改Cls配置需要传入Role参数，请修正后重试。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.DiskSize	DiskSize 参数取值不规范。
InvalidParameterValue.DnsInfo	环境变量DNS[OS_NAMESERVER]配置有误。
InvalidParameterValue.DynamicEnabled	DynamicEnabled 参数传入错误。
InvalidParameterValue.EipConfig	EipConfig参数错误。
InvalidParameterValue.Environment	Environment传入错误。
InvalidParameterValue.EnvironmentExceededLimit	环境变量大小超限，请保持在 4KB 以内。
InvalidParameterValue.EnvironmentSystemProtect	不支持修改函数系统环境变量和运行环境变量。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Handler	Handler传入错误。
InvalidParameterValue.IdleTimeOut	IdleTimeOut参数传入错误。
InvalidParameterValue.InstanceConcurrencyConfig	InstanceConcurrencyConfig 参数传入错误。
InvalidParameterValue.L5Enable	L5Enable取值与规范不符，请修正后再试。
InvalidParameterValue.Layers	Layers参数传入错误。
InvalidParameterValue.LimitExceeded	参数超出长度限制。
InvalidParameterValue.LogFormat	日志格式不正确。
InvalidParameterValue.MaxConcurrency	MaxConcurrency 参数传入错误。
InvalidParameterValue.Memory	Memory 取值与规范不符，函数运行时内存大小默认为128MB，可选范围64MB、128MB – 3072MB，以128MB为阶梯，请修正后重试。
InvalidParameterValue.MemorySize	MemorySize错误。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.NodeSpec	NodeSpec 参数传入错误。
InvalidParameterValue.NodeType	NodeType 参数传入错误。

错误码	描述
InvalidParameterValue.PublicNetConfig	PublicNetConfig参数错误。
InvalidParameterValue.Runtime	Runtime传入错误。
InvalidParameterValue.SystemEnvironment	系统环境变量错误。
InvalidParameterValue.TraceEnable	TraceEnable取值与规范不符，请修正后再试。
InvalidParameterValue.Vpc	VPC不正确。
InvalidParameterValue.WebSocketsParams	WebSocketsParams参数传入错误。
LimitExceeded.Eip	eip资源超限。
LimitExceeded.InitTimeout	InitTimeout达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
LimitExceeded.Intralp	内网固定IP个数超限。
LimitExceeded.Memory	内存超出最大限制。
LimitExceeded.PrivilegeContainer	特权容器属性开启失败
LimitExceeded.Timeout	Timeout超出最大限制。
ResourceNotFound.CfsMountInsNotMatch	指定的cfs下未找到您所指定的挂载点。
ResourceNotFound.CfsProtocolError	CfsProtocolError 参数异常。
ResourceNotFound.CfsStatusError	检测cfs状态为不可用。
ResourceNotFound.CfsVpcNotMatch	cfs与云函数所处vpc不一致。
ResourceNotFound.Cmq	Cmq不存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.GetCfsMountInsError	获取cfs挂载点信息错误。
ResourceNotFound.GetCfsNotMatch	获取cfs信息错误。
ResourceNotFound.Layer	layer不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Role	角色不存在。
ResourceNotFound.Vpc	VPC或子网不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.Role	没有权限访问您的Cos资源。
UnsupportedOperation	操作不支持。
UnsupportedOperation.DiskSize	DiskSize 配置内测中，需要申请开启。如需使用可联系我们。
UnsupportedOperation.EipFixed	指定的配置EipFixed暂不支持。
UnsupportedOperation.VpcConfig	指定的配置VpcConfig暂不支持。

复制函数

最近更新时间：2025-04-25 01:51:59

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

复制一个函数，您可以选择将复制出的新函数放置在特定的Region和Namespace。

注：本接口不会复制函数的以下对象或属性：

- 1. 函数的触发器
- 2. 除了\$LATEST以外的其它版本
- 3. 函数配置的日志投递到的CLS目标。

如有需要，您可以在复制后手动配置新函数。

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：CopyFunction。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	要复制的函数的名称 示例值：myfunc
NewFunctionName	是	String	新函数的名称 示例值：newfunc
Namespace	否	String	要复制的函数所在的命名空间，默认为default 示例值：default
TargetNamespace	否	String	将函数复制到的命名空间，默认为default 示例值：othersns
Description	否	String	新函数的描述 示例值：a new function
TargetRegion	否	String	要将函数复制到的地域，不填则默认为当前地域 示例值：ap-shanghai
Override	否	Boolean	如果目标Namespace下已有同名函数，是否覆盖，默认为否 (注意：如果选择覆盖，会导致同名函数被删除，请慎重操作) TRUE：覆盖同名函数 FALSE：不覆盖同名函数 示例值：FALSE
CopyConfiguration	否	Boolean	是否复制函数的属性，包括环境变量、内存、超时、函数描述、标签、VPC等，默认为否。 TRUE：复制函数配置 FALSE：不复制函数配置 示例值：TRUE

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 复制函数

复制函数

输入示例

```
https://scf.tencentcloudapi.com/?Action=CopyFunction
&Namespace=default
&FunctionName=myfunc
&TargetNamespace=othersns
&NewFunctionName=newfunc
&TargetRegion=ap-shanghai
&Override=FALSE
&CopyConfiguration=TRUE
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  }
}
```

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.CopyAsyncRun	CopyAsyncRun 传参异常。
FailedOperation.CopyFailed	复制函数失败。
FailedOperation.CopyFunction	不支持复制到该地域。
InternalServerError	内部错误。
InternalServerError.Exception	内部服务异常。
InternalServerError.System	内部系统错误。
InvalidParameterValue.Code	Code传入错误。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.Environment	Environment传入错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.Handler	Handler传入错误。
InvalidParameterValue.Runtime	Runtime传入错误。
LimitExceeded.Function	函数数量超出最大限制，可通过 提交工单 申请提升限制。
LimitExceeded.Memory	内存超出最大限制。
LimitExceeded.Timeout	Timeout超出最大限制。
ResourceInUse.FunctionName	FunctionName已存在。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.Region	Region错误。
UnsupportedOperation.CopyFunction	该函数无法复制。
UnsupportedOperation.Region	不支持此地域。

命名空间相关接口

删除命名空间

最近更新时间：2025-04-25 01:51:55

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入的参数删除命名空间。

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：DeleteNamespace。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Namespace	是	String	命名空间名称 示例值：namespace1

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 删除命名空间

输入示例

```
https://scf.tencentcloudapi.com/?Action=DeleteNamespace
&Namespace=namespace1
&<公共请求参数>
```

输出示例

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

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.DeleteNamespace	无法删除默认Namespace。
InternalServerError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Action	Action取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17235 。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.NamespaceInvalid	规则不正确，Namespace为英文字母、数字、- _ 符号组成，长度30。
ResourceInUse.Namespace	Namespace已存在。
ResourceNotFound.Namespace	Namespace不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnsupportedOperation	操作不支持。

更新命名空间

最近更新时间：2025-04-25 01:51:55

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

更新命名空间

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：UpdateNamespace。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Namespace	是	String	命名空间名称 示例值：test01
Description	是	String	命名空间描述 示例值：fdsgfds

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 更新命名空间

输入示例

```
https://scf.tencentcloudapi.com/?Action=UpdateNamespace
&Namespace=test01
&Description=fdsgfds
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "b2b845e1-d44a-4c3e-8314-b6b774b4e66f"
  }
}
```

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
InvalidParameterValue.Description	Description传入错误。
ResourceNotFound.Namespace	Namespace不存在。

创建命名空间

最近更新时间：2025-04-25 01:51:55

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

该接口根据传入的参数创建命名空间。

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：CreateNamespace。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Namespace	是	String	命名空间名称 示例值：ns1
Description	否	String	命名空间描述 示例值：a namespace
ResourceEnv	否	NamespaceResourceEnv	资源池配置

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 创建命名空间。

创建命名空间。

输入示例

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

{
  "Namespace": "ns1",
  "Description": "a ns"
}
```

输出示例

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

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.CreateNamespace	创建命名空间失败。
InternalServerError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Action	Action取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17235 。
InvalidParameterValue.DefaultNamespace	默认Namespace无法创建。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.NamespaceInvalid	规则不正确，Namespace为英文字母、数字、-、_ 符号组成，长度30。
InvalidParameterValue.Stamp	Stamp取值与规范不符，请修正后再试。
LimitExceeded.Namespace	命名空间数量超过最大限制，可通过 提交工单 申请提升限制。
ResourceInUse	资源被占用。
ResourceInUse.Namespace	Namespace已存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

错误码	描述
UnauthorizedOperation.Role	没有权限访问您的Cos资源。
UnsupportedOperation	操作不支持。

列出命名空间列表

最近更新时间：2025-04-25 01:51:55

-

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

列出命名空间列表

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：ListNamespaces。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	否	Integer	返回数据长度，默认值为 20 示例值：10
Offset	否	Integer	数据的偏移量，默认值为 0 示例值：0
Orderby	否	String	根据哪个字段进行返回结果排序,支持以下字段：Name,Updatetime 示例值：Name
Order	否	String	以升序还是降序的方式返回结果，可选值 ASC 和 DESC 示例值：DESC
SearchKey.N	否	Array of SearchKey	关键字匹配搜索，Key 可选值为 Namespace 和 Description，多个搜索条件之间是与的关系 示例值：{"Key":"Namespace":Value:"ns1"}

3. 输出参数

参数名称	类型	描述
Namespaces	Array of Namespace	namespace详情
TotalCount	Integer	返回的namespace数量 示例值：1
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 列出命名空间列表

列出命名空间列表

输入示例

```
https://scf.tencentcloudapi.com/?Action=ListNamespaces
&SearchKey.0.Key=Namespace
&SearchKey.0.Value=ns1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "Namespaces": [
      {
        "Name": "ns1",
        "Description": "",
        "AddTime": "2024-12-17 15:28:55",
        "ModTime": "2024-12-17 15:28:55",
        "Type": "default"
      },
      {
        "Name": "default",
        "Description": "",
        "AddTime": "2024-10-17 09:29:45",
        "ModTime": "2024-10-17 09:29:45",
        "Type": "default"
      }
    ],
    "RequestId": "sdfsdfs-3de5-459c-984a-cb43ec2625d2"
  }
}
```

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
InvalidParameterValue.Filters	Filters参数错误。
InvalidParameterValue.Order	Order传入错误。

错误码	描述
InvalidParameterValue.OrderBy	OrderBy传入错误。
InvalidParameterValue.SearchKey	searchkey 不是 Keyword,Tag 或者 Runtime。
InvalidParameterValue.Stamp	Stamp取值与规范不符，请修正后再试。

层管理相关接口

删除层版本

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

删除指定层的指定版本，被删除的版本无法再关联到函数上，但不会影响正在引用这个层的函数。

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：DeleteLayerVersion。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LayerName	是	String	层名称 示例值：layer-1
LayerVersion	是	Integer	版本号 示例值：v1

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 删除层版本

删除层版本

输入示例

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

{
  "LayerName": "layer-1",
  "LayerVersion": 1
}
```

输出示例

```
{
  "Response": {
    "RequestId": "a1ffbba5-5489-45bc-89c5-453e50d5386e"
  }
}
```

