

弹性微服务 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

产品版本

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

环境相关接口

创建环境

获取环境基础信息

获取环境状态

获取环境列表

销毁环境

编辑环境

网关相关接口

查询 Ingress 规则列表

查询应用关联的 Ingress 规则列表

删除 Ingress 规则

查询 Ingress 规则

创建或者更新 Ingress 规则

资源相关接口

绑定云资源

应用相关接口

创建应用

生成应用程序包预签名下载链接

修改应用基本信息

服务相关接口

获取运行服务列表

获取分批发布详情

开始下一批次发布

回滚分批发布

工作负载相关接口

应用部署

服务基本信息查看

获取应用实例列表

重启应用实例

更新应用部署版本

访问方式相关接口

新增访问方式

删除一条访问方式

查询应用访问方式列表

修改服务访问方式列表

配置文件相关接口

创建配置

查询配置详情

查询配置列表

销毁配置

编辑配置

日志采集相关接口

创建日志收集配置

- 查询日志收集配置详情
- 查询分页的日志收集配置列表
- 销毁日志收集配置
- 编辑日志收集配置

弹性规则相关接口

- 创建弹性伸缩策略组合
- 删除应用弹性策略组合
- 获取应用弹性策略组合
- 关闭应用弹性策略组合
- 启用应用弹性策略组合
- 修改弹性伸缩策略组合

服务版本相关接口

- 服务删除
- 单环境下所有应用状态查看
- 修改应用实例数量
- 服务重启
- 服务停止

其他相关接口

- 生成Cos临时密钥

数据结构

错误码

API 文档

产品版本

最近更新时间：2021-10-21 11:38:51

您正在浏览tem产品文档的版本: 2021-07-01

最新版本

- [2021-07-01](#) (当前版本)

以往版本

- [2020-12-21](#)

更新历史

最近更新时间：2025-04-24 01:27:22

第 48 次发布

发布时间：2025-04-24 01:27:15

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ServiceVersionBrief](#)
 - 修改成员： VersionName, Status, EnableEs, CurrentInstances, VersionId, LogOutputConf, ExpectedInstances, DeployMode, BuildTaskId, EnvironmentId, EnvironmentName, ApplicationId, ApplicationName, UnderDeploying, BatchDeployStatus, Zones, NodeInfos, PodList, WorkloadInfo, CreateDate
- [TemService](#)
 - 修改成员： ApplicationId, ApplicationName, Description, EnvironmentId, CreateDate, ModifyDate, Modifier, Creator, RepoType, InstanceId, RepoName, CodingLanguage, DeployMode, EnvironmentName, ActiveVersions, EnableTracing, Tags, HasAuthority

第 47 次发布

发布时间：2024-12-30 02:00:58

本次发布包含了以下内容：

改善已有的文档。

预下线接口：

- [ModifyIngress](#)

修改数据结构：

- [DescribeRunPodPage](#)
 - 修改成员： Offset, Limit, TotalCount, RequestId, PodList
- [NodeInfo](#)
 - 修改成员： Name, Zone, SubnetId, AvailableIpCount, Cidr
- [RunVersionPod](#)
 - 修改成员： Webshell, PodId, Status, CreateTime, PodIp, Zone, DeployVersion, RestartCount, Ready, ContainerState, NodeInfo, StartTime, Unhealthy, UnhealthyWarningMsg, VersionId, ApplicationName
- [WorkloadInfo](#)
 - 修改成员： ClusterId, ApplicationName, VersionName, ReadyReplicas, Replicas, UpdatedReplicas, UpdatedReadyReplicas, UpdateRevision, CurrentRevision

第 46 次发布

发布时间：2024-11-01 02:08:02

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 修改入参： DeployVersion

修改数据结构：

- [IngressInfo](#)
 - 修改成员： ClusterNamespace

第 45 次发布

发布时间：2024-10-31 02:13:44

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateApplicationAutoscaler](#)
 - 修改入参：Autoscaler
- [CreateApplicationService](#)
 - 修改入参：Service
- [CreateConfigData](#)
 - 修改入参：Data
- [CreateLogConfig](#)
 - 修改入参：ApplicationId
- [DeleteApplicationAutoscaler](#)
 - 修改入参：AutoscalerId
- [DeleteApplicationService](#)
 - 修改入参：ApplicationId, EnvironmentId, ServiceName
- [DescribeApplicationsStatus](#)
 - 修改入参：EnvironmentId
- [DescribeDeployApplicationDetail](#)
 - 修改入参：ApplicationId, EnvironmentId
- [DescribeIngresses](#)
 - 修改入参：EnvironmentId, ClusterNamespace
- [DescribeLogConfig](#)
 - 修改入参：ApplicationId
- [DescribeRelatedIngresses](#)
 - 修改入参：EnvironmentId, ClusterNamespace, ApplicationId
- [DisableApplicationAutoscaler](#)
 - 修改入参：AutoscalerId
- [EnableApplicationAutoscaler](#)
 - 修改入参：AutoscalerId
- [ModifyApplicationAutoscaler](#)
 - 修改入参：AutoscalerId, Autoscaler
- [ModifyConfigData](#)
 - 修改入参：Data
- [ModifyLogConfig](#)
 - 修改入参：Data, ApplicationId
- [RestartApplication](#)
 - 修改入参：EnvironmentId
- [ResumeDeployApplication](#)
 - 修改入参：ApplicationId, EnvironmentId
- [RevertDeployApplication](#)
 - 修改入参：ApplicationId, EnvironmentId
- [StopApplication](#)
 - 修改入参：EnvironmentId

修改数据结构：

- [IngressInfo](#)
 - 修改成员：CId

第 44 次发布

发布时间：2024-10-30 02:08:10

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [LogConfig](#)
 - 修改成员：Name, InputType, LogsetId, TopicId, LogType, BeginningRegex, LogPath, FilePattern, CreateDate, ModifyDate, ApplicationId, ApplicationName, ExtractRule

第 43 次发布

发布时间：2024-07-24 02:14:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 新增入参：PostStartEncoded, PreStopEncoded

第 42 次发布

发布时间：2024-06-25 01:23:38

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [UseDefaultRepoParameters](#)
 - 修改成员：EnterpriseInstanceName, EnterpriseInstanceChargeType, EnterpriseInstanceType

第 41 次发布

发布时间：2023-04-20 01:41:18

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [RollingUpdateApplicationByVersion](#)
 - 新增入参：Force

第 40 次发布

发布时间：2023-02-24 01:23:28

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [EksService](#)
 - 新增成员：ExternalDomain
- [IngressInfo](#)
 - 新增成员：Domain
- [ServicePortMapping](#)
 - 新增成员：ExternalDomain

第 39 次发布

发布时间：2023-02-16 01:58:58

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [LogConfigExtractRule](#)
 - 新增成员：Backtracking, Delimiter
- [ServiceVersionBrief](#)
 - 新增成员：RegionId
- [TemEnvironmentStartingStatus](#)
 - 新增成员：StartFailedApplicationNum
- [TemEnvironmentStoppingStatus](#)
 - 新增成员：StopFailedApplicationNum
- [TemServiceVersionInfo](#)
 - 新增成员：PreStopEncoded, PostStartEncoded

第 38 次发布

发布时间：2023-02-07 02:03:34

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateEnvironment](#)
 - 新增入参：SetupVpc, SetupPrometheus, PrometheusId, ApmlId
 - 修改入参：Vpc, SubnetIds
- [CreateResource](#)
 - 修改入参：ResourceId

第 37 次发布

发布时间：2022-12-15 01:49:45

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeEnvironments](#)
 - 新增入参：EnvironmentId

第 36 次发布

发布时间：2022-12-09 02:04:15

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateEnvironment](#)
 - 新增入参：EnvType, CreateRegion
- [DescribeApplications](#)
 - 新增入参：SortInfo
- [DescribeEnvironments](#)
 - 新增入参：SortInfo
- [ModifyEnvironment](#)
 - 新增入参：EnvType

新增数据结构：

- [SortType](#)

修改数据结构：

- [NamespaceInfo](#)
 - 新增成员：EnvType
- [TemNamespaceInfo](#)
 - 新增成员：EnvType, RegionId

第 35 次发布

发布时间：2022-11-17 06:48:59

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [HorizontalAutoscaler](#)
 - 新增成员：DoubleThreshold

第 34 次发布

发布时间：2022-11-08 06:44:33

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [TemNamespaceInfo](#)
 - 新增成员：HasAuthority
- [TemService](#)
 - 新增成员：HasAuthority

第 33 次发布

发布时间：2022-11-02 06:44:20

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateLogConfig](#)
 - 新增入参：ExtractRule

新增数据结构：

- [LogConfigExtractRule](#)

修改数据结构：

- [LogConfig](#)
 - 新增成员：ExtractRule

第 32 次发布

发布时间：2022-10-25 07:01:06

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateApplication](#)
 - 新增入参：Tags
- [CreateEnvironment](#)

- 新增入参: Tags
- [DescribeApplications](#)
 - 新增入参: Filters
- [DescribeEnvironments](#)
 - 新增入参: Filters
- [ModifyApplicationReplicas](#)
 - 新增出参: Result

新增数据结构:

- [QueryFilter](#)

修改数据结构:

- [DeployServicePodDetail](#)
 - 新增成员: Status
- [NamespaceInfo](#)
 - 新增成员: Tags
- [RunVersionPod](#)
 - 新增成员: NodeInfo, StartTime, Unhealthy, UnhealthyWarningMsg, VersionId, ApplicationName
- [ServicePage](#)
 - 新增成员: Current
- [ServiceVersionBrief](#)
 - 新增成员: BatchDeployStatus, Zones, NodeInfos, PodList, WorkloadInfo, CreateDate
- [TemService](#)
 - 新增成员: Tags
- [TemServiceVersionInfo](#)
 - 新增成员: AppId, SubAccountUin, Uin, Region, GroupId, EnableRegistry, AutoscalerList, Modifier, Creator, DeployStrategyConf, PodList, ConfEdited, Tags
- [WorkloadInfo](#)
 - 新增成员: VersionName, ReadyReplicas, Replicas, UpdatedReplicas, UpdatedReadyReplicas, UpdateRevision, CurrentRevision

第 31 次发布

发布时间: 2022-09-28 06:54:52

本次发布包含了以下内容:

改善已有的文档。

新增数据结构:

- [Tag](#)

修改数据结构:

- [NamespacePage](#)
 - 新增成员: Current
- [TemNamespaceInfo](#)
 - 新增成员: AppId, Uin, SubAccountUin, ClusterId, Tags

第 30 次发布

发布时间: 2022-09-27 06:45:31

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [CreateApplicationService](#)
- [DeleteApplicationService](#)
- [DescribeApplicationServiceList](#)

- [ModifyApplicationService](#)

第 29 次发布

发布时间：2022-09-07 06:51:54

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 新增入参：TcrInstanceId, RepoServer, RepoType

第 28 次发布

发布时间：2022-08-12 11:47:08

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 新增入参：EnableMetrics

第 27 次发布

发布时间：2022-08-11 15:11:54

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DisableApplicationAutoscaler](#)
- [EnableApplicationAutoscaler](#)

第 26 次发布

发布时间：2022-08-11 07:11:15

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [TemServiceVersionInfo](#)
 - 新增成员：EnableMetrics
 - 修改成员：ApplicationId, MinAliveInstances

第 25 次发布

发布时间：2022-08-10 16:02:58

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeEnvironment](#)

新增数据结构：

- [NamespaceInfo](#)

第 24 次发布

发布时间：2022-08-04 06:55:33

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateLogConfig](#)
- [DescribeLogConfig](#)
- [DescribePagedLogConfigList](#)
- [DestroyLogConfig](#)
- [ModifyLogConfig](#)

新增数据结构：

- [LogConfig](#)
- [LogConfigListPage](#)
- [ServicePortMapping](#)
- [ServicePortMappingItem](#)

修改数据结构：

- [EksService](#)
 - 新增成员：ServicePortMappingList, FlushAll, EnableRegistryNextDeploy, ApplicationId, AllIpDone

第 23 次发布

发布时间：2022-08-03 06:46:27

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateApplicationAutoscaler](#)
- [CreateConfigData](#)
- [DeleteApplicationAutoscaler](#)
- [DescribeApplicationAutoscalerList](#)
- [DescribeConfigData](#)
- [DescribeConfigDataList](#)
- [DestroyConfigData](#)
- [ModifyApplicationAutoscaler](#)
- [ModifyConfigData](#)