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.DeleteLayerVersion	删除layer版本失败。
FailedOperation.OperationConflict	操作冲突。
InternalServerError	内部错误。
InternalServerError.System	内部系统错误。
ResourceInUse.LayerVersion	Layer版本正在使用中。
ResourceNotFound.LayerVersion	Layer版本不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

获取层版本详细信息

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

获取层版本详细信息，包括用于下载层中文件的链接。

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：GetLayerVersion。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LayerName	是	String	层名称 示例值：layer-name1
LayerVersion	是	Integer	版本号 示例值：lv

3. 输出参数

参数名称	类型	描述
CompatibleRuntimes	Array of String	适配的运行时 示例值：demo
CodeSha256	String	层中版本文件的SHA256编码 示例值：xcdsfdfsfgsdgrfrew
Location	String	层中版本文件的下载地址 示例值：http://demo
AddTime	String	版本的创建时间 示例值：2023-01-01
Description	String	版本的描述信息 示例值：demo
LicenseInfo	String	许可证信息 示例值：LicenseInfoDemo
LayerVersion	Integer	版本号 示例值：11
LayerName	String	层名称 示例值：layer-name1
Status	String	层的具体版本当前状态，状态值 参考此处 示例值：Active
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 获取层版本详情

获取层版本详情

输入示例

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

{
  "LayerName": "layer-name1",
  "LayerVersion": 1
}
```

输出示例

```
{
  "Response": {
    "Status": "Active",
    "LayerVersion": 1,
    "CodeSha256": "c7c8bda601817ce4faab4c4fb3e73b9a32040e0e9f31f7a649d84d4bdde9e4a5",
    "Description": "desc",
    "LicenseInfo": "",
    "AddTime": "2019-11-26 16:15:33",
    "Location": "https://lambdalayerin-1253665819.cos.ap-mumbai.myqcloud.com/1253970226/layer3/1/1253970226_layer3_1.zip?sign=q-sign-algorithm%3Dsha1%26q-ak%3Dxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx%26q-sign-time%3D1582083327%3B1582093387%26q-key-tim%3D1582083327%3B1582093387%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D255d3e5aaa6c7224396b39f857e134a0dfb27c89&response-content-type=application/octet-stream",
    "CompatibleRuntimes": [
      "Nodejs8.9",
      "Nodejs6.10"
    ],
    "LayerName": "layer3",
    "RequestId": "6c7578ed-b816-4e50-930a-54e855452b8b"
  }
}
```

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。
InternalServerError.System	内部系统错误。
ResourceNotFound.LayerVersion	Layer版本不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

获取层版本列表

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

返回指定层的全部版本的信息

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：ListLayerVersions。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LayerName	是	String	层名称 示例值：layer-name1
CompatibleRuntime.N	否	Array of String	适配的运行时 示例值：Nodejs16.13

3. 输出参数

参数名称	类型	描述
LayerVersions	Array of LayerVersionInfo	层版本列表
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 获取层版本列表

获取层版本列表

输入示例

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

{
  "LayerName": "layer-name1",
  "CompatibleRuntime": [
    "Nodejs16.13"
```

```

    ]
  }
}

```

输出示例

```

{
  "Response": {
    "LayerVersions": [
      {
        "LayerName": "layer-name1",
        "LayerVersion": 1,
        "CompatibleRuntimes": [
          "Nodejs16.13"
        ],
        "Description": "vxox",
        "LicenseInfo": "",
        "AddTime": "2022-05-16 19:40:42",
        "Status": "Active",
        "Stamp": "default"
      }
    ],
    "RequestId": "ssssds-9783-424a-8f0b-48cfee89333b"
  }
}

```

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。
InternalServerError.System	内部系统错误。

错误码	描述
InvalidParameterValue.Name	Name参数传入错误。
InvalidParameterValue.Runtime	Runtime传入错误。
ResourceNotFound.Layer	layer不存在。
UnauthorizedOperation.CAM	CAM鉴权失败。

返回层列表

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

返回全部层的列表，其中包含了每个层最新版本的信息，可以通过适配运行时进行过滤。

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：ListLayers。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CompatibleRuntime	否	String	适配的运行时 示例值：Nodejs16.13
Offset	否	Integer	偏移位置 示例值：0
Limit	否	Integer	查询数目限制 示例值：10
SearchKey	否	String	查询key，模糊匹配名称 示例值：<SearchKey>

3. 输出参数

参数名称	类型	描述
Layers	Array of LayerVersionInfo	层列表
TotalCount	Integer	层总数 示例值：1
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 拉取层列表

拉取层列表

输入示例

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

<公共请求参数>

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

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "Layers": [
      {
        "LayerName": "test-layer",
        "LayerVersion": 1,
        "CompatibleRuntimes": [
          "Nodejs10.15"
        ],
        "Description": "Layer created by serverless component",
        "LicenseInfo": "",
        "AddTime": "2022-10-26 10:08:11",
        "Status": "Active",
        "Stamp": "default",
        "Tags": []
      }
    ]
  },
  "RequestId": "dsfdf-cac3-451d-acfa-667d831521b4"
}
```

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。
InternalError.System	内部系统错误。
InvalidParameterValue.Runtime	Runtime传入错误。
InvalidParameterValue.Stamp	Stamp取值与规范不符，请修正后再试。
LimitExceeded.Layers	layer数量超出最大限制。
UnauthorizedOperation.CAM	CAM鉴权失败。

发布层版本

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

使用给定的zip文件或cos对象创建一个层的新版本，每次使用相同的层的名称调用本接口，都会生成一个新版本。

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：PublishLayerVersion。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LayerName	是	String	层名称，支持26个英文字母大小写、数字、连接符和下划线，第一个字符只能以字母开头，最后一个字符不能为连接符或者下划线，名称长度1-64 示例值：layer-name1
CompatibleRuntimes.N	是	Array of String	层适用的运行时，可多选，可选的值对应函数的 Runtime 可选值。 示例值：Python2.7
Content	是	Code	层的文件来源或文件内容
Description	否	String	层的版本的描述 示例值：a layer desc
LicenseInfo	否	String	层的软件许可证 示例值：<LicenseInfo>
Tags.N	否	Array of Tag	层Tag 参数，以键值对数组形式传入

3. 输出参数

参数名称	类型	描述
LayerVersion	Integer	本次创建的层的版本号 示例值：1
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 发布层版本

发布层版本

输入示例


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

{
  "LayerName": "layer-name1",
  "CompatibleRuntimes": [
    "Nodejs16.13"
  ],
  "Content": {
    "CosBucketName": "layer-name1-2343243543",
    "CosObjectName": "layer-name1/node_modules.zip",
    "CosBucketRegion": "ap-singapore"
  },
  "Description": "a layer"
}
```

输出示例

```
{
  "Response": {
    "LayerVersion": 8,
    "RequestId": "197192d3-0b64-410a-8cd9-25cb7c1db0ee"
  }
}
```

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.PublishLayerVersion	发布layer版本失败。
InternalServerError	内部错误。
InternalServerError.Exception	内部服务异常。
InternalServerError.System	内部系统错误。
InvalidParameterValue.CompatibleRuntimes	CompatibleRuntimes参数传入错误。
InvalidParameterValue.Content	Content参数传入错误。
InvalidParameterValue.Cos	Cos传入错误。
InvalidParameterValue.CosBucketName	CosBucketName不符合规范。
InvalidParameterValue.CosBucketRegion	CosBucketRegion取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#Code 。
InvalidParameterValue.CosObjectName	CosObjectName不符合规范。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.LayerName	LayerName参数传入错误。
InvalidParameterValue.Name	Name参数传入错误。
InvalidParameterValue.Runtime	Runtime传入错误。
InvalidParameterValue.Stamp	Stamp取值与规范不符，请修正后再试。
InvalidParameterValue.TempCosObjectName	非法的TempCosObjectName。
InvalidParameterValue.ZipFileBase64BinasciiError	压缩文件base64解码失败: <code>Incorrect padding</code> ，请修正后再试。
LimitExceeded.LayerVersions	layer版本数量超出最大限制。
ResourceInUse	资源被占用。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.NotMC	不是从控制台调用的该接口。
UnsupportedOperation	操作不支持。
UnsupportedOperation.Cos	Cos操作不支持。

异步事件管理相关接口

拉取函数异步事件列表

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

拉取函数异步事件列表

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：ListAsyncEvents。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
FunctionName	是	String	函数名称 示例值：helloworld
Namespace	否	String	命名空间 示例值：default
Qualifier	否	String	过滤条件，函数版本 示例值：\$LATEST
InvokeType.N	否	Array of String	过滤条件，调用类型列表 示例值：[COS]
Status.N	否	Array of String	过滤条件，事件状态列表 示例值：RUNNING
StartTimeInterval	否	TimeInterval	过滤条件，开始执行时间左闭右开区间
EndTimeInterval	否	TimeInterval	过滤条件，结束执行时间左闭右开区间
Order	否	String	可选值 ASC 和 DESC，默认 DESC 示例值：DESC
Orderby	否	String	可选值 StartTime 和 EndTime，默认值 StartTime 示例值：StartTime
Offset	否	Integer	数据偏移量，默认值为 0 示例值：0
Limit	否	Integer	返回数据长度，默认值为 20，最大值 100 示例值：20
InvokeRequestId	否	String	过滤条件，事件调用请求id 示例值：5f825fd3-a59d-4879-aa28-3ac967c128885f825fd3-a59d-4879-aa28-3ac9

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	满足过滤条件的事件总数 示例值：100
EventList	Array of AsyncEvent	异步事件列表
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 拉取函数异步事件列表

输入示例

```
https://scf.tencentcloudapi.com/?Action=ListAsyncEvents
&FunctionName=helloworld
&Qualifier=1
&Offset=0
&Limit=20
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "EventList": [
      {
        "InvokeRequestId": "xxxxxxxxxxxxxx",
        "Qualifier": "1",
        "Status": "FINISHED",
        "InvokeType": "COS",
        "StartTime": "2020-12-01 08:35:00.000",
        "EndTime": "2020-12-01 08:35:30.000"
      },
      {
        "InvokeRequestId": "xxxxxxxxxxxxxx",
        "Qualifier": "1",
        "Status": "RUNNING",
        "InvokeType": "COS",
        "StartTime": "2020-12-01 08:35:00.000",
        "EndTime": ""
      }
    ],
    "RequestId": "26c5f1a9-ebd4-407e-83d7-ae8001cca644"
  }
}
```

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
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.InvokeType	InvokeType取值与规范不符，请修正后再试。
InvalidParameterValue.Limit	Limit传入错误。
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.Order	Order传入错误。
InvalidParameterValue.Orderby	Orderby传入错误。
InvalidParameterValue.Status	Status取值与规范不符，请修正后再试。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Version	版本不存在。

终止函数异步事件

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

终止正在运行中的函数异步事件

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：TerminateAsyncEvent。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
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-chengdu, ap-chongqing, ap-guanghongkong, ap-jakarta, ap-nanjing, ap-seoul, ap-shanghai, ap-shanghai-fsi, ap-shenzhen-fsi, ap-singapore, ap-tokyo, eu-frankfurt, na-siliconvalley, sa-saopaulo。
FunctionName	是	String	函数名称 示例值：helloworld
InvokeRequestId	是	String	终止的调用请求id 示例值：5f825fd3-a59d-4879-aa28-3ac967c128885f825fd3-a59d-4879-aa28-3ac9
Namespace	否	String	命名空间 示例值：default
GraceShutdown	否	Boolean	true，向指定请求[发送 SIGTERM 终止信号] (https://cloud.tencent.com/document/product/583/63969#.E5.8F.91.E9.80.81.E7.BB.88.E6.AD.A2.E4.BA.BE)，默认值为 false。 示例值：false

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 终止异步事件

输入示例

```
https://scf.tencentcloudapi.com/?Action=TerminateAsyncEvent
&Namespace=default
&FunctionName=helloworld
&InvokeRequestId=b379787f-8487-4bac-bd8c-17a20a9ce40e
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "ee33a89b-3825-4d2f-bd88-35a8fa27aae1"
  }
}
```

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.AsyncEventStatus	当前异步事件状态不支持此操作，请稍后重试。
InvalidParameterValue.FunctionName	函数不存在。
ResourceNotFound.AsyncEvent	未找到指定的AsyncEvent，请创建后再试。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。

获取函数异步事件状态

最近更新时间：2025-04-25 01:51:54

1. 接口描述

接口请求域名：scf.tencentcloudapi.com。

获取函数异步执行事件状态，事件状态保留 3 * 24 小时（从事件完成开始计时）。

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：GetAsyncEventStatus。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
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-chengdu, ap-chongqing, ap-guangzhou, ap-hongkong, ap-jakarta, ap-nanjing, ap-seoul, ap-shanghai, ap-shanghai-fsi, ap-shenzhen-fsi, ap-singapore, ap-tokyo, eu-frankfurt, na-siliconvalley, sa-saopaulo。
InvokeRequestId	是	String	异步执行请求 id 示例值：ea7bb7c7-c312-4255-b7dc-b2bf0cfda10e

3. 输出参数

参数名称	类型	描述
Result	AsyncEventStatus	异步事件状态
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 查询函数异步事件状态

查询函数异步事件状态

输入示例

```
https://scf.tencentcloudapi.com/?Action=GetAsyncEventStatus
&InvokeRequestId=b379787f-8487-4bac-bd8c-17a20a9ce40e
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Result": {
      "Status": "RUNNING",
      "StatusCode": 0,
      "InvokeRequestId": "5f825fd3-a59d-4879-aa28-3ac967c12888"
```



```
},
"RequestId": "5f825fd3-a59d-4879-aa28-3ac967c128885f82"
}
```

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.AsyncEventStatus	当前异步事件状态不支持此操作，请稍后重试。
ResourceNotFound.AsyncEvent	未找到指定的AsyncEvent，请创建后再试。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.Namespace	Namespace不存在。

其他接口

获取账户信息

最近更新时间：2025-04-25 01:51:59

1. 接口描述

接口请求域名：`scf.tencentcloudapi.com`。

获取账户信息

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：GetAccount。
Version	是	String	公共参数 ，本接口取值：2018-04-16。
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
AccountUsage	UsageInfo	命名空间已使用的信息
AccountLimit	LimitsInfo	命名空间限制的信息
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 获取账户信息

输入示例

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

输出示例

```
{
  "Response": {
    "AccountUsage": {
      "NamespacesCount": 1,
      "Namespace": [
        {
          "Namespace": "default",
          "Functions": [
```

```
"express_demo-test"
],
"TotalConcurrencyMem": 0,
"TotalAllocatedConcurrencyMem": 0,
"TotalAllocatedProvisionedMem": 0,
"FunctionsCount": 1
}
],
"TotalConcurrencyMem": 128000,
"TotalAllocatedConcurrencyMem": 0,
"UserConcurrencyMemLimit": 128000
},
"AccountLimit": {
  "NamespacesCount": 5,
  "Namespace": [
    {
      "Namespace": "default",
      "FunctionsCount": 50,
      "TimeoutLimit": 900,
      "InitTimeoutLimit": 300,
      "TestModelLimit": 20,
      "RetryNumLimit": 2,
      "MinMsgTTL": 60,
      "MaxMsgTTL": 21600,
      "ConcurrentExecutions": 100,
      "Trigger": {
        "Apigw": 10,
        "Ckafka": 10,
        "Clb": 10,
        "Cls": 10,
        "Cm": 10,
        "Cmq": 10,
        "Cos": 10,
        "Eb": 10,
        "Mps": 10,
        "Timer": 10,
        "Total": 110,
        "Vod": 10
      }
    }
  ]
},
"RequestId": "cvvdsds-af20-42a9-9314-84ad66b06ee4"
}
```

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
InvalidParameterValue.Stamp	Stamp取值与规范不符，请修正后再试。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnsupportedOperation	操作不支持。

数据结构

最近更新时间：2025-05-22 01:27:06

AccessInfo

HTTP域名相关信息

被如下接口引用：GetFunction。

名称	类型	必选	描述
Host	String	是	域名 示例值：baidu.com
Vip	String	是	VIP 示例值：10.0.0.3

Alias

函数的版本别名

被如下接口引用：ListAliases。

名称	类型	必选	描述
FunctionVersion	String	是	别名指向的主版本 示例值：1
Name	String	是	别名的名称 示例值：alias1
RoutingConfig	RoutingConfig	是	别名的路由信息 示例值：{AdditionalVersionWeights: [{Version: "\$LATEST", Weight: 0.8}]}
Description	String	否	描述信息 示例值：alias1
AddTime	Timestamp	否	创建时间 示例值："2024-12-19 12:19:07"
ModTime	Timestamp	否	更新时间 示例值："2024-12-19 12:19:07"

AsyncEvent

异步事件

被如下接口引用：ListAsyncEvents。

名称	类型	描述
InvokeRequestId	String	调用请求id 示例值：xxxx-yyyy-zzzz
InvokeType	String	调用类型 示例值：TRIGGER_TIMER
Qualifier	String	函数版本 示例值：\$LATEST
Status	String	事件状态，RUNNING 表示运行中，FINISHED 表示调用成功，ABORTED 表示调用终止，FAILED 表示调用失败 示例值：FINISHED
StartTime	String	调用开始时间，格式："%Y-%m-%d %H:%M:%S.%f" 示例值：2021-01-01 00:00:00.001

名称	类型	描述
EndTime	String	调用结束时间，格式: "%Y-%m-%d %H:%M:%S.%f" 示例值: 2021-01-01 00:00:00.002

AsyncEventStatus

异步事件状态

被如下接口引用: GetAsyncEventStatus。

名称	类型	描述
Status	String	异步事件状态, RUNNING 表示运行中, FINISHED 表示调用成功, ABORTED 表示调用终止, FAILED 表示调用失败。 示例值: RUNNING
StatusCode	Integer	请求状态码 示例值: 200
InvokeRequestId	String	异步执行请求 Id 示例值: ea7bb7c7-c312-4255-b7dc-b2bf0cfda10e

AsyncTriggerConfig

函数的异步重试配置详情

被如下接口引用: GetFunctionEventInvokeConfig, UpdateFunctionEventInvokeConfig。

名称	类型	必选	描述
RetryConfig	Array of RetryConfig	是	用户错误的异步重试重试配置 示例值: {"MsgTTL":21600,"RetryConfig":[{"RetryNum":2}]}
MsgTTL	Integer	是	消息保留时间 示例值: 2

CertConf

证书配置

被如下接口引用: CreateCustomDomain, GetCustomDomain, ListCustomDomains, UpdateCustomDomain。

名称	类型	必选	描述
CertificateId	String	否	ssl证书ID 示例值: Hz4OcOUu

CfsConfig

文件系统(cfs)配置描述

被如下接口引用: CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
CfsInsList	Array of CfsInsInfo	是	文件系统信息列表

CfsInsInfo

云函数关联的cfs配置信息

被如下接口引用: CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
UserId	String	是	用户id 示例值: 10000
UserGroupId	String	是	用户组id 示例值: 10000
CfsId	String	是	文件系统实例id 示例值: cfs-xxxxxxxxx
MountInsId	String	是	文件系统挂载点id 示例值: cfs-xxxxxxxxx
LocalMountDir	String	是	本地挂载点 示例值: /mnt/
RemoteMountDir	String	是	远程挂载点 示例值: /remotedir
IpAddress	String	否	文件系统ip, 配置 cfs 时无需填写。 注意: 此字段可能返回 null, 表示取不到有效值。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 10.0.0.3
MountVpcId	String	否	文件系统所在的私有网络id, 配置 cfs 时无需填写。 注意: 此字段可能返回 null, 表示取不到有效值。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: vpc-xxxxxxxxx
MountSubnetId	String	否	文件系统所在私有网络的子网id, 配置 cfs 时无需填写。 注意: 此字段可能返回 null, 表示取不到有效值。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: subnet-xxxxxxxxx

Code

函数代码

被如下接口引用: CreateFunction, PublishLayerVersion, UpdateFunctionCode。

名称	类型	必选	描述
CosBucketName	String	否	对象存储桶名称 (填写存储桶名称自定义部分, 不包含-appid) 示例值: BucketName
CosObjectName	String	否	对象存储中代码包文件路径, 以/开头 示例值: ObjectName
ZipFile	String	否	包含函数代码文件及其依赖项的 zip 格式文件, zip包大小上限为 50MB, 使用该接口时要求将 zip 文件的内容转成 base64 编码 示例值: base64编码
CosBucketRegion	String	否	对象存储的地域, 地域为北京时需要传入ap-beijing,北京一区时需要传递ap-beijing-1, 其他的地域不需要传递。 示例值: ap-chongqing
Demold	String	否	如果是通过Demo创建的话, 需要传入Demold 示例值: Demo的Id
TempCosObjectName	String	否	如果是从TempCos创建的话, 需要传入TempCosObjectName 示例值: TempCosObjectName
GitUrl	String	否	Git地址。该功能已下线。 示例值: Git地址
GitUserName	String	否	Git用户名。该功能已下线。 示例值: Git用户名
GitPassword	String	否	Git密码。该功能已下线。

名称	类型	必选	描述
			示例值：Git密码
GitPasswordSecret	String	否	加密后的Git密码，一般无需指定。该功能已下线。 示例值：加密后的Git密码
GitBranch	String	否	Git分支。该功能已下线。 示例值：Git分支
GitDirectory	String	否	代码在Git仓库中的路径。该功能已下线。 示例值：代码在Git仓库中的路径
GitCommitId	String	否	指定要拉取的版本。该功能已下线。 示例值：Git拉取的版本
GitUserNameSecret	String	否	加密后的Git用户名，一般无需指定。该功能已下线。 示例值：加密后的Git用户名
ImageConfig	ImageConfig	否	镜像部署时配置TCR镜像信息 示例值：TCR镜像配置

DeadLetterConfig

死信队列参数

被如下接口引用：CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
Type	String	是	死信队列模式 示例值：CMQ-QUEUE
Name	String	是	死信队列名称 示例值：cmq-test-queue
FilterType	String	否	死信队列主题模式的标签形式 示例值：暂无使用

DomainInfo

云函数自定义域名详情

被如下接口引用：ListCustomDomains。

名称	类型	必选	描述
Domain	String	是	域名，不支持泛域名 示例值：www.demo.com
Protocol	String	否	协议，取值范围：HTTP, HTTPS, HTTP&HTTPS 示例值：HTTP
EndpointsConfig	Array of EndpointsConf	否	路由配置信息
CertConfig	CertConf	否	证书配置信息，HTTPS协议必传路由配置
WafConfig	WafConf	否	web 应用防火墙配置
Tags	Array of Tag	否	标签