新增数据结构：

- [Autoscaler](#)
- [ConfigData](#)
- [DescribeConfigDataListPage](#)

第 22 次发布

发布时间：2022-08-02 06:33:46

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeApplicationInfo](#)

新增数据结构：

- [NodeInfo](#)
- [TemServiceVersionInfo](#)
- [WorkloadInfo](#)

第 21 次发布

发布时间：2022-08-01 11:54:48

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeApplications](#)
- [DescribeEnvironmentStatus](#)

新增数据结构：

- [NamespaceStatusInfo](#)
- [ServicePage](#)
- [TemEnvironmentStartingStatus](#)
- [TemEnvironmentStoppingStatus](#)
- [TemService](#)

第 20 次发布

发布时间：2022-07-22 06:14:55

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DestroyEnvironment](#)

第 19 次发布

发布时间：2022-07-11 06:18:55

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [PortMapping](#)
 - 新增成员：ServiceName

第 18 次发布

发布时间：2022-06-20 06:14:36

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeApplicationsStatus](#)

新增数据结构：

- [ServiceVersionBrief](#)

第 17 次发布

发布时间：2022-06-09 06:18:14

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 新增入参：EnableTracing

第 16 次发布

发布时间：2022-06-08 06:16:15

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [CronHorizontalAutoscaler](#)
 - 修改成员：Name, Period, Schedules, Enabled
- [HorizontalAutoscaler](#)
 - 新增成员：Enabled
 - 修改成员：MinReplicas, MaxReplicas, Metrics, Threshold

第 15 次发布

发布时间：2022-04-28 06:18:38

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateApplication](#)
 - 新增入参：UseDefaultImageServiceParameters

新增数据结构：

- [UseDefaultRepoParameters](#)

第 14 次发布

发布时间：2022-03-30 06:12:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateResource](#)
 - 新增入参：ResourceFrom, ResourceConfig
- [DeployApplication](#)
 - 新增入参：EnablePrometheusConf

新增数据结构：

- [EnablePrometheusConf](#)

修改数据结构：

- [IngressInfo](#)
 - 新增成员：RewriteType
- [MountedSettingConf](#)
 - 新增成员：SecretDataName
- [Pair](#)

- 新增成员：Secret

第 13 次发布

发布时间：2022-03-09 08:17:24

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 新增入参：OsFlavour

第 12 次发布

发布时间：2022-03-01 08:13:25

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DeployStrategyConf](#)
 - 新增成员：Force

第 11 次发布

发布时间：2022-01-21 08:18:01

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [RollingUpdateApplicationByVersion](#)
 - 新增入参：DeployStrategyType, TotalBatchCount, BatchInterval, BetaBatchNum, MinAvailable

修改数据结构：

- [TemNamespaceInfo](#)
 - 新增成员：Locked

第 10 次发布

发布时间：2022-01-12 08:15:47

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 新增入参：ConfEdited, SpeedUp, StartupProbe

修改数据结构：

- [DeployStrategyConf](#)
 - 新增成员：MinAvailable
- [Pair](#)
 - 新增成员：Type, Config

第 9 次发布

发布时间：2021-12-23 08:16:03

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [RestartApplication](#)
- [StopApplication](#)

第 8 次发布

发布时间：2021-12-22 08:17:21

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DeleteApplication](#)

第 7 次发布

发布时间：2021-12-21 08:11:22

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [RollingUpdateApplicationByVersion](#)

第 6 次发布

发布时间：2021-12-09 08:11:43

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ModifyApplicationReplicas](#)

第 5 次发布

发布时间：2021-10-29 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：

- [DescribeDeployApplicationDetail](#)
 - 新增入参：VersionId

第 4 次发布

发布时间：2021-08-24 08:00:55

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DeployApplication](#)
 - 新增入参：LogEnable

第 3 次发布

发布时间：2021-08-17 08:00:44

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DeployServiceBatchDetail](#)
 - 新增成员：NextBatchStartTime
- [TemDeployApplicationDetailInfo](#)
 - 新增成员：NewDeployVersion, OldDeployVersion, NewVersionPackageInfo, NextBatchStartTime

第 2 次发布

发布时间：2021-08-12 08:00:39

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateApplication](#)
- [CreateCosToken](#)
- [CreateEnvironment](#)
- [CreateResource](#)
- [DeleteIngress](#)
- [DeployApplication](#)
- [DescribeApplicationPods](#)
- [DescribeEnvironments](#)
- [DescribeIngress](#)
- [DescribeIngresses](#)
- [DescribeRelatedIngresses](#)
- [GenerateApplicationPackageDownloadUrl](#)
- [ModifyApplicationInfo](#)
- [ModifyEnvironment](#)
- [ModifyIngress](#)
- [RestartApplicationPod](#)

新增数据结构：

- [CosToken](#)
- [CronHorizontalAutoscaler](#)
- [CronHorizontalAutoscalerSchedule](#)
- [EksService](#)
- [EsInfo](#)
- [HealthCheckConfig](#)
- [HorizontalAutoscaler](#)
- [IngressInfo](#)
- [IngressRule](#)
- [IngressRuleBackend](#)
- [IngressRulePath](#)
- [IngressRuleValue](#)
- [IngressTls](#)
- [LogOutputConf](#)
- [MountedSettingConf](#)
- [NamespacePage](#)
- [Pair](#)
- [PortMapping](#)
- [StorageConf](#)

- [StorageMountConf](#)
- [TemNamespaceInfo](#)

修改数据结构：

- [DeployServiceBatchDetail](#)
 - 新增成员：OldPods, NewPods
- [DeployServicePodDetail](#)
 - 新增成员：Webshell
- [RunVersionPod](#)
 - 新增成员：Ready, ContainerState

第 1 次发布

发布时间：2021-08-04 08:00:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeDeployApplicationDetail](#)
- [ResumeDeployApplication](#)
- [RevertDeployApplication](#)

新增数据结构：

- [DeployServiceBatchDetail](#)
- [DeployServicePodDetail](#)
- [DeployStrategyConf](#)
- [DescribeRunPodPage](#)
- [RunVersionPod](#)
- [TemDeployApplicationDetailInfo](#)

简介

最近更新时间：2025-04-27 21:16:39

腾讯云弹性微服务（TEM）是面向微服务应用的Serverless Paas平台，为用户提供应用托管、生命周期管理、服务治理及多维度监控等微服务管理能力。实现IaaS资源serverless化，微服务自动弹性扩缩容，帮助用户免运维，解决成本和效率问题，进一步降低微服务应用上云的门槛。

API 概览

最近更新时间：2024-08-02 02:13:15

环境相关接口

接口名称	接口功能	频率限制（次/秒）
CreateEnvironment	创建环境	20
DescribeEnvironment	获取环境基础信息	20
DescribeEnvironmentStatus	获取环境状态	20
DescribeEnvironments	获取环境列表	20
DestroyEnvironment	销毁环境	20
ModifyEnvironment	编辑环境	20

网关相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeIngresses	查询 Ingress 规则列表	20
DescribeRelatedIngresses	查询应用关联的 Ingress 规则列表	20
DeleteIngress	删除 Ingress 规则	20
DescribeIngress	查询 Ingress 规则	20
ModifyIngress	创建或者更新 Ingress 规则	20

资源相关接口

接口名称	接口功能	频率限制（次/秒）
CreateResource	绑定云资源	20

应用相关接口

接口名称	接口功能	频率限制（次/秒）
CreateApplication	创建应用	20
GenerateApplicationPackageDownloadUrl	生成应用程序包预签名下载链接	20
ModifyApplicationInfo	修改应用基本信息	20

服务相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeApplications	获取运行服务列表	20
DescribeDeployApplicationDetail	获取分批发布详情	20
ResumeDeployApplication	开始下一批次发布	20
RevertDeployApplication	回滚分批发布	20

工作负载相关接口

接口名称	接口功能	频率限制（次/秒）
DeployApplication	应用部署	20
DescribeApplicationInfo	服务基本信息查看	20

接口名称	接口功能	频率限制（次/秒）
DescribeApplicationPods	获取应用实例列表	20
RestartApplicationPod	重启应用实例	20
RollingUpdateApplicationByVersion	更新应用部署版本	20

访问方式相关接口

接口名称	接口功能	频率限制（次/秒）
CreateApplicationService	新增访问方式	20
DeleteApplicationService	删除一条访问方式	20
DescribeApplicationServiceList	查询应用访问方式列表	20
ModifyApplicationService	修改服务访问方式列表	20

配置文件相关接口

接口名称	接口功能	频率限制（次/秒）
CreateConfigData	创建配置	20
DescribeConfigData	查询配置详情	20
DescribeConfigDataList	查询配置列表	20
DestroyConfigData	销毁配置	20
ModifyConfigData	编辑配置	20

日志采集相关接口

接口名称	接口功能	频率限制（次/秒）
CreateLogConfig	创建日志收集配置	20
DescribeLogConfig	查询日志收集配置详情	20
DescribePagedLogConfigList	查询分页的日志收集配置列表	20
DestroyLogConfig	销毁日志收集配置	20
ModifyLogConfig	编辑日志收集配置	20

弹性规则相关接口

接口名称	接口功能	频率限制（次/秒）
CreateApplicationAutoscaler	创建弹性伸缩策略组合	20
DeleteApplicationAutoscaler	删除应用弹性策略组合	20
DescribeApplicationAutoscalerList	获取应用弹性策略组合	20
DisableApplicationAutoscaler	关闭应用弹性策略组合	20
EnableApplicationAutoscaler	启用应用弹性策略组合	20
ModifyApplicationAutoscaler	修改弹性伸缩策略组合	20

服务版本相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteApplication	服务删除	20
DescribeApplicationsStatus	单环境下所有应用状态查看	20
ModifyApplicationReplicas	修改应用实例数量	20
RestartApplication	服务重启	20
StopApplication	服务停止	20

其他相关接口

接口名称	接口功能	频率限制（次/秒）
CreateCosToken	生成Cos临时密钥	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：2025-03-28 02:05:37

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 `tem.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `tem.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 `tem.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）	tem.tencentcloudapi.com
华南地区(广州)	tem.ap-guangzhou.tencentcloudapi.com
华东地区(上海)	tem.ap-shanghai.tencentcloudapi.com
华北地区(北京)	tem.ap-beijing.tencentcloudapi.com
西南地区(成都)	tem.ap-chengdu.tencentcloudapi.com
西南地区(重庆)	tem.ap-chongqing.tencentcloudapi.com
港澳台地区(中国香港)	tem.ap-hongkong.tencentcloudapi.com
亚太东南(新加坡)	tem.ap-singapore.tencentcloudapi.com
亚太东南(曼谷)	tem.ap-bangkok.tencentcloudapi.com
亚太东北(首尔)	tem.ap-seoul.tencentcloudapi.com
亚太东北(东京)	tem.ap-tokyo.tencentcloudapi.com
美国东部(弗吉尼亚)	tem.na-ashburn.tencentcloudapi.com
美国西部(硅谷)	tem.na-siliconvalley.tencentcloudapi.com
欧洲地区(法兰克福)	tem.eu-frankfurt.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：2024-11-15 02:06:05

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 tem；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 。
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。

参数名称	类型	必选	描述
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
华北地区（北京）	ap-beijing
西南地区（成都）	ap-chengdu
华南地区（广州）	ap-guangzhou
港澳台地区（中国香港）	ap-hongkong
华东地区（南京）	ap-nanjing
华东地区（上海）	ap-shanghai
亚太东南（新加坡）	ap-singapore
亚太东北（东京）	ap-tokyo

签名方法 v3

最近更新时间：2024-12-25 02:03:16

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

puts canonical_request

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

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

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

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

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

DotNet

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

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

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

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

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

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

// ***** 步骤 2: 拼接待签名字符串 *****

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

// ***** 步骤 3: 计算签名 *****

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

// ***** 步骤 4: 拼接 Authorization *****

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

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

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

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

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

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

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

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

NodeJS

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

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

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

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

function main(){
```

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

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

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

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

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

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

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

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

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

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

C++

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

using namespace std;

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

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

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

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

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

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

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

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

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

    return (ss.str());
}

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

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

```
}
return output;
}

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

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

// ***** 步骤 1: 拼接规范请求串 *****
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string lower = action;
std::transform(action.begin(), action.end(), lower.begin(), ::tolower);
string canonicalHeaders = string("content-type:application/json; charset=utf-8\n")
+ "host:" + host + "\n"
+ "x-tc-action:" + lower + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}] }";
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
cout << canonicalRequest << endl;

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

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

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

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

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

C

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

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

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

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

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

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

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

}

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

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

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

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

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

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

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

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

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

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

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

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

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

其他语言

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

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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

签名方法

最近更新时间：2024-12-25 02:03:16

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
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

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

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
  'Action' : 'DescribeInstances',
  'InstanceIds.0' : 'ins-09dx96dg',
  'Limit' : 20,
  'Nonce' : 11886,
  'Offset' : 0,
  'Region' : 'ap-guangzhou',
  'SecretId' : 'AKID*****',
  'Timestamp' : 1465185768,
  'Version': '2017-03-12',
}
```

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

2.2. 拼接请求字符串

此步骤生成请求字符串。

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

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****
&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

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

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

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-gu
angzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。

首先使用 HMAC-SHA1 算法对上一步中获得的签名原文字符串进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例：

```
$secretKey = '*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&R
egion=ap-guangzhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12';
```

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

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 7RAM2xfNMO9EiVTNmPg06MRnCvQ=，最终得到的签名串请求参数（Signature）为：

7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。

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

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

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

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

5. 签名演示

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

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)
- [Ruby](#)

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

- [Signature Demo](#)

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

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-`

guangzhou&SecretId=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12。

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

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

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，请以对应的实际文档为准。

Java

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

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

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

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

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

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
        // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：params.put("Timestamp", System.currentTimeMillis() / 1000);
        params.put("Timestamp", 1465185768); // 公共参数
        // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID*****
        params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 公共参数
    }
}
```

```
params.put("Action", "DescribeInstances"); // 公共参数
params.put("Version", "2017-03-12"); // 公共参数
params.put("Region", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "HmacSHA1")); // 公共参数
System.out.println(getUrl(params));
}
}
```

Python

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

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import os
import time

import requests

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

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
```

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

Golang

```
package main

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

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

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

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

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

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

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

PHP

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

ksort($params);

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

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

Ruby

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

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

```
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

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

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

DotNet

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

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

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

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

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

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

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

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

NodeJS

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

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

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

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

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

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

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

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

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

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

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

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

返回结果

最近更新时间：2024-03-12 01:54:30

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 06:52:00

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。

环境相关接口

创建环境

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

创建环境

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：CreateEnvironment。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentName	是	String	环境名称 示例值：namespace-a
Description	否	String	环境描述 示例值：命名空间名
Vpc	否	String	私有网络名称 示例值：vpc-1n5javez
SubnetIds.N	否	Array of String	子网列表 示例值：subnet-xxxx
K8sVersion	否	String	K8s version 示例值：1.16.9
SourceChannel	否	Integer	来源渠道 示例值：0
EnableTswTraceService	否	Boolean	是否开启tsw服务。默认值：false 示例值：true
Tags.N	否	Array of Tag	标签 示例值：1
EnvType	否	String	环境类型：test、pre、prod。默认值：prod 示例值：prod
CreateRegion	否	String	创建环境的region 示例值：ap-shanghai
SetupVpc	否	Boolean	是否创建私有网络.默认值:true 示例值：false
SetupPrometheus	否	Boolean	是否创建 Prometheus 实例。默认值：false 示例值：false
PrometheusId	否	String	prometheus 实例 id 示例值：prom-xxx

参数名称	必选	类型	描述
ApmId	否	String	apm id 示例值：apm-xxx

3. 输出参数

参数名称	类型	描述
Result	String	成功时为环境ID，失败为null 示例值：en-xxxxxx
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 创建环境

创建环境

输入示例

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

{
  "EnvironmentName": "name-xxx",
  "Description": "这是一个描述",
  "K8sVersion": "1.69.1",
  "SubnetIds": [
    "subnet-xxx"
  ],
  "EnableTswTraceService": true,
  "SourceChannel": 0,
  "Vpc": "vpc-1n5javez"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": "en-xxxx"
  }
}
```

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.DefaultInternalError	服务器繁忙,请稍后再试。
InternalServerError.CreateEksClusterError	底层集群创建失败。
InternalServerError.DefaultInternalError	服务器繁忙,请稍后再试。
InvalidParameter.UnauthorizedOrMissingRole	权限不足或缺少相关角色
InvalidParameter.VpcOverQuota	自动创建将超过 vpc 限额。
InvalidParameterValue.CreateNamespaceUnsupportRegionError	创建环境createRegion参数错误。
InvalidParameterValue.DailyCreateNamespaceReachMaximum	每日环境创建次数超过上限。
InvalidParameterValue.NamespaceDuplicateError	环境重复。
InvalidParameterValue.NamespaceReachMaximum	环境创建失败，达到上限。
MissingParameter.EnvironmentNameNull	环境名称不能为空。
OperationDenied.BalanceNotEnough	账户余额不足。
UnauthorizedOperation.UnauthorizedOperation	未授权。

获取环境基础信息

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

获取环境基础信息

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：DescribeEnvironment。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	命名空间id 示例值：en-xxxxxx
SourceChannel	否	Integer	来源Channel 示例值：0

3. 输出参数

参数名称	类型	描述
Result	NamespaceInfo	环境信息
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 获取环境状态

获取环境状态

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
  "SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "85435f86-a454-4bf8-8e8c-074a51f55bb2",
    "Result": {
      "EnvironmentId": "en-15mmxey5",
      "EnvironmentName": "bugfix-gz",
      "Region": "ap-guangzhou",
      "VpcId": "vpc-9xcgo41t",
      "SubnetIds": [
        "subnet-orey48cy"
      ],
      "Description": "这是一个描述",
      "CreateDate": "2024-07-01 17:48:35",
      "ApmInstanceId": "",
      "Locked": 0,
      "NamespaceName": "",
      "Tags": [],
      "EnvType": "prod"
    }
  }
}
```

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.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.NamespaceNotFound	环境不存在。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。

错误码	描述
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

获取环境状态

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

获取环境状态

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：DescribeEnvironmentStatus。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentIds.N	是	Array of String	命名空间id 示例值：en-xxxxxx
SourceChannel	否	Integer	来源Channel 示例值：0

3. 输出参数

参数名称	类型	描述
Result	Array of NamespaceStatusInfo	返回状态列表 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 获取环境状态

获取环境状态

输入示例

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

{
  "EnvironmentIds": [
    "en-xxxxxx"
  ],
  "SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "Result": [
      {
        "EnvironmentId": "en-xxxxxx",
        "EnvironmentName": "en-name-xxx",
        "ClusterId": "cls-xxxxxx",
        "ClusterStatus": "running",
        "EnvironmentStartingStatus": {
          "ApplicationNumNeedToStart": 0,
          "StartedApplicationNum": 0,
          "StartFailedApplicationNum": 0
        },
        "EnvironmentStoppingStatus": {
          "ApplicationNumNeedToStop": 0,
          "StoppedApplicationNum": 0,
          "StopFailedApplicationNum": 0
        }
      }
    ],
    "RequestId": "1se3sd-xxx-xxx-xxx"
  }
}
```

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.ActionReadTimeout	请求响应超时。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。

错误码	描述
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.NamespaceNotFound	环境不存在。
UnauthorizedOperation.UnauthorizedOperation	未授权。

获取环境列表

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：`tem.tencentcloudapi.com`。

获取环境列表

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：DescribeEnvironments。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	否	Integer	分页limit，默认：20 示例值：20
Offset	否	Integer	分页下标，默认：0 示例值：0
SourceChannel	否	Integer	来源source 示例值：0
Filters.N	否	Array of QueryFilter	查询过滤器 示例值：1
SortInfo	否	SortType	排序字段 示例值：{"Key":"UpdateTime","Type":1}
EnvironmentId	否	String	环境id 示例值：en-xxxxxx

3. 输出参数

参数名称	类型	描述
Result	NamespacePage	返回结果 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 获取环境列表

获取环境列表

输入示例

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

X-TC-Action: DescribeEnvironments

<公共请求参数>

```
{}
```

输出示例

```
{
  "Response": {
    "RequestId": "a438e9af-34a0-47d8-bdfc-fc42f652ad6f",
    "Result": {
      "Pages": 1,
      "Records": [
        {
          "EnvironmentId": "en-xxxxxx",
          "AppId": "130573xxx",
          "ApplicationNum": 0,
          "RunInstancesNum": 0,
          "Uin": "10001905xxx",
          "SubAccountUin": "100000799xxx",
          "EnvironmentName": "test-cluster",
          "Region": "ap-guangzhou",
          "RegionId": "1",
          "Description": "",
          "ClusterId": "cls-gtqz9mxx",
          "ClusterStatus": "NORMAL",
          "Vpc": "vpc-3le4yaxx",
          "SubnetId": "subnet-pm9fq4xx",
          "CreateDate": "2024-11-07 16:57:21",
          "ModifyDate": "2024-11-07 16:57:21",
          "Modifier": "100019051593",
          "Creator": "100019051593",
          "Locked": 0,
          "Channel": "",
          "Status": 1,
          "EnableTswTraceService": false,
          "Tags": [],
          "EnvType": "prod",
          "HasAuthority": true
        }
      ],
      "Total": 1,
      "Size": 20,
      "Current": 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

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.ActionReadTimeout	请求响应超时。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.InvalidTenantInfo	无法识别租户信息。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.NamespaceNotFound	环境不存在。

销毁环境

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

销毁环境

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：DestroyEnvironment。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	命名空间ID 示例值：en-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	Boolean	返回结果 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 销毁环境

销毁环境

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
  "SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceInUse.EnvironmentAlreadyLocked	环境已锁定。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

编辑环境

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

编辑环境

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：ModifyEnvironment。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境id 示例值：en-36pm4p8r
EnvironmentName	否	String	环境名称。环境名称不可修改 示例值：namespace-a
Description	否	String	环境描述 示例值：命名空间
Vpc	否	String	私有网络名称 示例值：vpc-1n5javez
SubnetIds.N	否	Array of String	子网网络 示例值：subnet-m48cb76m
SourceChannel	否	Integer	来源渠道 示例值：0
EnvType	否	String	环境类型：test、pre、prod 示例值：prod

3. 输出参数

参数名称	类型	描述
Result	Boolean	成功时为环境ID，失败为null 示例值：true
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时 需要提供该次请求的 RequestId。

4. 示例

示例1 编辑环境

编辑环境

输入示例

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

```
{
  "EnvironmentName": "name-xxx",
  "Description": "这是一个描述",
  "EnvironmentId": "en-xxxxxx",
  "SubnetIds": [
    "subnet-xxxxxx"
  ],
  "SourceChannel": 0,
  "Vpc": "vpc-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "xxx-xxx-xxx"
  }
}
```

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	添加子网/虚拟节点异常。

错误码	描述
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.EnvironmentNameImmutable	环境名称不可变。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
OperationDenied.ResourceIsolated	账号欠费状态下不支持该操作,请冲正后重试。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

网关相关接口

查询 Ingress 规则列表

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询 Ingress 规则列表

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：DescribeIngresses。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 id 示例值：en-xxxxxx
ClusterNamespace	是	String	环境 namespace 示例值：default
SourceChannel	否	Integer	来源渠道 示例值：0
IngressNames.N	否	Array of String	ingress 规则名列表 示例值：["aa"]

3. 输出参数

参数名称	类型	描述
Result	Array of IngressInfo	ingress 数组
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 Ingress 规则列表

查询 Ingress 规则列表

输入示例

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

<公共请求参数>

```
{
  "ClusterNamespace": "default",
  "EnvironmentId": "en-xxxxxx",
  "IngressNames": [
    "ingress-name"
  ],
  "SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": [
      {
        "ClusterId": "cls-xxxxxx",
        "EnvironmentId": "en-xxxxxx",
        "IngressName": "ingress-name-xxx",
        "ClusterNamespace": "default",
        "AddressIPVersion": "IPV4",
        "ClbId": "clb-xxxx",
        "Vip": "10.xx.xx.xx",
        "Rules": [
          {
            "Host": "",
            "Http": {
              "Paths": [
                {
                  "Path": "/path",
                  "Backend": {
                    "ServiceName": "kubernetes",
                    "ServicePort": 443
                  }
                }
              ]
            }
          }
        ],
        "Tls": [
          {
            "Hosts": [
              "xxx.com"
            ],
            "SecretName": "secret-name-xxx"
          }
        ],
        "CreateTime": "2024-12-04 20:46:29",
        "Mixed": true,
        "RewriteType": "AUTO",
        "Domain": "xxx.xxx.com"
      }
    ]
  }
}
```