EipConfigIn

公网访问固定ip配置

被如下接口引用：CreateFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
EipStatus	String	是	Eip开启状态，取值['ENABLE','DISABLE'] 示例值：DISABLE

EipConfigOut

公网访问固定ip配置

被如下接口引用：GetFunction。

名称	类型	描述
EipStatus	String	是否是固定IP，["ENABLE","DISABLE"] 示例值：DISABLE
EipAddress	Array of String	IP列表 示例值：1.1.1.1

EipOutConfig

EipOutConfig

被如下接口引用：GetFunction。

名称	类型	描述
EipFixed	String	是否是固定IP，["TRUE","FALSE"] 示例值：FALSE
Eips	Array of String	IP列表 示例值：["1.1.1.1"]

EndpointsConf

后端路由配置信息

被如下接口引用：CreateCustomDomain, GetCustomDomain, ListCustomDomains, UpdateCustomDomain。

名称	类型	必选	描述
Namespace	String	是	函数命名空间 示例值：default
FunctionName	String	是	函数名 示例值：function-demo
Qualifier	String	是	函数别名或版本 示例值：\$DEFAULT
PathMatch	String	是	路径,取值规范：/, /, /xxx, /xxx/a, /xxx/" 示例值：/*
PathRewrite	Array of PathRewriteRule	否	路径重写策略 注意：此字段可能返回 null，表示取不到有效值。

Environment

函数的环境变量参数

被如下接口引用：CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
Variables	Array of Variable	否	环境变量数组

Filter

描述键值对过滤器，用于条件过滤查询。例如过滤ID、名称、状态等

若存在多个Filter时，Filter间的关系为逻辑与（AND）关系。

若同一个Filter存在多个Values，同一Filter下Values间的关系为逻辑或（OR）关系。

被如下接口引用：ListCustomDomains, ListFunctions, ListTriggers。

名称	类型	必选	描述
Name	String	是	需要过滤的字段。过滤条件数量限制为10。 Name可选值：VpcId, SubnetId, ClsTopicId, ClsLogsetId, Role, CfsId, CfsMountInsId, Eip; Values 长度限制为1。 Name可选值：Status, Runtime, FunctionType, PublicNetStatus, AsyncRunEnable, TraceEnable; Values 长度限制为20。 当 Name = Runtime 时，CustomImage 表示过滤镜像类型函数。 示例值：Status
Values	Array of String	是	字段的过滤值。 示例值：["Creating", "Updating", "Deleting"]

Function

函数列表

被如下接口引用：ListFunctions。

名称	类型	描述
ModTime	String	修改时间 示例值：2021-01-10 23:49:31
AddTime	String	创建时间 示例值：2021-01-10 23:49:31
Runtime	String	运行时 注意：此字段可能返回 null，表示取不到有效值。 示例值：python2.7
FunctionName	String	函数名称 示例值：helloworld
FunctionId	String	函数ID 示例值：lam-functionid
Namespace	String	命名空间 示例值：Default
Status	String	函数状态，状态值及流转 参考此处 示例值：Active
StatusDesc	String	函数状态详情 示例值："
Description	String	函数描述 示例值：a function
Tags	Array of Tag	函数标签 示例值：[{"Key":"auto-test2","Value":"event"}, {"Key":"var6","Value":"6666"}, {"Key":"cccc","Value":"3333"}]
Type	String	函数类型，取值为 HTTP 或者 Event 示例值：Event
StatusReasons	Array of StatusReason	函数状态失败原因 示例值："
TotalProvisionedConcurrencyMem	Integer	函数所有版本预置并发内存总和 注意：此字段可能返回 null，表示取不到有效值。 示例值：128000

名称	类型	描述
ReservedConcurrencyMem	Integer	函数并发保留内存 注意：此字段可能返回 null，表示取不到有效值。 示例值：12800
AsyncRunEnable	String	函数异步属性，取值 TRUE 或者 FALSE 示例值：FALSE
TraceEnable	String	异步函数是否开启调用追踪，取值 TRUE 或者 FALSE 示例值：FALSE

FunctionLog

日志信息

被如下接口引用：GetFunctionLogs。

名称	类型	描述
FunctionName	String	函数的名称 示例值：function-demo
RetMsg	String	函数执行完成后的返回值 示例值："Hello World"
RequestId	String	执行该函数对应的requestId 示例值：633934da-c2b6-11ef-87ca-5254007bbe22
StartTime	String	函数开始执行时的时间点 示例值：2024-12-25 19:50:07
RetCode	Integer	函数执行结果，如果是 0 表示执行成功，2表示函数运行中，3表示函数执行中断，其他值表示失败 示例值：0
InvokeFinished	Integer	函数调用是否结束，如果是 1 表示执行结束，其他值表示调用异常 示例值：1
Duration	Float	函数执行耗时，单位为 ms 示例值：8
BillDuration	Integer	函数计费时间，根据 duration 向上取最近的 100ms，单位为ms 示例值：8
MemUsage	Integer	函数执行时消耗实际内存大小，单位为 Byte 示例值：8253440
Log	String	函数执行过程中的日志输出 示例值：START RequestId: 633934da-c2b6-11ef-87ca-5254007bbe22
RetryNum	Integer	重试次数 示例值：0

FunctionVersion

函数版本信息

被如下接口引用：ListVersionByFunction。

名称	类型	必选	描述
Version	String	是	函数版本名称 示例值：\$LATEST
Description	String	否	版本描述信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：a version
AddTime	Timestamp	否	创建时间 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	必选	描述
			示例值：2024-12-19 12:00:00
ModTime	Timestamp	否	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2024-12-19 12:00:00
Status	String	否	版本状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：Active

ImageConfig

TCR镜像信息描述

被如下接口引用：CreateFunction, GetFunction, PublishLayerVersion, UpdateFunctionCode。

名称	类型	必选	描述
ImageType	String	是	镜像仓库类型，个人版或者企业版：personal/enterprise 示例值：enterprise
ImageUri	String	是	{domain}/{namespace}/{imageName}:{tag}@{digest} 示例值：tcr.tencentcloudcr.com/namespace/test:latest
RegistryId	String	否	用于企业版TCR获取镜像拉取临时凭证，ImageType为"enterprise"时必须填 注意：此字段可能返回 null，表示取不到有效值。 示例值：tcr-xxx
EntryPoint	String	否	该参数即将下线，不推荐用户使用 注意：此字段可能返回 null，表示取不到有效值。 示例值："entrypoint.sh"
Command	String	否	容器的启动命令。该参数为可选参数，如果不填写，则默认使用 Dockerfile 中的 Entrypoint。传入规范，填写可运行的指令，例如 python 注意：此字段可能返回 null，表示取不到有效值。 示例值：python
Args	String	否	容器的启动参数。该参数为可选参数，如果不填写，则默认使用 Dockerfile 中的 CMD。传入规范，以“空格”作为参数的分割标识，例如 -u app.py 注意：此字段可能返回 null，表示取不到有效值。 示例值：-u app.py
ContainerImageAccelerate	Boolean	否	镜像加速开关，默认False 注意：此字段可能返回 null，表示取不到有效值。 示例值：False
ImagePort	Integer	否	镜像函数端口设置，可指定镜像类型 Web Server镜像：9000 Job 镜像：-1 注意：此字段可能返回 null，表示取不到有效值。 默认值：9000 示例值：9000 注意：此字段可能返回 null，表示取不到有效值。 示例值：9000

InstanceConcurrencyConfig

多并发执行配置描述

被如下接口引用：CreateFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
DynamicEnabled	String	否	是否开启智能动态并发。'FALSE'时是静态并发。"时取消多并发配置。 注意：此字段可能返回 null，表示取不到有效值。 示例值：'FALSE'

名称	类型	必选	描述
MaxConcurrency	Integer	否	单实例并发数最大值。取值范围 [1,100] 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

IntranetConfigIn

内网配置

被如下接口引用：CreateFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
IpFixed	String	是	是否开启固定内网IP ENABLE 为开启 DISABLE 为不开启 示例值：ENABLE

IntranetConfigOut

内网配置

被如下接口引用：GetFunction。

名称	类型	描述
IpFixed	String	是否启用固定内网IP ENABLE 为启用 DISABLE 为不启用 示例值：ENABLE
IpAddress	Array of String	若已启用固定内网IP，则该字段返回使用的IP列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：1.1.1.1

K8SLabel

k8s label

被如下接口引用：CreateNamespace。

名称	类型	必选	描述
Key	String	是	label的名称 示例值：labelkey
Value	String	是	label的值 示例值：labelvalue

K8SToleration

Kubernetes污点容忍，使用时请注意您的Kubernetes版本所支持的字段情况。

可参考 <https://kubernetes.io/zh-cn/docs/concepts/scheduling-eviction/taint-and-toleration/>

被如下接口引用：CreateNamespace。

名称	类型	必选	描述
Key	String	否	匹配的污点名 示例值：key1
Operator	String	否	匹配方式，默认值为：Equal 注意：此字段可能返回 null，表示取不到有效值。 示例值：Equal

名称	类型	必选	描述
Effect	String	否	执行策略 注意：此字段可能返回 null，表示取不到有效值。 示例值：NoExecute
Value	String	否	匹配的污点值，当Operator为Equal时必须填 注意：此字段可能返回 null，表示取不到有效值。 示例值：value1
TolerationSeconds	Integer	否	当污点不被容忍时，Pod还能在节点上运行多久 注意：此字段可能返回 null，表示取不到有效值。 示例值：3600

LayerVersionInfo

层版本信息

被如下接口引用：GetFunction, ListLayerVersions, ListLayers。

名称	类型	描述
CompatibleRuntimes	Array of String	版本适用的运行时 示例值：["Python3.10","Python3.9"]
AddTime	String	创建时间 示例值：2024-11-22 16:52:21
Description	String	版本描述 示例值：a layer
LicenseInfo	String	许可证信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：""
LayerVersion	Integer	版本号 示例值：1
LayerName	String	层名称 示例值：layer-name1
Status	String	层的具体版本当前状态，状态值 参考此处 示例值：Active
Stamp	String	Stamp 示例值：Default
Tags	Array of Tag	返回层绑定的标签信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"Key":"xx","Value":"xx-xx"}]

LayerVersionSimple

指定某个Layer版本

被如下接口引用：CreateFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
LayerName	String	是	绑定的层名称。解绑层需传递空字符串。 示例值：layername
LayerVersion	Integer	是	绑定或解绑层的版本号。解绑函数版本关联的最后一个层版本时，LayerVersion 填 0。 示例值：1

LimitsInfo

限制信息

被如下接口引用：GetAccount。

名称	类型	描述
NamespacesCount	Integer	命名空间个数限制 示例值：5
Namespace	Array of NamespaceLimit	命名空间限制信息

LogFilter

日志过滤条件，用于区分正确与错误日志

被如下接口引用：GetFunctionLogs。

名称	类型	必选	描述
RetCode	String	否	filter.RetCode的取值有： not0 表示只返回错误日志， is0 表示只返回正确日志， TimeLimitExceeded 返回函数调用发生超时的日志， ResourceLimitExceeded 返回函数调用发生资源超限的日志， UserCodeException 返回函数调用发生用户代码错误的日志， 无输入则返回所有日志。 示例值：is0

LogSearchContext

日志搜索上下文

被如下接口引用：GetFunctionLogs。

名称	类型	必选	描述
Offset	String	是	偏移量 示例值：0
Limit	Integer	是	日志条数 示例值：100
Keyword	String	否	日志关键词 示例值：Some Keyword
Type	String	否	日志类型，支持Application和Platform，默认为Application 示例值：Application

Namespace

命名空间

被如下接口引用：ListNamespaces。

名称	类型	描述
ModTime	Timestamp	命名空间创建时间 示例值：2024-12-19 10:00:00
AddTime	Timestamp	命名空间修改时间 示例值：2024-12-19 10:00:00
Description	String	命名空间描述 示例值：a namespace desc
Name	String	命名空间名称 示例值：ns1
Type	String	默认default，TCB表示是小程序云开发创建的 示例值：Default

NamespaceLimit

命名空间限制

被如下接口引用：GetAccount。

名称	类型	描述
FunctionsCount	Integer	函数总数 示例值：0
Trigger	TriggerCount	Trigger信息 示例值： {"Apigw":10,"Ckafka":10,"Clb":10,"Cls":10,"Cm":10,"Cmq":10,"Cos":10,"Eb":10,"Mps":10,"Timer":10,
Namespace	String	Namespace名称 示例值：namespace1
ConcurrentExecutions	Integer	并发量 示例值：0
TimeoutLimit	Integer	Timeout限制 示例值：0
TestModelLimit	Integer	测试事件限制 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
InitTimeoutLimit	Integer	初始化超时限制 示例值：0
RetryNumLimit	Integer	异步重试次数限制 示例值：0
MinMsgTTL	Integer	异步重试消息保留时间下限 示例值：0
MaxMsgTTL	Integer	异步重试消息保留时间上限 示例值：0

NamespaceResourceEnv

命名空间资源池配置

被如下接口引用：CreateNamespace。

名称	类型	必选	描述
TKE	NamespaceResourceEnvTKE	否	基于TKE集群的资源池 注意：此字段可能返回 null，表示取不到有效值。 示例值：x
OFFLINE	Boolean	否	近离线计算类型的命名空间 示例值：true

NamespaceResourceEnvTKE

基于TKE的资源池选项

被如下接口引用：CreateNamespace。

名称	类型	必选	描述
ClusterID	String	是	集群ID 示例值：cls-xxxx
SubnetID	String	是	子网ID 示例值：subnet-xx

名称	类型	必选	描述
Namespace	String	是	命名空间 示例值：kube-system
DataPath	String	否	数据存储地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：/var/lib/scf
NodeSelector	Array of K8SLabel	否	node选择器 注意：此字段可能返回 null，表示取不到有效值。 示例值：Key0=a,Value0=x
Tolerations	Array of K8SToleration	否	污点容忍 注意：此字段可能返回 null，表示取不到有效值。 示例值：Key: "key1" Operator: "Exists"
Port	Integer	否	scf组件将占用的节点端口起始号 注意：此字段可能返回 null，表示取不到有效值。 示例值：38002
PodTemplatePatch	String	否	yaml格式的pod patch内容，例如 metadata: labels: key: value 注意：此字段可能返回 null，表示取不到有效值。 示例值：metadata: labels: key: value

NamespaceUsage

名称空间已使用信息

被如下接口引用：GetAccount。

名称	类型	描述
Functions	Array of String	函数数组 示例值：func1
Namespace	String	命名空间名称 示例值：default
FunctionsCount	Integer	命名空间函数个数 示例值：1
TotalConcurrencyMem	Integer	命名空间配额总量 示例值：0
TotalAllocatedConcurrencyMem	Integer	命名空间并发使用量 示例值：0
TotalAllocatedProvisionedMem	Integer	命名空间预置使用量 示例值：0

PathRewriteRule

路径路由重写规则

被如下接口引用：CreateCustomDomain, GetCustomDomain, UpdateCustomDomain。

名称	类型	必选	描述
Path	String	是	需要重路由的路径，取值规范：/, /, /xxx, /xxx/a, /xxx/ 示例值：/aa/
Type	String	是	匹配规，取值范围：WildcardRules 通配符匹配，ExactRules 精确匹配 示例值：ExactRules
Rewrite	String	是	替换值：比如/, /\$

名称	类型	必选	描述
			示例值：/\$1

ProtocolParams

HTTP函数支持其他访问协议的参数

被如下接口引用：CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
WSParams	WSParams	否	WebSockets协议支持的参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：{ IdleTimeout: 15 }

PublicNetConfigIn

公网访问配置

被如下接口引用：CreateFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
PublicNetStatus	String	是	是否开启公网访问能力取值['DISABLE','ENABLE'] 示例值：DISABLE
EipConfig	EipConfigIn	否	Eip配置 示例值：{"EipStatus":"DISABLE"}

PublicNetConfigOut

公网访问配置

被如下接口引用：GetFunction。

名称	类型	描述
PublicNetStatus	String	是否开启公网访问能力取值['DISABLE','ENABLE'] 示例值：ENABLE
EipConfig	EipConfigOut	Eip配置 示例值：{"EipStatus":"ENABLE","EipAddress":"1.1.1.1"}

RequestStatus

函数运行状态

被如下接口引用：GetRequestStatus。

名称	类型	描述
FunctionName	String	函数的名称 示例值：function-demo
RetMsg	String	函数执行完成后的返回值 示例值："Hello World"
RequestId	String	查询的请求 id 示例值："RequestId": "a63607c3-581f-4be9-8c4b-631c75909f07",
StartTime	String	请求开始时间 示例值：2024-12-25 19:50:07
RetCode	Integer	请求执行结果，0 表示执行成功，1表示运行中，-1 表示执行异常。 示例值：0

名称	类型	描述
Duration	Float	请求运行耗时，单位：ms 示例值：8
MemUsage	Float	请求消耗内存，单位为 MB 示例值：8253440
RetryNum	Integer	重试次数 示例值：0

Result

运行函数的返回

被如下接口引用：Invoke, InvokeFunction。

名称	类型	描述
Log	String	表示执行过程中的日志输出，异步调用返回为空 示例值：""
RetMsg	String	表示执行函数的返回，异步调用返回为空 示例值："hello from scf"
ErrMsg	String	表示执行函数的错误返回信息，异步调用返回为空 示例值：""
MemUsage	Integer	执行函数时的内存大小，单位为Byte，异步调用返回为空 示例值：32
Duration	Float	表示执行函数的耗时，单位是毫秒，异步调用返回为空 示例值：100
BillDuration	Integer	表示函数的计费耗时，单位是毫秒，异步调用返回为空 示例值：100
FunctionRequestId	String	此次函数执行的Id 示例值：c2af8a64-c922-4d55-ae0-bd86a5c2cd12
InvokeResult	Integer	该参数不再维护，不推荐用户继续使用。 示例值：200

RetryConfig

异步重试配置

被如下接口引用：GetFunctionEventInvokeConfig, UpdateFunctionEventInvokeConfig。

名称	类型	必选	描述
RetryNum	Integer	是	重试次数 示例值：2

RoutingConfig

别名的版本路由配置

其中：随机权重路由附加版本和规则路由附加版本不可以同时配置

被如下接口引用：CreateAlias, GetAlias, ListAliases, UpdateAlias。

名称	类型	必选	描述
AdditionalVersionWeights	Array of VersionWeight	否	随机权重路由附加版本 示例值：[{"Version":"1", "Weight":0.01}]
AdditionVersionMatches	Array of VersionMatch	否	规则路由附加版本 示例值：[{"Version":"1", "Key":"invoke.headers.User", "Method":"exact",

名称	类型	必选	描述
			"Expression":"testuser" }]

SearchKey

包含搜索关键字和对应的内容

被如下接口引用：ListNamespaces。

名称	类型	必选	描述
Key	String	是	搜索关键字 示例值：SearchKey1
Value	String	是	搜索内容 示例值：SearchKeyValue

StatusReason

状态原因描述

被如下接口引用：GetFunction, ListFunctions。

名称	类型	描述
ErrorCode	String	错误码 示例值：ResourceNotFound.xxx
ErrorMessage	String	错误描述 示例值：The specified xxx is not found

Tag

函数标签

被如下接口引用：CreateCustomDomain, CreateFunction, GetCustomDomain, GetFunction, ListCustomDomains, ListFunctions, ListLayerVersions, ListLayers, PublishLayerVersion。

名称	类型	必选	描述
Key	String	是	标签的key 示例值：运营部门
Value	String	是	标签的value 示例值：xxx产品中心

TimeInterval

左闭右开时间区间，包括起始时间和结束时间，格式为"%Y-%m-%d %H:%M:%S"

被如下接口引用：ListAsyncEvents。

名称	类型	必选	描述
Start	String	是	起始时间（包括在内），格式"%Y-%m-%d %H:%M:%S" 示例值：2024-12-14 00:00:00
End	String	是	结束时间（不包括在内），格式"%Y-%m-%d %H:%M:%S" 示例值：2024-12-14 01:00:00

Trigger

触发器类型

被如下接口引用：CreateTrigger, GetFunction。

名称	类型	描述
ModTime	Timestamp	触发器最后修改时间 示例值：2024-12-20 11:27:07
Type	String	触发器类型 示例值：timer
TriggerDesc	String	触发器详细配置 示例值：{"cron": "0 0 0 */1 * *"}
TriggerName	String	触发器名称 示例值：SCF-timer-1734665224
AddTime	Timestamp	触发器创建时间 示例值：2024-12-20 11:27:07
Enable	Integer	使能开关 示例值：1
CustomArgument	String	客户自定义参数 示例值：" "
AvailableStatus	String	触发器状态 示例值：Available
Qualifier	String	触发器绑定的别名或版本 示例值：\$DEFAULT
Description	String	触发器描述 示例值：this is a timer trigger

TriggerAction

预置定时任务动作

被如下接口引用：GetProvisionedConcurrencyConfig, PutProvisionedConcurrencyConfig。

名称	类型	必选	描述
TriggerName	String	是	定时预置名称 示例值：timer-1
TriggerProvisionedConcurrencyNum	Integer	是	定时预置并发数量 示例值：1
TriggerCronConfig	String	是	设置定时触发器的时间配置，cron表达式。Cron 表达式有七个必需字段，按空格分隔。 示例值：30 42 15 20 12 * 2024
ProvisionedType	String	否	预置类型 Default 注意：此字段可能返回 null，表示取不到有效值。 示例值：Default

TriggerCount

TriggerCount描述不同类型触发器的数量

被如下接口引用：GetAccount。

名称	类型	描述
Cos	Integer	Cos触发器数量 示例值：0
Timer	Integer	Timer触发器数量 示例值：0
Cmq	Integer	Cmq触发器数量 示例值：0

名称	类型	描述
Total	Integer	触发器总数 示例值：0
Ckafka	Integer	Ckafka触发器数量 示例值：0
Apigw	Integer	Apigw触发器数量 示例值：0
Cls	Integer	Cls触发器数量 示例值：0
Clb	Integer	Clb触发器数量 示例值：0
Mps	Integer	Mps触发器数量 示例值：0
Cm	Integer	Cm触发器数量 示例值：0
Vod	Integer	Vod触发器数量 示例值：0
Eb	Integer	Eb触发器数量 示例值：0