```
}
}
```

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.DescribeServiceListError	查询service列表失败。
InternalServerError.DescribeServiceListError	查询service列表失败。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。

查询应用关联的 Ingress 规则列表

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询应用关联的 Ingress 规则列表

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：DescribeRelatedIngresses。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 id 示例值：en-xxxxxx
ClusterNamespace	是	String	环境 namespace 示例值：default
ApplicationId	是	String	应用 ID 示例值：app-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	Array of IngressInfo	ingress 数组
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 查询应用关联的 Ingress 规则列表

查询应用关联的 Ingress 规则列表

输入示例

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

```
{
  "EnvironmentId": "en-xxxxxx",
  "ClusterNamespace": "default",
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": [
      {
        "ClusterId": "cls-xxxxxx",
        "IngressName": "ingress-name-xxx",
        "ClusterNamespace": "default",
        "EnvironmentId": "en-xxxxxx",
        "AddressIPVersion": "IPV4",
        "ClbId": "clb-xxxx",
        "Vip": "10.xx.xx.xx",
        "Mixed": true,
        "Rules": [
          {
            "Host": "",
            "Http": {
              "Paths": [
                {
                  "Path": "/path",
                  "Backend": {
                    "ServiceName": "svc-name-xxx",
                    "ServicePort": 443
                  }
                }
              ]
            }
          }
        ],
        "Tls": [
          {
            "Hosts": [
              "xxx.com"
            ],
            "SecretName": "secret-name-xxx"
          }
        ],
        "CreateTime": "2024-12-04 20:46:29",
        "RewriteType": "AUTO",
        "Domain": "xxx.xxx.com"
      }
    ]
  }
}
```

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.DescribeServiceIngressError	查询服务关联的 ingress 失败。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
MissingParameter.NamespaceIdNull	环境ID不能为空。
MissingParameter.ServiceIdNull	服务ID不能为空。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。

删除 Ingress 规则

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

删除 Ingress 规则

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：DeleteIngress。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境ID 示例值：env-xxxxxx
ClusterNamespace	是	String	环境 namespace 示例值：default
IngressName	是	String	ingress 规则名 示例值：ingress-name
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 删除 Ingress 规则

删除 Ingress 规则

输入示例

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

```
{
  "EnvironmentId": "en-xxxxxx",
  "ClusterNamespace": "default",
  "SourceChannel": 0,
  "IngressName": "ingress-name"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.DeleteIngressError	删除 ingress 失败。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
UnauthorizedOperation.UnauthorizedOperation	未授权。

查询 Ingress 规则

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询 Ingress 规则

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：DescribeIngress。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
ClusterNamespace	是	String	环境namespace 示例值：default
IngressName	是	String	ingress 规则名 示例值：name-xxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	IngressInfo	Ingress 规则配置 示例值：ingressInfo
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 Ingress 规则

查询 Ingress 规则

输入示例

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

```
{
  "EnvironmentId": "en-xxxxxx",
  "ClusterNamespace": "default",
  "SourceChannel": 0,
  "IngressName": "name-xxx"
}
```

输出示例

```
{
  "Response": {
    "Result": {
      "EnvironmentId": "en-xxxxxx",
      "AddressIPVersion": "IPV4",
      "IngressName": "ingress-name-xxx",
      "Rules": [
        {
          "Host": "11.xx.xx.xx",
          "Http": {
            "Paths": [
              {
                "Path": "/path",
                "Backend": {
                  "ServiceName": "svc-name-xxx",
                  "ServicePort": 8080
                }
              }
            ]
          },
          "Protocol": "TCP"
        }
      ],
      "ClbId": "lb-xxx",
      "ClusterNamespace": "default",
      "Tls": [
        {
          "Hosts": [
            "11.xx.xx.xx"
          ],
          "SecretName": "secret-name-xxx",
          "CertificateId": "certId-xxx"
        }
      ],
      "ClusterId": "cls-xxxxxx",
      "Vip": "11.xx.xx.xx",
      "CreateTime": "2024-12-04 20:46:29",
      "Mixed": true,
      "RewriteType": "AUTO",
      "Domain": ""
    },
    "RequestId": "abc-xxx-xxx-xxx"
  }
}
```

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.DescribeIngressError	查询 ingress 失败。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

创建或者更新 Ingress 规则

最近更新时间：2025-04-25 02:00:10

1. 接口描述

此接口处于预下线状态。

预计下线时间：2025-03-27 17:07:14

接口请求域名：tem.tencentcloudapi.com。

此接口没有被使用了

创建或者更新 Ingress 规则

默认接口请求频率限制：20次/秒。

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyIngress。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Ingress	是	IngressInfo	Ingress 规则配置
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	Boolean	创建成功 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建或者更新 Ingress 规则

创建或者更新 Ingress 规则

输入示例

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

{
  "SourceChannel": 0,
  "Ingress": {
    "Tls": [
      {
```

```
"CertificateId": "xx",
"Hosts": [
  "xxx.com"
],
"SecretName": "xx"
},
{
  "EnvironmentId": "xx",
  "ClusterNamespace": "xx",
  "Rules": [
    {
      "Protocol": "xx",
      "Host": "xx",
      "Http": {
        "Paths": [
          {
            "Path": "xx",
            "Backend": {
              "ServiceName": "xx",
              "ServicePort": 443
            }
          }
        ]
      }
    }
  ],
  "ClbId": "xx",
  "ClusterId": "xx",
  "IngressName": "xx",
  "Vip": "xx",
  "Mixed": true,
  "AddressIPVersion": "xx",
  "CreateTime": "xx"
}
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.UpdateIngressError	更新 ingress 失败。
InternalServerError.UpdateIngressError	更新 ingress 失败。
InvalidParameterValue.IngressRewriteRequiredHttpsEnable	访问配置重定向需要启用 HTTPS。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
UnauthorizedOperation.UnauthorizedOperation	未授权。

资源相关接口

绑定云资源

最近更新时间：2025-04-25 02:00:09

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

绑定云资源

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：CreateResource。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 Id 示例值：en-xxxxxx
ResourceType	是	String	资源类型，目前支持文件系统：CFS；注册中心：TSE_SRE 示例值：CFS
ResourceId	否	String	资源 Id 示例值：cls-xxx
SourceChannel	否	Integer	来源渠道 示例值：0
ResourceFrom	否	String	资源来源，目前支持：existing，已有资源；creating，自动创建 示例值：existing
ResourceConfig	否	String	设置 resource 的额外配置 示例值：resource-cfg-xxx

3. 输出参数

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

4. 示例

示例1 绑定云资源

绑定云资源

输入示例

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

```
{
  "EnvironmentId": "en-xxxxxx",
  "ResourceType": "CFS",
  "ResourceId": "resource-xxxxxx",
  "SourceChannel": 0,
  "ResourceFrom": "existing"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.NamespaceReachMaximum	环境创建失败，达到上限。

错误码	描述
InvalidParameterValue.NamespaceResourceReachMaximum	环境资源创建失败，达到上限。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
InvalidParameterValue.VpcInvalid	vpc错误, 与环境vpc不一致。
ResourceInUse.ResourceAlreadyUsed	资源已绑定。
ResourceNotFound.NamespaceNotFound	环境不存在。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

应用相关接口

创建应用

最近更新时间：2025-04-27 21:16:38

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

创建应用

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：CreateApplication。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationName	是	String	应用名 示例值：service-a
Description	是	String	描述 示例值：服务a
UseDefaultImageService	否	Integer	是否使用默认镜像服务 1-是，0-否 示例值：0
RepoType	否	Integer	如果是绑定仓库，绑定的仓库类型，0-个人版，1-企业版 示例值：0
Instanceld	否	String	企业版镜像服务的实例id 示例值：tr-123
RepoServer	否	String	绑定镜像服务器地址 示例值：tcr.tencent.com
RepoName	否	String	绑定镜像仓库名 示例值：tem-test/repo_xxx
SourceChannel	否	Integer	来源渠道 示例值：0
SubnetList.N	否	Array of String	应用所在子网 示例值：["subnet-xxxxxx"]
CodingLanguage	否	String	编程语言 - JAVA - OTHER 示例值：JAVA
DeployMode	否	String	部署方式 - IMAGE - JAR - WAR 示例值：IMAGE

参数名称	必选	类型	描述
EnableTracing	否	Integer	是否开启 Java 应用的 APM 自动上报功能，1 表示启用；0 表示关闭 示例值：1
UseDefaultImageServiceParameters	否	UseDefaultRepoParameters	使用默认镜像服务额外参数
Tags.N	否	Array of Tag	标签

3. 输出参数

参数名称	类型	描述
Result	String	应用ID 示例值：app-xxxxxx
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 创建应用

创建应用

输入示例

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

{
  "ApplicationName": "tem-app",
  "Description": "这是一个描述",
  "CodingLanguage": "JAVA",
  "RepoType": 3
}
```

输出示例

```
{
  "Response": {
    "Result": "57d58b44-ee28-4db6-acba-c1cb0cxxx",
    "RequestId": "app-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.CreateServiceError	创建服务失败。
FailedOperation.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.ActionReadTimeout	请求响应超时。
InternalServerError.CreateServiceError	创建服务失败。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.TagInterfaceError	调用标签接口异常。
InvalidParameterValue.InvalidServiceName	应用名格式非法。
InvalidParameterValue.PublicRepoTypeParameterError	公有镜像参数错误。
InvalidParameterValue.ServiceLowerCase	服务名必须小写。
InvalidParameterValue.ServiceNameDuplicateError	应用名已存在。
InvalidParameterValue.ServiceReachMaximum	应用创建失败，达到上限。
InvalidParameterValue.TcrEntInstanceNameNotValid	企业版 TCR 实例名称不合法。
MissingParameter.TcrEntInstanceNameNull	企业版 TCR 实例名称不能为空。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
UnauthorizedOperation.UnauthorizedOperation	未授权。
UnsupportedOperation.UnsupportAction	不支持的ACTION。

生成应用程序包预签名下载链接

最近更新时间：2025-04-27 21:16:37

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

生成应用程序包预签名下载链接

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：GenerateApplicationPackageDownloadUrl。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	应用ID 示例值：app-xxxxxx
PkgName	是	String	包名 示例值：test.jar
DeployVersion	是	String	需要下载的包版本 示例值：v1
SourceChannel	否	Integer	来源 channel 示例值：0

3. 输出参数

参数名称	类型	描述
Result	String	包下载临时链接 示例值：http://download.demo
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 生成应用程序包预签名下载链接

生成应用程序包预签名下载链接

输入示例

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

```
{
  "PkgName": "test.jar",
  "DeployVersion": "v1",
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": "abc-xxxx"
  }
}
```

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
UnauthorizedOperation.UnauthorizedOperation	未授权。

修改应用基本信息

最近更新时间：2025-04-27 21:16:37

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

修改应用基本信息

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：ModifyApplicationInfo。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	应用ID 示例值：app-xxxxxxx
Description	是	String	描述 示例值：服务a
SourceChannel	否	Integer	来源渠道 示例值：0
EnableTracing	否	Integer	是否开启调用链,（此参数已弃用） 示例值：1

3. 输出参数

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

4. 示例

示例1 修改应用基本信息

修改应用基本信息

输入示例

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