TriggerInfo

触发器信息

被如下接口引用：ListTriggers。

名称	类型	描述
Enable	Integer	使能开关 示例值：1
Qualifier	String	函数版本或别名 示例值：\$DEFAULT
TriggerName	String	触发器名称 示例值：hulrzbh9f
Type	String	触发器类型 示例值：http
TriggerDesc	String	触发器详细配置 示例值：{"AuthType":"NONE","NetConfig":{"EnableIntranet":true}}
AvailableStatus	String	触发器是否可用 示例值：Available
CustomArgument	String	客户自定义参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：custom
AddTime	Timestamp	触发器创建时间 示例值：2024-12-02 17:18:47
ModTime	Timestamp	触发器最后修改时间 示例值：2024-12-02 17:18:47
Description	String	客户自定义触发器描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：Some Description
BoundResources	String	与此触发器关联的资源。目前仅函数URL关联的自定义域名会返回 示例值：[{"domain":"test.com","PathMatch":"/"}]

UsageInfo

已使用的信息

被如下接口引用：GetAccount。

名称	类型	描述
NamespacesCount	Integer	命名空间个数 示例值：1
Namespace	Array of NamespaceUsage	命名空间详情 示例值：[{"Namespace":"default","Functions":["functionName1"],"TotalConcurrencyMem":0,"TotalAllocatedConcurrencyMem":(
TotalConcurrencyMem	Integer	当前地域用户并发内存配额上限 示例值：128000
TotalAllocatedConcurrencyMem	Integer	当前地域用户已配置并发内存额度 示例值：0
UserConcurrencyMemLimit	Integer	用户实际配置的账号并发配额 示例值：128000

Variable

变量参数

被如下接口引用：CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
Key	String	否	变量的名称，不可为空字符 示例值：testKey
Value	String	否	变量的值，不可为空字符 示例值：testValue

VersionMatch

带有匹配规则的函数版本

被如下接口引用：CreateAlias, GetAlias, UpdateAlias。

名称	类型	必选	描述
Version	String	是	函数版本名称 示例值：1
Key	String	是	匹配规则的key，调用时通过传key来匹配规则路由到指定版本 header方式： key填写"invoke.headers.User"，并在 invoke 调用函数时传参 RoutingKey: {"User":"value"}规则匹配调用 示例值：invoke.headers.User
Method	String	是	匹配方式。取值范围： range：范围匹配 exact：字符串精确匹配 示例值：exact
Expression	String	是	range 匹配规则要求： 需要为开区间或闭区间描述 (a,b) [a,b]，其中 a、b 均为整数 exact 匹配规则要求： 字符串精确匹配 示例值：testuser

VersionProvisionedConcurrencyInfo

函数版本的预置并发信息，包括设置预置并发数、已完成预置的并发数和预置任务状态。

被如下接口引用：GetProvisionedConcurrencyConfig。

名称	类型	描述
AllocatedProvisionedConcurrencyNum	Integer	设置的预置并发数。 示例值：1
AvailableProvisionedConcurrencyNum	Integer	当前已完成预置的并发数。 示例值：1
Status	String	预置任务状态，Done表示已完成，InProgress表示进行中，Failed表示部分或全部失败。 示例值：Done
StatusReason	String	对预置任务状态Status的说明。 示例值："
Qualifier	String	函数版本号 示例值：1
TriggerActions	Array of TriggerAction	预置并发定时任务。 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"TriggerName":"timer-1","ProvisionedType":"Default","TriggerProvisionedConcurrencyNum":1,"Trigger42 15 20 12 * 2024"}]

VersionWeight

带有权重的函数版本

被如下接口引用：CreateAlias, GetAlias, UpdateAlias。

名称	类型	必选	描述
Version	String	是	函数版本名称 示例值：1
Weight	Float	是	该版本的权重 示例值：0.8

VpcConfig

私有网络参数配置

被如下接口引用：CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
VpcId	String	否	私有网络的 Id 示例值：vpc-xxxxxxxx
SubnetId	String	否	子网的 Id 示例值：subnet-xxxxxxxx

WSParams

HTTP函数通过WebSockets协议访问时的参数

被如下接口引用：CreateFunction, GetFunction, UpdateFunctionConfiguration。

名称	类型	必选	描述
IdleTimeOut	Integer	否	空闲超时时间, 单位秒，默认15s。可配置范围1~1800s。 注意：此字段可能返回 null，表示取不到有效值。 示例值：15

WafConf

web应用防火墙配置信息

被如下接口引用：CreateCustomDomain, GetCustomDomain, ListCustomDomains, UpdateCustomDomain。

名称	类型	必选	描述
WafOpen	String	否	web应用防火墙是否打开， 取值范围:OPEN, CLOSE 示例值： OPEN
WafInstanceId	String	否	web应用防火墙实例ID 示例值： waf_2kzh3vlq0u

错误码

最近更新时间：2025-05-23 01:27:23

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
FailedOperation.AccountInsufficient	账户余额不足。
FailedOperation.ApiGateway	API网关触发器创建失败。
FailedOperation.Apigw	创建触发器失败。
FailedOperation.ApmConfigInstanceId	获取Apm InstanceId失败。
FailedOperation.AsyncEventStatus	当前异步事件状态不支持此操作，请稍后重试。
FailedOperation.AuthFailure	登录信息验证失败，token 验证失败。
FailedOperation.BindPlugin	绑定插件失败，请检查参数。
FailedOperation.CallNetDeployFailed	调用 NetDeploy 失败。
FailedOperation.CallRoleFailed	请求role信息失败。
FailedOperation.ClsServiceUnregistered	CLS服务未注册。
FailedOperation.ClusterNotFound	函数集群不存在。
FailedOperation.CopyAsyncRun	CopyAsyncRun 传参异常。
FailedOperation.CopyFailed	复制函数失败。
FailedOperation.CopyFunction	不支持复制到该地域。
FailedOperation.Cos	操作COS资源失败。
FailedOperation.CreateAlias	创建别名失败。

错误码	说明
FailedOperation.CreateFunction	操作失败。
FailedOperation.CreateNamespace	创建命名空间失败。
FailedOperation.CreateTrigger	当前函数状态无法进行此操作。
FailedOperation.DebugModeStatus	当前调试状态无法执行此操作。
FailedOperation.DebugModeUpdateTimeOutFail	调试状态下无法更新执行超时时间。
FailedOperation.DeleteAlias	删除别名失败。
FailedOperation.DeleteFunction	当前函数状态无法进行此操作，请在函数状态正常时重试。
FailedOperation.DeleteLayerVersion	删除layer版本失败。
FailedOperation.DeleteNamespace	无法删除默认Namespace。
FailedOperation.DeleteTrigger	删除触发器失败。
FailedOperation.DeleteTrigger.UrlUsed	Url被占用，无法删除。
FailedOperation.Domain.UnExist	自定义域名不存在
FailedOperation.FunctionNameStatusError	当前函数状态无法更新代码，请在状态为正常时更新。
FailedOperation.FunctionStatusError	函数在部署中,无法做此操作。
FailedOperation.FunctionVersionStatusNotActive	当前函数版本状态无法进行此操作，请在版本状态为正常时重试。
FailedOperation.GetAlias	获取别名信息失败。
FailedOperation.GetFunctionAddress	获取函数代码地址失败。
FailedOperation.InstanceNotFound	InstanceNotFound 实例不存在。
FailedOperation.InsufficientBalance	当前账号或命名空间处于欠费状态，请在可用时重试。
FailedOperation.InsufficientResources	本地域资源售罄。
FailedOperation.InvokeFunction	调用函数失败。
FailedOperation.Namespace	命名空间已存在，请勿重复创建。
FailedOperation.OpenService	服务开通失败。
FailedOperation.OperationConflict	操作冲突。
FailedOperation.ProvisionCreateTimer	创建定时预置任务失败。
FailedOperation.ProvisionDeleteTimer	删除定时预置任务失败。
FailedOperation.ProvisionedExceedAvailable	预置超过可用。
FailedOperation.ProvisionedExceedReserved	预置超限。
FailedOperation.ProvisionedInProgress	当前函数版本已有预置任务处于进行中，请稍后重试。
FailedOperation.PublishLayerVersion	发布layer版本失败。
FailedOperation.PublishVersion	当前函数状态无法发布版本，请在状态为正常时发布。
FailedOperation.QcsRoleNotFound	角色不存在。
FailedOperation.QueryError	查询失败。
FailedOperation.ReservedExceedTotal	ReservedExceedTotal 总保留超限。
FailedOperation.ReservedInProgress	当前函数已有保留并发设置任务处于进行中，请稍后重试。
FailedOperation.ServiceClosed	ServiceClosed 请确认后再操作。
FailedOperation.Ssl	更新自定义域名失败，SSL异常。

错误码	说明
FailedOperation.TopicNotExist	Topic不存在。
FailedOperation.TotalConcurrencyMemoryInProgress	用户并发内存配额设置任务处于进行中，请稍后重试。
FailedOperation.UnOpenedService	指定的服务未开通，可以提交工单申请开通服务。
FailedOperation.UpdateAlias	更新别名失败。
FailedOperation.UpdateFunctionCode	当前函数状态无法更新代码，请在状态为正常时更新。
FailedOperation.UpdateFunctionConfiguration	UpdateFunctionConfiguration操作失败。
FailedOperation.UpdateStatus	更新状态操作失败。
FailedOperation.UpdateTrigger	更新触发器出错。
FailedOperation.UpdateTriggerStatus	与原状态一致，无需更改。
InternalError.ApiGateway	创建apigw触发器内部错误。
InternalError.Ckafka	ckafka接口失败。
InternalError.Cmq	删除cmq触发器失败。
InternalError.Cos	更新触发器失败。
InternalError.ES	ES错误。
InternalError.Exception	内部服务异常。
InternalError.GetRoleError	内部服务错误。
InternalError.GetStsTokenFailed	获取sts票据信息失败。
InternalError.System	内部系统错误。
InternalError.SystemError	内部服务错误。
InvalidParameter.Cls	日志参数有误。
InvalidParameter.FunctionName	FunctionName取值与规范不符，请修正后再试。
InvalidParameter.FunctionNameOrResourceIdRequired	资源ID和函数名参数不能同时为空。
InvalidParameter.ParamError	创建函数传参异常。
InvalidParameter.Payload	请求参数不合法。
InvalidParameter.RequestTooLarge	请求大小超限。
InvalidParameter.RoleCheck	RoleCheck 传参有误。
InvalidParameter.RoutingConfig	RoutingConfig参数传入错误。
InvalidParameterValue.Action	Action取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17235 。
InvalidParameterValue.AdditionalVersionWeights	AdditionalVersionWeights参数传入错误。
InvalidParameterValue.Alias	不支持删除默认别名，请修正后重试。
InvalidParameterValue.ApiGateway	ApiGateway参数错误。
InvalidParameterValue.ApmConfig	ApmConfig参数传入错误。
InvalidParameterValue.ApmConfigInstanceId	ApmConfigInstanceId参数传入错误。
InvalidParameterValue.ApmConfigRegion	ApmConfigRegion参数传入错误。
InvalidParameterValue.Args	Args 参数值有误。
InvalidParameterValue.ArgsList	容器启动参数取值与规范不符。