```
{
  "SourceChannel": 0,
  "ApplicationId": "app-xxxxxx",
  "Description": "这是一个描述",
  "EnableTracing": 1
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "1dew34d-xxx-xxx-xxx"
  }
}
```

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.CreateApmResourceError	创建apm资源失败。
InternalServerError.CreateServiceError	创建服务失败。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.ServiceNotFound	找不到应用。
UnauthorizedOperation.UnauthorizedOperation	未授权。

服务相关接口

获取运行服务列表

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

获取运行服务列表

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：DescribeApplications。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	否	String	命名空间ID 示例值：en-xxxxxx
Limit	否	Integer	分页Limit，默认值：20 示例值：20
Offset	否	Integer	分页offset,默认值：0 示例值：0
SourceChannel	否	Integer	来源渠道 示例值：0
ApplicationId	否	String	服务id 示例值：app-xxxxxx
Keyword	否	String	搜索关键字 示例值：1
Filters.N	否	Array of QueryFilter	查询过滤器 示例值：1
SortInfo	否	SortType	排序字段 示例值：{}

3. 输出参数

参数名称	类型	描述
Result	ServicePage	返回结果 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 获取运行服务列表

获取运行服务列表

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "Result": {
      "Records": [
        {
          "ApplicationId": "app-5vaz8d85",
          "ApplicationName": "pk-test-internet",
          "Description": "",
          "EnvironmentId": "",
          "CreateDate": "2024-10-16 14:17:08",
          "ModifyDate": "2024-10-23 20:17:34",
          "Modifier": "100019051593",
          "Creator": "100019051593",
          "RepoType": 3,
          "InstanceId": "",
          "RepoName": "",
          "EnvironmentName": "",
          "CodingLanguage": "JAVA",
          "DeployMode": "",
          "ActiveVersions": [
            {
              "VersionName": "20241023161452",
              "VersionId": "revision-5n3oveqj",
              "Status": "",
              "BatchDeployStatus": "",
              "EnableEs": -1,
              "LogOutputConf": null,
              "ExpectedInstances": null,
              "CurrentInstances": 0,
              "DeployMode": "JAR",
              "BuildTaskId": "",
              "EnvironmentId": "en-259d2b6j",
              "EnvironmentName": "pk-test-12",
              "ApplicationId": "app-5vaz8d85",
              "ApplicationName": "pk-test-internet",
              "Zones": [],
              "NodeInfos": [],
              "PodList": null,
              "WorkloadInfo": null,
              "CreateDate": "2024-10-23 20:17:34",
              "UnderDeploying": null,
            }
          ]
        }
      ]
    }
  }
}
```

```

"RegionId": "1"
}
],
"EnableTracing": 0,
"Tags": [],
"HasAuthority": true
}
],
"Total": 0,
"Size": 0,
"Pages": 0,
"Current": 0
},
"RequestId": "67796ca7-0b76-4281-a280-c9811cd8ce62"
}
}

```

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.ActionReadTimeout	请求响应超时。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。

获取分批发布详情

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

获取分批发布详情

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：DescribeDeployApplicationDetail。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境id 示例值：en-xxxxxx
VersionId	否	String	版本部署id 示例值：version-123

3. 输出参数

参数名称	类型	描述
Result	TemDeployApplicationDetailInfo	分批发布结果详情
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 获取分批发布详情

获取分批发布详情

输入示例

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

{
  "ApplicationId": "app-xxxxxx",
  "EnvironmentId": "en-xxxxxx",
```

```
"VersionId": "version-xxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "7559e31e-4e32-476d-93f2-db6bcbab530a",
    "Result": {
      "DeployStrategyConf": {
        "TotalBatchCount": 1,
        "BetaBatchNum": 0,
        "DeployStrategyType": 3,
        "BatchInterval": 0,
        "MinAvailable": 1,
        "Force": true
      },
      "StartTime": "2024-12-20 17:00:17",
      "EndTime": "",
      "Status": "Error",
      "CurrentBatchStatus": "Finish",
      "ErrorMessage": "",
      "BetaBatchDetail": null,
      "OtherBatchDetail": [
        {
          "OldPods": [],
          "NewPods": [
            {
              "PodId": "app0925-4sbct",
              "PodVersion": "",
              "CreateTime": "2024-12-20 17:00:23",
              "Zone": "ap-shanghai-2",
              "Webshell": "https://tkecache.cloud.tencent.com/xxx",
              "Status": "CrashLoopBackOff",
              "PodStatus": [
                "CrashLoopBackOff"
              ]
            },
            {
              "PodId": "app0925-s4vpj",
              "PodVersion": "",
              "CreateTime": "2024-12-20 17:00:23",
              "Zone": "ap-shanghai-2",
              "Webshell": "https://tkecache.cloud.tencent.com/xxx",
              "Status": "CrashLoopBackOff",
              "PodStatus": [
                "CrashLoopBackOff"
              ]
            }
          ],
          "BatchStatus": "Finish",
          "PodNum": 3,
          "BatchIndex": 0,
          "NextBatchStartTime": 0,
          "OldPodList": null,
          "NewPodList": null
        }
      ]
    }
  }
}
```

```
}
],
"OldVersionPodList": {
  "Offset": 0,
  "Limit": 1,
  "TotalCount": 1,
  "PodList": [
    {
      "Zone": "ap-shanghai-2",
      "NodeInfo": {
        "Name": "eklet-subnet-8indzsuv-457482",
        "Zone": "ap-shanghai-2",
        "SubnetId": "subnet-8indzsuv",
        "AvailableIpCount": "226",
        "Cidr": "10.0.30.0-24"
      },
      "DeployVersion": "20241202165113",
      "RestartCount": 0,
      "Ready": true,
      "StartTime": "2024-12-19T03:12:44Z",
      "ContainerState": "running",
      "Unhealthy": null,
      "UnhealthyWarningMsg": "",
      "VersionId": "revision-57moqvm5",
      "ApplicationName": "app0925",
      "Status": "Running",
      "CreateTime": "2024-12-19 11:11:32",
      "Webshell": "https://tkecache.cloud.tencent.com/xxx",
      "PodId": "app0925-jt5bn",
      "PodIp": "10.0.30.158"
    }
  ],
  "RequestId": ""
},
"CurrentBatchIndex": 0,
"NewVersionPackageInfo": "K8sEventDemo-1.0.jar",
"NewDeployVersion": "20241202165113",
"OldDeployVersion": "20241202165113",
"NextBatchStartTime": null
}
}
}
```

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.DescribeRunPodListError	查询实例信息失败。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DescribeRunPodListError	查询实例信息失败。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
ResourceNotFound.VersionServiceNotFound	找不到版本对应的应用。
UnauthorizedOperation.UnauthorizedOperation	未授权。

开始下一批次发布

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

开始下一批次发布

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：ResumeDeployApplication。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	需要开始下一批次的服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境id 示例值：en-xxxxxx

3. 输出参数

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

4. 示例

示例1 开始下一批次发布

输入示例

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

{
  "ApplicationId": "app-xxxxxx",
  "EnvironmentId": "en-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "xx-xxx-xxx"
  }
}
```

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
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

回滚分批发布

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

回滚分批发布

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：RevertDeployApplication。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	需要回滚的服务id 示例值：app-xxxxxx
EnvironmentId	是	String	需要回滚的服务所在环境id 示例值：en-xxxxxx

3. 输出参数

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

4. 示例

示例1 回滚分批发布

输入示例

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

{
  "ApplicationId": "app-xxxxxx",
  "EnvironmentId": "en-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "abc-xxx-xxxx"
  }
}
```

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
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

工作负载相关接口

应用部署

最近更新时间：2025-04-27 21:16:39

1. 接口描述

接口请求域名：tem.tencentcloudapi.com 。

应用部署

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：DeployApplication。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	应用ID 示例值：app-xxxxxx
InitPodNum	是	Integer	初始化 pod 数 示例值：1
CpuSpec	是	Float	cpu规格 单位：核 示例值：1
MemorySpec	是	Float	内存规格 单位：G 示例值：2
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
DeployVersion	是	String	部署类型为 IMAGE 时，该参数表示镜像 tag。 部署类型为 JAR/WAR 时，该参数表示包版本号。 示例值：v1
ImgRepo	否	String	镜像仓库 示例值：1
VersionDesc	否	String	版本描述信息 示例值：version-desc
JvmOpts	否	String	启动参数 示例值：-Xms128m -XX:MetaspaceSize=128m
EsInfo	否	EsInfo	弹性伸缩配置（已废弃，请使用HorizontalAutoscaler设置弹性策略） 示例值：1
EnvConf.N	否	Array of Pair	环境变量配置 示例值：1
LogConfs.N	否	Array of String	日志配置 示例值：1
StorageConfs.N	否	Array of StorageConf	数据卷配置

参数名称	必选	类型	描述
			示例值：1
StorageMountConfs.N	否	Array of StorageMountConf	数据卷挂载配置 示例值：1
DeployMode	否	String	部署类型。 - JAR：通过 jar 包部署 - WAR：通过 war 包部署 - IMAGE：通过镜像部署 示例值：JAR
PkgName	否	String	传入内容为 /jar包名字 的形式。也就是在 jar包名字前增加一个/。 如上传的 jar 包名字为 demo-1.0.0.jar，那么这里传入内容为：/demo-1.0.0.jar 注：jar 包需要通过 tem 页面上传过，tem 后端才能拉到该 jar 包。 示例值：/demo-1.0.0.jar
JdkVersion	否	String	JDK 版本。 - KONA:8：使用 kona jdk 8。 - OPEN:8：使用 open jdk 8。 - KONA:11：使用 kona jdk 11。 - OPEN:11：使用 open jdk 11。 示例值：KONA:8
SecurityGroupIds.N	否	Array of String	安全组ID s 示例值：1
LogOutputConf	否	LogOutputConf	日志输出配置 示例值：1
SourceChannel	否	Integer	来源渠道 示例值：0
Description	否	String	版本描述 示例值：1
ImageCommand	否	String	镜像命令 示例值：cd
ImageArgs.N	否	Array of String	镜像命令参数 示例值：/home
UseRegistryDefaultConfig	否	Boolean	是否添加默认注册中心配置 示例值：true
SettingConfs.N	否	Array of MountedSettingConf	挂载配置信息 示例值：[{"ConfigDataName":"cfg-test-1","MountedPath":"/data","SecretDataName":""}]
Service	否	EksService	应用访问设置 示例值：1
VersionId	否	String	要回滚到的历史版本id 示例值：version-123
PostStart	否	String	启动后执行的脚本 示例值：1
PreStop	否	String	停止前执行的脚本 示例值：1
Liveness	否	HealthCheckConfig	存活探针配置 示例值：1
Readiness	否	HealthCheckConfig	就绪探针配置 示例值：1

参数名称	必选	类型	描述
DeployStrategyConf	否	DeployStrategyConf	分批发布策略配置 示例值：1
HorizontalAutoscaler.N	否	Array of HorizontalAutoscaler	弹性策略（已弃用，请使用弹性伸缩策略组合相关接口） 示例值：1
CronHorizontalAutoscaler.N	否	Array of CronHorizontalAutoscaler	定时弹性策略（已弃用，请使用弹性伸缩策略组合相关接口） 示例值：1
LogEnable	否	Integer	是否启用log，1为启用，0为不启用 示例值：1
ConfEdited	否	Boolean	（除开镜像配置）配置是否修改 示例值：true
SpeedUp	否	Boolean	是否开启应用加速 示例值：true
StartupProbe	否	HealthCheckConfig	启动探针配置 示例值：1
OsFlavour	否	String	操作系统版本； 当选择openjdk时，可选参数： - ALPINE - CENTOS 当选择konajdk时，可选参数： - ALPINE - TENCENTOS 示例值：1
EnablePrometheusConf	否	EnablePrometheusConf	metrics业务指标监控配置 示例值：{"Port":9090,"Path":"/data"}
EnableTracing	否	Integer	1：开始自动apm采集（skywalking）； 0：关闭apm采集； 示例值：0
EnableMetrics	否	Integer	1：开始自动metrics采集（open-telemetry）； 0：关闭metrics采集； 示例值：0
TcrInstanceId	否	String	镜像部署时，选择的tcr实例id 示例值：tcr-123
RepoServer	否	String	镜像部署时，选择的镜像服务器地址 示例值：ccr.ccs.tencentcryn.com
RepoType	否	Integer	镜像部署时，仓库类型：0：个人仓库；1：企业版；2：公共仓库；3：tem托管仓库；4：demo仓库 示例值：0
PostStartEncoded	否	String	启动后执行的脚本，base64 编码 示例值：post-start-encoded-xxx
PreStopEncoded	否	String	停止前执行的脚本，base64 编码 示例值：pre-stop-encoded-xxx

3. 输出参数

参数名称	类型	描述
Result	String	版本ID（前端可忽略） 示例值：revision-xxxxxx
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 应用部署

应用部署

输入示例

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

{
  "ApplicationId": "app-5vaz8x85",
  "ConfEdited": false,
  "DeployMode": "JAR",
  "JdkVersion": "KONA:11",
  "PkgName": "tem/pkg/130xxxx/app-5vaz8x85/1733xxxx/K8sDemo-1.0.jar",
  "SpeedUp": true,
  "OsFlavour": "ALPINE",
  "DeployVersion": "20241202165113",
  "JvmOpts": "-Xms128m -XX:MetaspaceSize=128m",
  "InitPodNum": 3,
  "CpuSpec": 1,
  "MemorySpec": 2,
  "EnvConf": [
    {
      "Key": "key",
      "Value": "val",
      "Type": "default",
      "Config": "",
      "Secret": ""
    }
  ],
  "LogEnable": 0,
  "StorageConfs": [],
  "StorageMountConfs": [],
  "SecurityGroupIds": [
    "sg-1zwhplpi"
  ],
  "EnvironmentId": "en-dpxydz5",
  "Service": {
    "ServicePortMappingList": [
      {
        "Type": "CLUSTER",
        "ServiceName": "k8s-svc-test",
        "ClusterIp": "172.16.56.122",
        "ExternalIp": "10.xx.xx.xx",
        "ExternalDomain": "ext.domain.com",
        "SubnetId": "subnet-xxx",
        "VpcId": "vpc-oxyzq5xx",
        "LoadBalanceId": "10.xx.xx.xx",
        "Yaml": "apiVersion: xxxxx",
        "Ports": [
          8080
        ]
      }
    ]
  }
}
```

```
"PortMappingItemList": [
  {
    "Port": 8080,
    "TargetPort": 8080,
    "Protocol": "TCP"
  }
],
"EnableRegistryNextDeploy": 0,
"FlushAll": true
},
"UseRegistryDefaultConfig": false,
"EnableTracing": 0,
"SettingConfs": [
  {
    "ConfigDataName": "cfg-test-1",
    "MountedPath": "/data",
    "SecretDataName": "secret-name"
  }
],
"DeployStrategyConf": {
  "TotalBatchCount": 1,
  "BetaBatchNum": 0,
  "DeployStrategyType": 3,
  "MinAvailable": 1,
  "Force": true,
  "BatchInterval": 0
},
"PostStart": "/bin/sh\n-c\ncurl localhost:8080/postStart",
"PreStop": "/bin/sh\n-c\nwget --server-response --spider -S --no-verbose --header=\"Content-Type:application/json\" --post-data='{\"serverName\": \"api-basic\"}' \"http://localhost:9900/actuator/stopmsg\"\\nsleep 30s"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": "revision-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.ActionReadTimeout	请求响应超时。
FailedOperation.CreateServiceError	创建服务失败。
FailedOperation.DefaultInternalError	服务器繁忙,请稍后再试。
FailedOperation.DescribeServiceListError	查询service列表失败。
InternalError.ActionReadTimeout	请求响应超时。
InternalError.CreateApmResourceError	创建apm资源失败。
InternalError.CreateServiceError	创建服务失败。
InternalError.DefaultInternalError	服务器繁忙,请稍后再试。
InternalError.DeployVersionError	版本部署调用失败。
InvalidParameterValue.ApmNotBind	APM 没有与当前环境绑定。
InvalidParameterValue.AutoScalerLargerThanOne	只支持绑定一种弹性伸缩。
InvalidParameterValue.CannotOverWriteOtherApplicationService	不能覆盖其他应用的访问方式。
InvalidParameterValue.InvalidDeployVersion	版本号格式非法。
InvalidParameterValue.InvalidEnvName	环境变量名非法, 要求有由字母、数字, ".", "_" 和 "-" 组成, 不能由数字开头。
InvalidParameterValue.InvalidEnvValue	环境变量长度不能超过1000。
InvalidParameterValue.InvalidMountPath	挂载路径不合法, 不能为 /app。
InvalidParameterValue.InvalidTenantInfo	无法识别租户信息。
InvalidParameterValue.JdkVersionRequired	JDK 版本不能为空。
InvalidParameterValue.MustProvidePortMappingRules	必须提供PortMapping规则。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.OsNotSupport	操作系统不支持。
InvalidParameterValue.PostStartNotValid	post start 不合法。
InvalidParameterValue.PreStopNotValid	pre stop 不合法。
InvalidParameterValue.RegistryNotBind	注册中心没有与当前环境绑定。
InvalidParameterValue.ServiceNameDuplicateError	应用名已存在。
InvalidParameterValue.ServiceNotBelongToAppid	应用不属于此账户。
InvalidParameterValue.ServicePodReachMaximum	实例创建失败, 达到上限。
InvalidParameterValue.ServiceUseReserveSuffix	服务名使用了保留后缀。

错误码	描述
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
InvalidParameterValue.TraitsTracingNotSupported	非 JAVA 应用不支持链路追踪特性。
InvalidParameterValue.VersionLengthLimit	version 不能超过128位。
InvalidParameterValue.VersionLowerCase	版本必须小写。
MissingParameter.DeployModeNull	部署方式不能为空。
MissingParameter.DeployVersionNull	部署版本不能为空。
MissingParameter.ImgRepoNull	镜像仓库不能为空。
MissingParameter.LogsetOrTopicNull	logset 和 topic 不能为空。
MissingParameter.PkgNameNull	包名不能为空。
MissingParameter.RepoServerNull	镜像仓库服务器不能为空。
MissingParameter.ServiceIdNull	服务ID不能为空。
MissingParameter.SvcRepoNotReady	镜像仓库还未就绪。
MissingParameter.VpcServiceSubnetNull	vpc内访问类型子网ID不能为空。
ResourceInUse.ServiceDeploying	服务部署中。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
ResourceUnavailable.WaitForKruise	等待组件安装。
UnauthorizedOperation.MissingEksLogRole	缺少容器服务的 CLS 日志角色，请打开控制台授权。
UnauthorizedOperation.UnauthorizedOperation	未授权。

服务基本信息查看

最近更新时间：2025-04-27 21:16:39

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

服务基本信息查看

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：DescribeApplicationInfo。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务版本ID 示例值：app-xxxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0
EnvironmentId	否	String	环境ID 示例值：en-xxxxxxx

3. 输出参数

参数名称	类型	描述
Result	TemServiceVersionInfo	返回结果 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 服务基本信息查看

服务基本信息查看

输入示例

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

{
  "SourceChannel": 0,
```

```
"ApplicationId": "app-xxxxxx",
"EnvironmentId": "en-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b86d4065-771d-485c-9130-8914fe5fd1bc",
    "Result": {
      "AppId": "1305738457",
      "SubAccountUin": "100000799529",
      "Uin": "100019051593",
      "Region": "ap-shanghai",
      "VersionId": "revision-jorxnl6j",
      "VersionName": "20241202165113",
      "ApplicationId": "app-5vaz8x85",
      "Vpc": "vpc-e3ay8gku",
      "SubnetId": "subnet-kctzw7nj,subnet-8indzsuv,subnet-gau65gm1",
      "DeployMode": "JAR",
      "JdkVersion": "KONA:11",
      "Description": "",
      "DeployVersion": "20241202165113",
      "GroupId": "group-job6vv75",
      "PublishMode": "",
      "InitPodNum": 3,
      "CpuSpec": 1,
      "MemorySpec": 2,
      "ImgRepo": "",
      "ImgName": "tem-100019051593-sfuw/app0925",
      "ImgVersion": "20241202165113",
      "EsInfo": null,
      "EnvConf": [
        {
          "Key": "TEM_APM_TOKEN",
          "Value": "DBaxEudGGxxOFmAYNBuq",
          "Type": "reserved",
          "Config": "",
          "Secret": ""
        },
        {
          "Key": "TEM_APM_COLLECTOR_SKYWALKING",
          "Value": "pl.ap-shanghai.apm.tencentcs.com:11800",
          "Type": "reserved",
          "Config": "",
          "Secret": ""
        },
        {
          "Key": "TEM_APM_COLLECTOR_JAEGER",
          "Value": "pl.ap-shanghai.apm.tencentcs.com:14250",
          "Type": "reserved",
          "Config": "",
          "Secret": ""
        },
        {
          "Key": "TEM_APM_COLLECTOR_OPENTELEMETRY",
```

```
"Value": "pl.ap-shanghai.apm.tencentcs.com:4317",
"Type": "reserved",
"Config": "",
"Secret": ""
},
{
  "Key": "TEM_APM_INSTANCE_ID",
  "Value": "apm-JcXK7eNzw",
  "Type": "reserved",
  "Config": "",
  "Secret": ""
},
{
  "Key": "PROMETHEUS_INSTANCE_ID",
  "Value": "prom-hdielllob",
  "Type": "reserved",
  "Config": "",
  "Secret": ""
},
{
  "Key": "PROMETHEUS_TOKEN",
  "Value": ";F1Lh7zVs-LFhT&+3dRqUA5bhd1",
  "Type": "reserved",
  "Config": "",
  "Secret": ""
},
{
  "Key": "PROMETHEUS_REMOTE_WRITE_URL",
  "Value": "http://10.0.30.147:9090/api/v1/prom/write",
  "Type": "reserved",
  "Config": "",
  "Secret": ""
},
{
  "Key": "PROMETHEUS_REMOTE_READ_URL",
  "Value": "http://10.0.30.147:9090/api/v1/read",
  "Type": "reserved",
  "Config": "",
  "Secret": ""
},
{
  "Key": "PROMETHEUS_PUSH_GATEWAY_ADDR",
  "Value": "10.0.30.147:9090",
  "Type": "reserved",
  "Config": "",
  "Secret": ""
},
{
  "Key": "PROMETHEUS_HTTP_API",
  "Value": "http://10.0.30.147:9090/api/v1",
  "Type": "reserved",
  "Config": "",
  "Secret": ""
},
{
  "Key": "PROMETHEUS_GRAFANA_ADDR",
  "Value": "",
```

```
"Type": "reserved",
"Config": "",
"Secret": ""
},
],
"SettingConfs": [
{
"ConfigDataName": "pk-test-1",
"SecretDataName": "",
"MountedPath": "/data",
"Data": [
{
"Key": "name",
"Value": "value",
"Type": "",
"Config": "",
"Secret": ""
}
]
}
],
"LogConfs": [],
"LogOutputConf": null,
"PostStart": "/bin/sh\n-c\ncurl localhost:8080/postStart",
"PostStartEncoded": "",
"PreStop": "/bin/sh\n-c\nwget --server-response --spider -S --no-verbose --header=\"Content-Type:application/json\" --post -data='{\"serverName\": \"api-basic\"}' \"http://localhost:9900/actuator/stopmsg\"\\nsleep 30s",
"PreStopEncoded": "",
"StorageConfs": [],
"StorageMountConfs": [],
"JvmOpts": "-Xms128m -XX:MetaspaceSize=128m",
"Service": {
"PortMappings": [],
"LoadBalanceId": "",
"SubnetId": "",
"Type": "",
"Name": "",
"ApplicationName": "app0925",
"ClusterIp": [],
"ExternalIp": "",
"ExternalDomain": "",
"VersionName": "",
"Ports": [],
"Yaml": "",
"ApplicationId": "app-5vaz8x85",
"ServicePortMappingList": [
{
"Type": "CLUSTER",
"ServiceName": "k8s-svc-test",
"ClusterIp": "172.16.56.122",
"ExternalIp": "",
"ExternalDomain": "",
"SubnetId": "",
"VpcId": "",
"LoadBalanceId": "",
"Yaml": "apiVersion: xxx",
"Ports": [
```

```
8080
],
"PortMappingItemList": [
{
"Port": 8080,
"TargetPort": 8080,
"Protocol": "TCP"
}
],
},
{
"Type": "CLUSTER",
"ServiceName": "pk-svc",
"ClusterIp": "172.16.52.35",
"ExternalIp": "",
"ExternalDomain": "",
"SubnetId": "",
"VpcId": "",
"LoadBalanceId": "",
"Yaml": "apiVersion: xxxx",
"Ports": [
8080
],
"PortMappingItemList": [
{
"Port": 8080,
"TargetPort": 8080,
"Protocol": "TCP"
}
]
},
],
"FlushAll": false,
"EnableRegistryNextDeploy": 0,
"AllIpDone": true
},
"Liveness": null,
"Readiness": null,
"StartupProbe": null,
"EsEnable": null,
"EnableTracing": 0,
"EnableRegistry": 0,
"EnableTracingReport": 0,
"EnableMetrics": 0,
"EnablePrometheusConf": null,
"ApmInstanceId": "",
"LogEnable": null,
"RepoType": 3,
"CreateDate": "2024-09-25 11:36:11",
"ModifyDate": "2024-12-20 17:00:16",
"MinAliveInstances": "",
"ImageTag": "20241202165113",
"SecurityGroupIds": [
"sg-1zwhplpi"
],
"EsStrategy": null,
"AutoscalerList": [
```

```
{
  "AutoscalerId": "scaler-3j2o6ypg",
  "AutoscalerName": "elastic-test",
  "MinReplicas": 0,
  "MaxReplicas": 4,
  "CreateDate": "2024-12-05 11:10:19",
  "ModifyDate": "2024-12-20 17:00:23",
  "EnableDate": "2024-12-20 17:00:23",
  "Description": "",
  "Enabled": true,
  "HorizontalAutoscaler": [
    {
      "MinReplicas": 0,
      "MaxReplicas": 4,
      "Metrics": "CPU",
      "Threshold": 0,
      "DoubleThreshold": 60,
      "Enabled": true
    }
  ],
  "CronHorizontalAutoscaler": []
},
{
  "AutoscalerId": "scaler-b5q33njo",
  "AutoscalerName": "pk-test-clb",
  "MinReplicas": 0,
  "MaxReplicas": 4,
  "CreateDate": "2024-10-25 16:36:10",
  "ModifyDate": "2024-11-29 10:35:52",
  "EnableDate": "2024-11-29 10:35:52",
  "Description": "",
  "Enabled": false,
  "HorizontalAutoscaler": [
    {
      "MinReplicas": 0,
      "MaxReplicas": 4,
      "Metrics": "CPU",
      "Threshold": 0,
      "DoubleThreshold": 50,
      "Enabled": true
    }
  ],
  "CronHorizontalAutoscaler": [
    {
      "Name": "pk-test",
      "Period": "* * * *",
      "Schedules": [
        {
          "StartAt": "02:00",
          "TargetReplicas": 1
        }
      ],
      "Enabled": false,
      "Priority": 0
    }
  ]
},
},
```

```
{
  "AutoscalerId": "scaler-ojr3nlj9",
  "AutoscalerName": "pk-test-1",
  "MinReplicas": 1,
  "MaxReplicas": 3,
  "CreateDate": "2024-12-19 23:38:45",
  "ModifyDate": "2024-12-19 23:38:45",
  "EnableDate": null,
  "Description": "",
  "Enabled": false,
  "HorizontalAutoscaler": [
    {
      "MinReplicas": 1,
      "MaxReplicas": 3,
      "Metrics": "CPU",
      "Threshold": 0,
      "DoubleThreshold": 50,
      "Enabled": true
    }
  ],
  "CronHorizontalAutoscaler": [
    {
      "Name": "policy-test",
      "Period": "* * * *",
      "Schedules": [
        {
          "StartAt": "00:00",
          "TargetReplicas": 1
        }
      ],
      "Enabled": true,
      "Priority": 0
    }
  ],
  "ApplicationName": "app0925",
  "ApplicationDescription": "preStop-test",
  "EnvironmentName": "tt",
  "EnvironmentId": "en-dpxydz5",
  "RepoServer": "tem-sh-custom-registry-prd-01.tencentcloudcr.com",
  "PublicDomain": "",
  "EnablePublicAccess": null,
  "Modifier": "100019051593",
  "Creator": "100019051593",
  "CurrentInstances": 1,
  "ExpectedInstances": 3,
  "Status": "abnormal",
  "UnderDeploying": true,
  "BatchDeployStatus": "",
  "CodingLanguage": "JAVA",
  "ImageType": 1,
  "ImageCommand": "",
  "ImageArgs": null,
  "UseRegistryDefaultConfig": false,
  "DeployStrategyConf": null,
  "Zones": [
```

```
"ap-shanghai-2"
],
"LastDeployDate": "2024-12-20 17:00:17",
"LastDeploySuccessDate": "2024-12-20 17:00:17",
"NodeInfos": [
{
"Name": "eklet-subnet-kctzw7nj-711231",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-kctzw7nj",
"AvailableIpCount": "243",
"Cidr": "10.0.10.0-24"
},
{
"Name": "eklet-subnet-8indzsuv-457482",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-8indzsuv",
"AvailableIpCount": "226",
"Cidr": "10.0.30.0-24"
},
{
"Name": "eklet-subnet-qau65gm1-638453",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-qau65gm1",
"AvailableIpCount": "246",
"Cidr": "10.0.20.0-24"
}
],
"WorkloadInfo": {
"ApplicationName": "app0925",
"ClusterId": "cls-kotuimb9",
"VersionName": "20241202165113",
"ReadyReplicas": 1,
"Replicas": 3,
"UpdatedReplicas": 2,
"UpdatedReadyReplicas": 0,
"UpdateRevision": "app0925-574b65c5f7",
"CurrentRevision": "app0925-77969b5b49"
},
"PodList": {
"Offset": 0,
"Limit": 3,
"TotalCount": 3,
"PodList": [
{
"Zone": "ap-shanghai-2",
"NodeInfo": {
"Name": "eklet-subnet-kctzw7nj-711231",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-kctzw7nj",
"AvailableIpCount": "243",
"Cidr": "10.0.10.0-24"
},
"DeployVersion": "",
"RestartCount": 0,
"Ready": false,
"StartTime": "2024-12-20T09:01:01Z",
"ContainerState": "waiting",
```

```
"Unhealthy": null,
"UnhealthyWarningMsg": "",
"VersionId": "revision-jorxnl6j",
"ApplicationName": "app0925",
"Status": "CrashLoopBackOff",
"CreateTime": "2024-12-20T09:00:23Z",
"PodIp": "10.0.10.220",
"Webshell": "https://tkecache.cloud.tencent.com/xxxx",
"PodId": "app0925-4sbct"
},
{
"Zone": "ap-shanghai-2",
"NodeInfo": {
"Name": "eklet-subnet-qau65gm1-638453",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-qau65gm1",
"AvailableIpCount": "246",
"Cidr": "10.0.20.0-24"
},
"DeployVersion": "",
"RestartCount": 0,
"Ready": false,
"StartTime": "2024-12-20T09:01:11Z",
"ContainerState": "waiting",
"Unhealthy": null,
"UnhealthyWarningMsg": "",
"VersionId": "revision-jorxnl6j",
"ApplicationName": "app0925",
"Status": "CrashLoopBackOff",
"CreateTime": "2024-12-20T09:00:23Z",
"PodIp": "10.0.20.37",
"Webshell": "https://tkecache.cloud.tencent.com/xxxx",
"PodId": "app0925-s4vpj"
},
{
"Zone": "ap-shanghai-2",
"NodeInfo": {
"Name": "eklet-subnet-8indzsuv-457482",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-8indzsuv",
"AvailableIpCount": "226",
"Cidr": "10.0.30.0-24"
},
"DeployVersion": "",
"RestartCount": 0,
"Ready": true,
"StartTime": "2024-12-19T03:12:44Z",
"ContainerState": "running",
"Unhealthy": null,
"UnhealthyWarningMsg": "",
"VersionId": "revision-57moqvm5",
"ApplicationName": "app0925",
"Status": "Running",
"CreateTime": "2024-12-19T03:11:32Z",
"PodIp": "10.0.30.158",
"Webshell": "https://tkecache.cloud.tencent.com/qcloud/xxxx",
"PodId": "app0925-jt5bn"
```

```

}
],
"RequestId": "csdse233-xxx-xxx-xxx"
},
"ConfEdited": null,
"SpeedUp": true,
"OsFlavour": "ALPINE",
"StoppedManually": false,
"TcrInstanceId": "tcr-d8g5il2x",
"Tags": [],
"PkgName": "tem/pkg/1305738457/app-5vaz8x85/1733129495394/K8sEventDemo-1.0.jar",
"HorizontalAutoscaler": [],
"CronHorizontalAutoscaler": []
}
}
}