错误码	说明
InvalidParameterValue.AsyncRunEnable	AsyncRunEnable 取值不正确。
InvalidParameterValue.AsyncTriggerConfig	函数异步重试配置参数无效。
InvalidParameterValue.Cdn	Cdn传入错误。
InvalidParameterValue.CertConfig.empty	CertConfig配置为空。
InvalidParameterValue.CfsParameterDuplicate	cfs配置项重复。
InvalidParameterValue.CfsParameterError	cfs配置项取值与规范不符。
InvalidParameterValue.CfsStructionError	cfs参数格式与规范不符。
InvalidParameterValue.Ckafka	Ckafka传入错误。
InvalidParameterValue.ClientContext	运行函数时的参数传入有误。
InvalidParameterValue.Cls	Cls传入错误。
InvalidParameterValue.ClsRole	修改Cls配置需要传入Role参数，请修正后重试。
InvalidParameterValue.Cmq	Cmq传入错误。
InvalidParameterValue.Code	Code传入错误。
InvalidParameterValue.CodeSecret	CodeSecret传入错误。
InvalidParameterValue.CodeSource	CodeSource传入错误。
InvalidParameterValue.Command	Command[Entrypoint] 参数值有误。
InvalidParameterValue.CommandList	容器启动命令参数值与规范不符。
InvalidParameterValue.CompatibleRuntimes	CompatibleRuntimes参数传入错误。
InvalidParameterValue.Content	Content参数传入错误。
InvalidParameterValue.Cors	Cors 字段传值异常。
InvalidParameterValue.Cos	Cos传入错误。
InvalidParameterValue.CosBucketName	CosBucketName不符合规范。
InvalidParameterValue.CosBucketRegion	CosBucketRegion取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#Code 。
InvalidParameterValue.CosNotifyRuleConflict	COS通知规则冲突。
InvalidParameterValue.CosObjectName	CosObjectName不符合规范。
InvalidParameterValue.CustomArgument	CustomArgument参数长度超限。
InvalidParameterValue.DateTime	DateTime传入错误。
InvalidParameterValue.DeadLetterConfig	DeadLetterConfig取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#DeadLetterConfi
InvalidParameterValue.DefaultNamespace	默认Namespace无法创建。
InvalidParameterValue.Demo	DemoID 对应的函数模板，code 参数值有误，请确认后重试。
InvalidParameterValue.Demold	Demold 不存在。
InvalidParameterValue.Description	Description传入错误。
InvalidParameterValue.DiskSize	DiskSize 参数取值不规范。
InvalidParameterValue.DnsInfo	环境变量DNS[OS_NAMESERVER]配置有误。
InvalidParameterValue.DynamicEnabled	DynamicEnabled 参数传入错误。

错误码	说明
InvalidParameterValue.EASConfig	EASConfig参数异常。
InvalidParameterValue.EipConfig	EipConfig参数错误。
InvalidParameterValue.Enable	Enable取值与规范不符，请修正后再试。
InvalidParameterValue.Environment	Environment传入错误。
InvalidParameterValue.EnvironmentExceededLimit	环境变量大小超限，请保持在 4KB 以内。
InvalidParameterValue.EnvironmentSystemProtect	不支持修改函数系统环境变量和运行环境变量。
InvalidParameterValue.Filters	Filters参数错误。
InvalidParameterValue.FunctionName	函数不存在。
InvalidParameterValue.FunctionRequestId	请求 id 传参错误。
InvalidParameterValue.FunctionType	FunctionType参数错误。
InvalidParameterValue.GitBranch	GitBranch不符合规范。
InvalidParameterValue.GitCommitId	GitCommitId取值与规范不符，请修正后再试。
InvalidParameterValue.GitDirectory	GitDirectory不符合规范。
InvalidParameterValue.GitPassword	GitPassword不符合规范。
InvalidParameterValue.GitPasswordSecret	GitPasswordSecret 传参有误。
InvalidParameterValue.GitUrl	GitUrl不符合规范。
InvalidParameterValue.GitUserName	GitUserName不符合规范。
InvalidParameterValue.GitUserNameSecret	GitUserNameSecret 传参有误。
InvalidParameterValue.Handler	Handler传入错误。
InvalidParameterValue.IdleTimeOut	IdleTimeOut参数传入错误。
InvalidParameterValue.ImageType	ImageType 参数值有误。
InvalidParameterValue.ImageUri	imageUri 传入有误。
InvalidParameterValue.InlineZipFile	InlineZipFile非法。
InvalidParameterValue.InstanceConcurrencyConfig	InstanceConcurrencyConfig 参数传入错误。
InvalidParameterValue.InvokeType	InvokeType取值与规范不符，请修正后再试。
InvalidParameterValue.L5Enable	L5Enable取值与规范不符，请修正后再试。
InvalidParameterValue.LayerName	LayerName参数传入错误。
InvalidParameterValue.Layers	Layers参数传入错误。
InvalidParameterValue.Limit	Limit传入错误。
InvalidParameterValue.LimitExceeded	参数超出长度限制。
InvalidParameterValue.LogFormat	日志格式不正确。
InvalidParameterValue.MaxConcurrency	MaxConcurrency 参数传入错误。
InvalidParameterValue.Memory	Memory 取值与规范不符，函数运行时内存大小默认为128MB，可选范围64MB、128MB – 3072MB，以128MB为阶梯，请修正后重试。
InvalidParameterValue.MemorySize	MemorySize错误。
InvalidParameterValue.MinCapacity	MinCapacity 参数传入错误。
InvalidParameterValue.Name	Name参数传入错误。

错误码	说明
InvalidParameterValue.Namespace	Namespace参数传入错误。
InvalidParameterValue.NamespaceInvalid	规则不正确，Namespace为英文字母、数字、_ 符号组成，长度30。
InvalidParameterValue.NodeSpec	NodeSpec 参数传入错误。
InvalidParameterValue.NodeType	NodeType 参数传入错误。
InvalidParameterValue.Offset	偏移量不合法。
InvalidParameterValue.Order	Order传入错误。
InvalidParameterValue.OrderBy	OrderBy取值与规范不符，请修正后再试。可参考对应接口文档中的示例值： https://cloud.tencent.com/document/product/583/17235
InvalidParameterValue.Orderby	Orderby传入错误。
InvalidParameterValue.Param	入参不是标准的json。
InvalidParameterValue.ProtocolType	ProtocolType参数传入错误。
InvalidParameterValue.ProvisionTriggerCronConfigDuplicate	定时预置的cron配置重复。
InvalidParameterValue.ProvisionTriggerName	TriggerName参数传入错误。
InvalidParameterValue.ProvisionTriggerNameDuplicate	TriggerName重复。
InvalidParameterValue.ProvisionType	ProvisionType 参数传入错误。
InvalidParameterValue.PublicNetConfig	PublicNetConfig参数错误。
InvalidParameterValue.Qualifier	不支持的函数版本。
InvalidParameterValue.QueryVersion	查询版本详情，版本参数传入错误。
InvalidParameterValue.RegistryId	企业版镜像实例ID[RegistryId]传值错误。
InvalidParameterValue.RetCode	RetCode不合法。
InvalidParameterValue.RoutingConfig	RoutingConfig取值与规范不符，请修正后再试。可参考： https://cloud.tencent.com/document/product/583/17244#RoutingConfig 。
InvalidParameterValue.Runtime	Runtime传入错误。
InvalidParameterValue.SearchKey	searchkey 不是 Keyword,Tag 或者 Runtime。
InvalidParameterValue.SecretInfo	SecretInfo错误。
InvalidParameterValue.ServiceName	ServiceName命名不规范。
InvalidParameterValue.Stamp	Stamp取值与规范不符，请修正后再试。
InvalidParameterValue.StartTime	起始时间传入错误。
InvalidParameterValue.StartTimeOrEndTime	需要同时指定开始日期与结束日期。
InvalidParameterValue.Status	Status取值与规范不符，请修正后再试。
InvalidParameterValue.SystemEnvironment	系统环境变量错误。
InvalidParameterValue.TempCosObjectName	非法的TempCosObjectName。
InvalidParameterValue.TraceEnable	TraceEnable取值与规范不符，请修正后再试。
InvalidParameterValue.TrackingTarget	TrackingTarget 参数输入错误。
InvalidParameterValue.TriggerCronConfig	TriggerCronConfig参数传入错误。
InvalidParameterValue.TriggerCronConfigTimeInterval	TriggerCronConfig参数定时触发间隔小于指定值。
InvalidParameterValue.TriggerDesc	TriggerDesc传入参数错误。

错误码	说明
InvalidParameterValue.TriggerName	TriggerName传入错误。
InvalidParameterValue.TriggerProvisionedConcurrencyNum	TriggerProvisionedConcurrencyNum参数传入错误。
InvalidParameterValue.Type	Type传入错误。
InvalidParameterValue.Vpc	VPC不正确。
InvalidParameterValue.VpcNotSetWhenOpenCfs	开启cfs配置的同时必须开启vpc。
InvalidParameterValue.WebSocketsParams	WebSocketsParams参数传入错误。
InvalidParameterValue.ZipFile	检测到不是标准的zip文件，请重新压缩后再试。
InvalidParameterValue.ZipFileBase64BinnasciiError	压缩文件base64解码失败: <code>Incorrect padding</code> ，请修正后再试。
LimitExceeded.Alias	别名个数超过最大限制。
LimitExceeded.Cdn	Cdn使用超过最大限制。
LimitExceeded.ContainerImageAccelerate	用户开启镜像加速函数版本超限。
LimitExceeded.ContainerImageAccelerateQuota	用户开启镜像加速函数版本超限。
LimitExceeded.Eip	eip资源超限。
LimitExceeded.Function	函数数量超出最大限制，可通过 提交工单 申请提升限制。
LimitExceeded.FunctionOnTopic	同一个主题下的函数超过最大限制。
LimitExceeded.FunctionProvisionedConcurrencyMemory	FunctionProvisionedConcurrencyMemory数量达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
LimitExceeded.FunctionReservedConcurrencyMemory	函数保留并发内存超限。
LimitExceeded.FunctionTotalProvisionedConcurrencyMemory	FunctionTotalProvisionedConcurrencyMemory达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
LimitExceeded.FunctionTotalProvisionedConcurrencyNum	函数预置并发总数达到限制。
LimitExceeded.InitTimeout	InitTimeout达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
LimitExceeded.Intrapl	内网固定IP个数超限。
LimitExceeded.LayerVersions	layer版本数量超出最大限制。
LimitExceeded.Layers	layer数量超出最大限制。
LimitExceeded.MaxCapacity	动态扩容最大值超限。
LimitExceeded.Memory	内存超出最大限制。
LimitExceeded.MsgTTL	函数异步重试配置消息保留时间超过限制。
LimitExceeded.Namespace	命名空间数量超过最大限制，可通过 提交工单 申请提升限制。
LimitExceeded.Offset	Offset超出限制。
LimitExceeded.PrivilegeContainer	特权容器属性开启失败
LimitExceeded.ProvisionTriggerAction	定时预置数量超过最大限制。
LimitExceeded.ProvisionTriggerInterval	定时触发间隔小于最大限制。
LimitExceeded.Quota	配额超限。
LimitExceeded.RetryNum	函数异步重试配置异步重试次数超过限制。
LimitExceeded.Timeout	Timeout超出最大限制。
LimitExceeded.TotalConcurrencyMemory	用户并发内存配额超限。