```

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.ActionReadTimeout	请求响应超时。
FailedOperation.DefaultInternalServerError	服务器繁忙,请稍后再试。
FailedOperation.DescribeRunPodListError	查询实例信息失败。
FailedOperation.DescribeServiceError	查询service失败。
InternalServerError.ActionReadTimeout	请求响应超时。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DescribeRunPodListError	查询实例信息失败。
InternalServerError.DescribeServiceError	查询service失败。

错误码	描述
InvalidParameterValue.ServiceNotBelongToAppid	应用不属于此账户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
MissingParameter.ServiceIdNull	服务ID不能为空。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
ResourceNotFound.VersionServiceNotFound	找不到版本对应的应用。
UnauthorizedOperation.UnauthorizedOperation	未授权。

获取应用实例列表

最近更新时间：2025-04-27 21:16:38

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

获取应用实例列表

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：DescribeApplicationPods。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境id 示例值：en-xxxxxx
ApplicationId	是	String	应用id 示例值：app-xxxxxx
Limit	否	Integer	单页条数，默认值20 示例值：10
Offset	否	Integer	分页下标，默认值0 示例值：1
Status	否	String	实例状态 - Running - Pending - Error 示例值：Running
PodName	否	String	实例名字 示例值：pod-a
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	DescribeRunPodPage	返回结果 示例值：page
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 获取应用实例列表

获取应用实例列表

输入示例

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

<公共请求参数>

```
{
  "Status": "Running",
  "EnvironmentId": "en-xxxxxx",
  "Offset": 0,
  "SourceChannel": 0,
  "Limit": 20,
  "PodName": "pod-name-xxx",
  "ApplicationId": "app-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "0f09d0d3-3414-48ba-b8cc-e1c543b1fc1a",
    "Result": {
      "Offset": 0,
      "Limit": 3,
      "TotalCount": 3,
      "PodList": [
        {
          "Zone": "ap-shanghai-2",
          "NodeInfo": {
            "Name": "eklet-subnet-kctzw7nj-711231",
            "Zone": "ap-shanghai-2",
            "SubnetId": "subnet-kctzw7nj",
            "AvailableIpCount": "243",
            "Cidr": "10.0.10.0-24"
          },
          "DeployVersion": "20241202165113",
          "RestartCount": 0,
          "Ready": false,
          "StartTime": "2024-12-20T09:01:01Z",
          "ContainerState": "waiting",
          "Unhealthy": null,
          "UnhealthyWarningMsg": "",
          "VersionId": "revision-jorxnl6j",
          "ApplicationName": "app0925",
          "Status": "CrashLoopBackOff",
          "CreateTime": "2024-12-20 17:00:23",
          "Webshell": "https://tkecache.cloud.tencent.com/xxx",
          "PodId": "app0925-4sbct",
          "PodIp": "10.0.10.220"
        },
        {
          "Zone": "ap-shanghai-2",
          "NodeInfo": {
```

```
"Name": "eklet-subnet-qau65gm1-638453",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-qau65gm1",
"AvailableIpCount": "246",
"Cidr": "10.0.20.0-24"
},
"DeployVersion": "20241202165113",
"RestartCount": 0,
"Ready": false,
"StartTime": "2024-12-20T09:01:11Z",
"ContainerState": "waiting",
"Unhealthy": null,
"UnhealthyWarningMsg": "",
"VersionId": "revision-jorxnl6j",
"ApplicationName": "app0925",
"Status": "CrashLoopBackOff",
"CreateTime": "2024-12-20 17:00:23",
"Webshell": "https://tkecache.cloud.tencent.com/xxx",
"PodId": "app0925-s4vpj",
"PodIp": "10.0.20.37"
},
{
"Zone": "ap-shanghai-2",
"NodeInfo": {
"Name": "eklet-subnet-8indzsuv-457482",
"Zone": "ap-shanghai-2",
"SubnetId": "subnet-8indzsuv",
"AvailableIpCount": "226",
"Cidr": "10.0.30.0-24"
},
"DeployVersion": "20241202165113",
"RestartCount": 0,
"Ready": true,
"StartTime": "2024-12-19T03:12:44Z",
"ContainerState": "running",
"Unhealthy": null,
"UnhealthyWarningMsg": "",
"VersionId": "revision-57moqvm5",
"ApplicationName": "app0925",
"Status": "Running",
"CreateTime": "2024-12-19 11:11:32",
"Webshell": "https://tkecache.cloud.tencent.com/xxx",
"PodId": "app0925-jt5bn",
"PodIp": "10.0.30.158"
}
],
"RequestId": "0f09d0d3-xxx-xxx-xxx"
}
}
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
FailedOperation.DescribeRunPodListError	查询实例信息失败。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DescribeRunPodListError	查询实例信息失败。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

重启应用实例

最近更新时间：2025-04-27 21:16:38

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

重启应用实例

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：RestartApplicationPod。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境id 示例值：en-xxxxxx
ApplicationId	是	String	应用id 示例值：app-xxxxxx
PodName	是	String	名字 示例值：pod-a
Limit	否	Integer	单页条数 示例值：10
Offset	否	Integer	分页下标 示例值：1
Status	否	String	pod状态 示例值：Running
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 重启应用实例

重启应用实例

输入示例

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

<公共请求参数>

```
{
  "Status": "Running",
  "EnvironmentId": "en-xxxxxx",
  "Offset": 0,
  "SourceChannel": 0,
  "Limit": 0,
  "PodName": "pod-name-xxx",
  "ApplicationId": "app-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。