错误码	说明
LimitExceeded.Trigger	触发器数量超出最大限制，可通过 提交工单 申请提升限制。
LimitExceeded.UserTotalConcurrencyMemory	UserTotalConcurrencyMemory达到限制，可提交工单申请提升限制： https://tencentcs.com/7Fixwt63 。
MissingParameter.Code	Code没有传入。
MissingParameter.Runtime	缺失 Runtime 字段。
OperationDenied.AccountIsolate	账号已被隔离。
OperationDenied.AccountNotExists	账号不存在。
ResourceInUse.Alias	Alias已被占用。
ResourceInUse.Cdn	Cdn已被占用。
ResourceInUse.Cmq	Cmq已被占用。
ResourceInUse.Cos	Cos已被占用。
ResourceInUse.Function	函数已存在。
ResourceInUse.FunctionName	FunctionName已存在。
ResourceInUse.LayerVersion	Layer版本正在使用中。
ResourceInUse.Namespace	Namespace已存在。
ResourceInUse.Trigger	TriggerName已存在。
ResourceInUse.TriggerName	TriggerName已存在。
ResourceInsufficient.COS	COS资源不足。
ResourceNotFound.Alias	别名不存在。
ResourceNotFound.AsyncEvent	未找到指定的AsyncEvent，请创建后再试。
ResourceNotFound.AttachedTagKeyNotFound	函数需要关联的标签Key不存在。
ResourceNotFound.Cdn	Cdn不存在。
ResourceNotFound.CfsMountInsNotMatch	指定的cfs下未找到您所指定的挂载点。
ResourceNotFound.CfsProtocolError	CfsProtocolError 参数异常。
ResourceNotFound.CfsStatusError	检测cfs状态为不可用。
ResourceNotFound.CfsVpcNotMatch	cfs与云函数所处vpc不一致。
ResourceNotFound.Ckafka	Ckafka不存在。
ResourceNotFound.Cmq	Cmq不存在。
ResourceNotFound.Cos	Cos不存在。
ResourceNotFound.Demo	不存在的Demo。
ResourceNotFound.Function	函数不存在。
ResourceNotFound.FunctionName	函数不存在。
ResourceNotFound.FunctionVersion	函数版本不存在。
ResourceNotFound.GetCfsMountInsError	获取cfs挂载点信息错误。
ResourceNotFound.GetCfsNotMatch	获取cfs信息错误。
ResourceNotFound.ImageConfig	未找到指定的ImageConfig，请创建后再试。
ResourceNotFound.Layer	layer不存在。

错误码	说明
ResourceNotFound.LayerVersion	Layer版本不存在。
ResourceNotFound.Namespace	Namespace不存在。
ResourceNotFound.Qualifier	版本不存在。
ResourceNotFound.Role	角色不存在。
ResourceNotFound.RoleCheck	Role不存在。
ResourceNotFound.Timer	Timer不存在。
ResourceNotFound.TotalConcurrencyMemory	并发内存配额资源未找到。
ResourceNotFound.Trigger	触发器不存在。
ResourceNotFound.Version	版本不存在。
ResourceNotFound.Vpc	VPC或子网不存在。
ResourceUnavailable.InsufficientBalance	余额不足，请先充值。
ResourceUnavailable.Namespace	Namespace不可用。
ResourceUnavailable.ServiceSuspended	服务已冻结。
UnauthorizedOperation.CAM	CAM鉴权失败。
UnauthorizedOperation.CodeSecret	无访问代码权限。
UnauthorizedOperation.CreateTrigger	没有权限。
UnauthorizedOperation.DeleteFunction	没有权限的操作。
UnauthorizedOperation.DeleteTrigger	没有权限。
UnauthorizedOperation.NotMC	不是从控制台调用的该接口。
UnauthorizedOperation.Region	Region错误。
UnauthorizedOperation.Role	没有权限访问您的Cos资源。
UnauthorizedOperation.TempCosAppid	TempCos的Appid和请求账户的APPID不一致。
UnauthorizedOperation.UpdateFunctionCode	无法进行此操作。
UnauthorizedOperation.UpdateTriggerStatus	没有权限。
UnsupportedOperation.AliasBind	资源还有别名绑定，不支持当前操作，请解绑别名后重试。
UnsupportedOperation.AsyncRunEnable	指定的配置AsyncRunEnable暂不支持，请修正后再试。
UnsupportedOperation.Cdn	Cdn不支持。
UnsupportedOperation.CopyFunction	该函数无法复制。
UnsupportedOperation.Cos	Cos操作不支持。
UnsupportedOperation.DiskSize	DiskSize 配置内测中，需要申请开启。如需使用可联系我们。
UnsupportedOperation.EipFixed	指定的配置EipFixed暂不支持。
UnsupportedOperation.NotSupportRegion	不支持的地域。
UnsupportedOperation.PublishVersion	当前状态无法发布版本。
UnsupportedOperation.Region	不支持此地域。
UnsupportedOperation.Trigger	Trigger操作不支持。
UnsupportedOperation.UpdateFunctionCode	更新函数代码失败。
UnsupportedOperation.UpdateFunctionEventInvokeConfig	指定的配置暂不支持，请修正后再试。

错误码	说明
UnsupportedOperation.VpcConfig	指定的配置VpcConfig暂不支持。

附录

函数和层的状态说明

最近更新时间：2024-12-30 01:54:43

当使用 [云 API](#) 管理函数（增、删、改、查）时，由于接口异步，首先需要查询函数当前状态，然后进行下一步操作。

函数状态

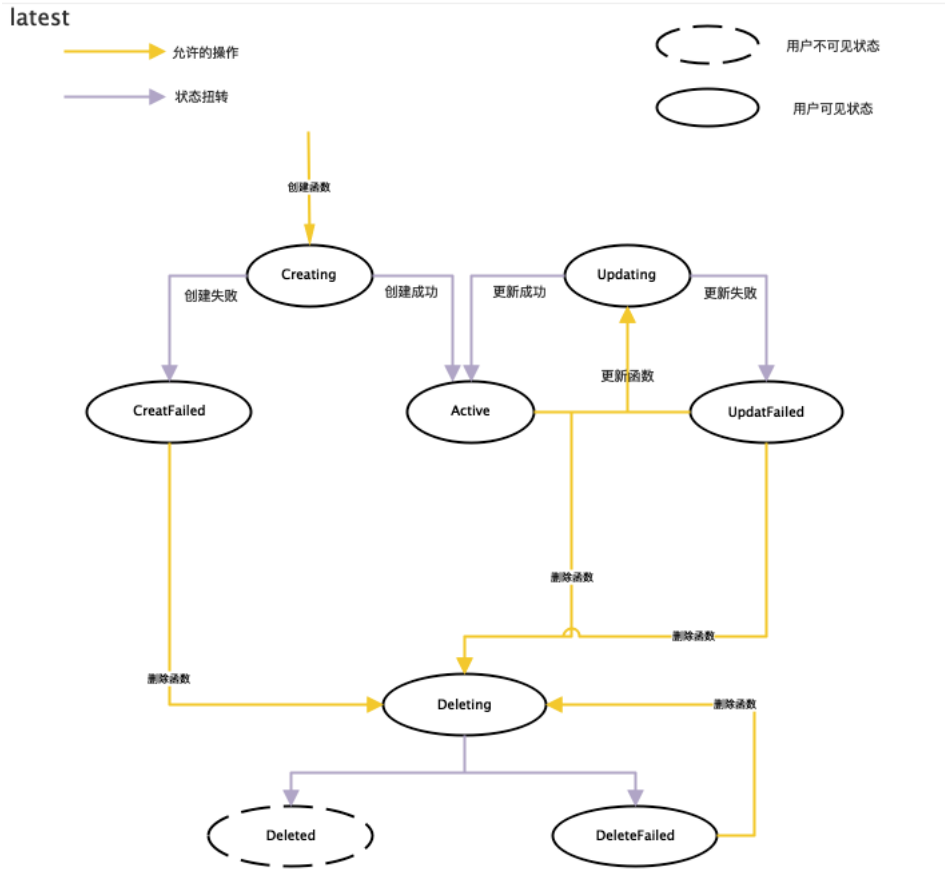
创建、更新、删除、发布函数时，函数存在以下状态：

Status（接口返回 Status 值）	说明
Creating	函数创建中
CreateFailed	函数创建失败（如果已生成函数信息，则可以删除后再创建）
Active	函数可用
Updating	函数更新中
UpdateFailed	函数更新失败
Publishing	函数版本发布中
PublishFailed	函数版本发布失败
Deleting	函数删除中
DeleteFailed	函数删除失败

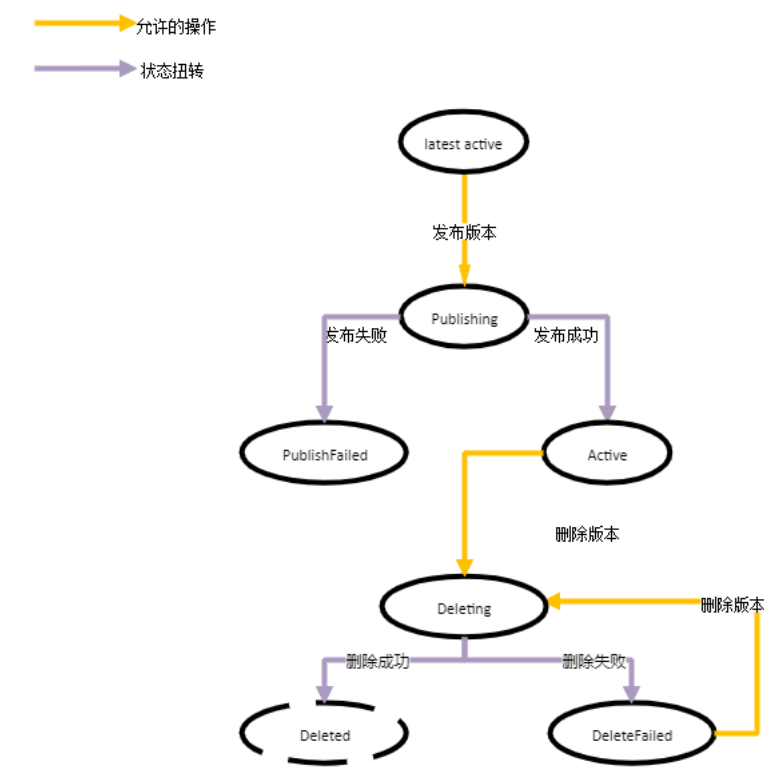
注意：

函数发布、更新失败不会影响已正常发布的函数版本的使用，调用函数时会正常调用更新之前的函数内容。

函数创建及更新的状态扭转。如下图所示：



函数发版的状态扭转。如下图所示：



函数计费状态

AvailableStatus（接口返回 AvailableStatus 值）	说明
Available	可用状态
InsufficientBalance	因欠费导致函数不可用

层（Layer）状态

Status（接口返回 Status 值）	说明
Publishing	层发布中
PublishFailed	层发布失败
Active	层可用
Deleting	层删除中
DeleteFailed	层删除失败
Deleted	层已删除