错误码	描述
UnauthorizedOperation.UnauthorizedOperation	未授权。

更新应用部署版本

最近更新时间：2025-04-27 21:16:38

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

更新应用部署版本

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：RollingUpdateApplicationByVersion。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	应用ID 示例值：app-xxxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxxx
DeployVersion	是	String	更新版本，IMAGE 部署为 tag 值；JAR/WAR 部署 为 Version 示例值：v1
PackageName	否	String	JAR/WAR 包名，仅 JAR/WAR 部署时必须填 示例值：package-name-xxx.jar
From	否	String	请求来源平台，含 IntelliJ，Coding 示例值：IntelliJ
DeployStrategyType	否	String	部署策略，AUTO 为全自动；BETA 为小批量验证后自动；MANUAL 为全手动； 示例值：AUTO
TotalBatchCount	否	Integer	发布批次数 示例值：2
BatchInterval	否	Integer	批次间隔时间 示例值：60
BetaBatchNum	否	Integer	小批量验证批次的实例数 示例值：1
MinAvailable	否	Integer	发布过程中保障的最小可用实例数 示例值：1
Force	否	Boolean	是否强制发布 示例值：force

3. 输出参数

参数名称	类型	描述
Result	String	版本ID 示例值：revision-xxx
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 应用部署

应用部署

输入示例

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

{
  "ApplicationId": "app-xxxxxx",
  "EnvironmentId": "en-xxxxxx",
  "DeployVersion": "v1",
  "PackageName": "test.jar",
  "From": "IntelliJ",
  "DeployStrategyType": "0",
  "TotalBatchCount": 0,
  "BatchInterval": 0,
  "BetaBatchNum": 0,
  "MinAvailable": 0,
  "Force": true
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": "revision-xxx"
  }
}
```

示例2 部署成功

部署成功

输入示例

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

{
  "EnvironmentId": "env-qgnzgy8r",
  "TotalBatchCount": "2",
  "PackageName": "tem/pkg/1300555551/service-gn54vdy6/1641528220867/hello-world-0.0.1-SNAPSHOT.jar",
  "DeployStrategyType": "AUTO",
  "DeployVersion": "20220107122452",
  "BatchInterval": "60",
  "ApplicationId": "service-gn54vdy6",
  "BetaBatchNum": "1"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "28fe46b2-dbd9-4ee6-b388-603acb6c9263",
    "Result": ""
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DeployVersionError	版本部署调用失败。
InvalidParameterValue.ApmNotBind	APM 没有与当前环境绑定。
InvalidParameterValue.InvalidDeployVersion	版本号格式非法。
InvalidParameterValue.InvalidEnvName	环境变量名非法，要求有由字母、数字，"."、"_" 和 "-" 组成，不能由数字开头。
InvalidParameterValue.InvalidMountPath	挂载路径不合法，不能为 /app。
InvalidParameterValue.JdkVersionRequired	JDK 版本不能为空。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.OsNotSupport	操作系统不支持。
InvalidParameterValue.RegistryNotBind	注册中心没有与当前环境绑定。
InvalidParameterValue.ServiceNotBelongToAppid	应用不属于此账户。
InvalidParameterValue.ServicePodReachMaximum	实例创建失败，达到上限。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。

错误码	描述
InvalidParameterValue.VersionLengthLimit	version 不能超过128位。
InvalidParameterValue.VersionLowerCase	版本必须小写。
MissingParameter.DeployModeNull	部署方式不能为空。
MissingParameter.DeployVersionNull	部署版本不能为空。
MissingParameter.LogsetOrTopicNull	logset 和 topic 不能为空。
MissingParameter.NamespaceIdNull	环境ID不能为空。
MissingParameter.SvcRepoNotReady	镜像仓库还未就绪。
ResourceInUse.ServiceDeploying	服务部署中。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
ResourceUnavailable.WaitForKruise	等待组件安装。
UnauthorizedOperation.MissingEksLogRole	缺少容器服务的 CLS 日志角色，请打开控制台授权。
UnauthorizedOperation.UnauthorizedOperation	未授权。

访问方式相关接口

新增访问方式

最近更新时间：2025-04-25 02:00:10

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

新增访问方式

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：CreateApplicationService。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
Service	是	ServicePortMapping	访问方式详情 示例值：{"ServiceName":"k8s-svc-test","PortMappingItemList":[{"Port":8080,"TargetPort":8080,"Protocol":"TCP"}],"Type":"CLUSTER"}
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 新增访问方式

输入示例

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

<公共请求参数>

```
{
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0,
  "EnvironmentId": "en-xxxxxx",
  "Service": {
    "Type": "VPC",
    "ServiceName": "svc-name-xxx",
    "ClusterIp": "10.xx.xx.xx",
    "ExternalIp": "10.xx.xx.xx",
    "SubnetId": "subnet-xxxxxx",
    "VpcId": "vpc-xxxxxx",
    "LoadBalanceId": "10.xx.xx.xx",
    "Yaml": "yaml-apiVersion-xxx",
    "Ports": [
      0
    ],
    "PortMappingItemList": [
      {
        "Port": 8080,
        "TargetPort": 8080,
        "Protocol": "TCP"
      }
    ],
    "ExternalDomain": "ext.domain.xxx"
  }
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "xx-xxx-xxx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DescribeServiceListError	查询service列表失败。
InvalidParameter.ApplicationAccessServiceReachMaximum	应用访问配置服务数量达到上限。
InvalidParameter.LBServiceCannotSupportTcpUdpSameTime	LB类型服务不能同时支持TCP和UDP。
InvalidParameter.MustProvidePortMappingRules	必须提供PortMapping规则。
InvalidParameter.ServiceNameNotValid	服务名不符合规范。
InvalidParameter.ServiceUseReserveSuffix	服务名使用了保留后缀。
InvalidParameter.TooManyPortMappingRules	PortMapping规则数量超过限制。
InvalidParameterValue.ApplicationAccessServiceReachMaximum	应用访问配置服务数量达到上限。
InvalidParameterValue.ApplicationServiceAlreadyExist	已存在同名访问方式。
InvalidParameterValue.ApplicationServiceNotFound	没有找到该访问方式。
InvalidParameterValue.CannotOverWriteOtherApplicationService	不能覆盖其他应用的访问方式。
InvalidParameterValue.CannotUpdateServiceByBothMethods	不能同时使用全量以及单条更新。
InvalidParameterValue.InvalidEksServiceType	访问类型不支持。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.PortDuplicateError	服务端口映射重复。
InvalidParameterValue.PortIsReserved	服务端口是保留端口。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnsupportedOperation.UnsupportAction	不支持的ACTION。

删除一条访问方式

最近更新时间：2025-04-25 02:00:09

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

删除一条访问方式

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：DeleteApplicationService。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
ServiceName	是	String	访问方式服务名 示例值：name-xxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 删除一条访问方式

输入示例

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

{
```

```
"ApplicationId": "app-xxxxxx",
"SourceChannel": 0,
"EnvironmentId": "en-xxxxxx",
"ServiceName": "abc-xxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "xx-xxx-xxxx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.ApplicationServiceNotFound	没有找到该访问方式。
InvalidParameterValue.CannotUpdateServiceByBothMethods	不能同时使用全量以及单条更新。
InvalidParameterValue.InvalidEksServiceType	访问类型不支持。
InvalidParameterValue.PortIsReserved	服务端口是保留端口。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
UnauthorizedOperation.UnauthorizedOperation	未授权。

查询应用访问方式列表

最近更新时间：2025-04-25 02:00:09

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询应用访问方式列表

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：DescribeApplicationServiceList。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	namespace id 示例值：en-xxxxxx
ApplicationId	是	String	服务ID 示例值：app-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	EksService	应用 EKS Service 列表 示例值：ss
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 查询应用访问方式列表

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
  "ApplicationId": "app-xxxxxx",
```

```
"SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "7003cb9f-0932-4cba-b15a-21788df3a8fb",
    "Result": {
      "PortMappings": [],
      "LoadBalanceId": "",
      "SubnetId": "",
      "Type": "",
      "Name": "",
      "ApplicationName": "app0925",
      "ClusterIp": [],
      "ExternalIp": "",
      "ExternalDomain": "",
      "VersionName": "",
      "Ports": [],
      "Yaml": "",
      "ApplicationId": "app-5vaz8x85",
      "ServicePortMappingList": [
        {
          "Type": "CLUSTER",
          "ServiceName": "k8s-svc-test",
          "ClusterIp": "172.16.56.122",
          "ExternalIp": "",
          "ExternalDomain": "",
          "SubnetId": "",
          "VpcId": "",
          "LoadBalanceId": "",
          "Yaml": "apiVersion: xxx",
          "Ports": [
            8080
          ],
          "PortMappingItemList": [
            {
              "Port": 8080,
              "TargetPort": 8080,
              "Protocol": "TCP"
            }
          ]
        },
        {
          "Type": "CLUSTER",
          "ServiceName": "pk-svc",
          "ClusterIp": "172.16.52.35",
          "ExternalIp": "",
          "ExternalDomain": "",
          "SubnetId": "",
          "VpcId": "",
          "LoadBalanceId": "",
          "Yaml": "apiVersion: xxx",
          "Ports": [
            8080
          ]
        }
      ]
    }
  }
}
```

```

],
"PortMappingItemList": [
{
"Port": 8080,
"TargetPort": 8080,
"Protocol": "TCP"
}
]
}
],
"FlushAll": false,
"EnableRegistryNextDeploy": 0,
"AllIpDone": true
}
}
}

```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.InterfaceNotFound	未提供该接口。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。
UnsupportedOperation.UnsupportAction	不支持的ACTION。

修改服务访问方式列表

最近更新时间：2025-04-25 02:00:09

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

修改服务访问方式列表

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：ModifyApplicationService。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0
Service	否	EksService	全量访问方式设置
Data	否	ServicePortMapping	单条访问方式设置

3. 输出参数

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

4. 示例

示例1 修改服务访问方式列表

输入示例

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

```
{
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0,
  "EnvironmentId": "en-xxxxxx",
  "Service": {
    "Name": "svc-name-xxx",
    "Ports": [
      8080
    ],
    "Yaml": "apiVersion: xxx",
    "ApplicationName": "app-name-xxx",
    "VersionName": "ver-name-xxx",
    "ClusterIp": [
      "11.xx.xx.xx"
    ],
    "ExternalIp": "10.xx.xx.xx",
    "Type": "CLUSTER",
    "SubnetId": "subnet-xxx",
    "LoadBalanceId": "lb-xxxxxx",
    "PortMappings": [
      {
        "Port": 8080,
        "TargetPort": 8080,
        "Protocol": "TCP",
        "ServiceName": "svc-name-xxx"
      }
    ],
    "ServicePortMappingList": [
      {
        "Type": "CLUSTER",
        "ServiceName": "svc-name-xxx",
        "ClusterIp": "11.xx.xx.xx",
        "ExternalIp": "10.xx.xx.xx",
        "SubnetId": "subnet-xxxxxx",
        "VpcId": "vpc-xxxxxx",
        "LoadBalanceId": "lb-xxxxxx",
        "Yaml": "apiVersion: xxx",
        "Ports": [
          8080
        ],
        "PortMappingItemList": [
          {
            "Port": 8080,
            "TargetPort": 8080,
            "Protocol": "TCP"
          }
        ],
        "ExternalDomain": "ext.com.xxx"
      }
    ],
    "FlushAll": true,
    "EnableRegistryNextDeploy": 0,
    "ApplicationId": "app-xxxxxx",
    "AllIpDone": true,
    "ExternalDomain": "ext.com.xxx"
  },
  "Data": {
```

```
"Type": "CLUSTER",
"ServiceName": "svc-name-xxx",
"ClusterIp": "11.xx.xx.xx",
"ExternalIp": "10.xx.xx.xx",
"SubnetId": "subnet-xxxxxxx",
"VpcId": "vpc-xxxxxxx",
"LoadBalanceId": "lb-xxxxxxx",
"Yaml": "apiVersion: xxx",
"Ports": [
  8080
],
"PortMappingItemList": [
  {
    "Port": 8080,
    "TargetPort": 8080,
    "Protocol": "TCP"
  }
],
"ExternalDomain": "ext.com.xxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "asdc122-xxx-xxx-xxx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameter.ApplicationAccessServiceReachMaximum	应用访问配置服务数量达到上限。
InvalidParameter.LBServiceCannotSupportTcpUdpSameTime	LB类型服务不能同时支持TCP和UDP。
InvalidParameter.MustProvidePortMappingRules	必须提供PortMapping规则。
InvalidParameter.ServiceNameNotValid	服务名不符合规范。
InvalidParameter.ServiceUseReserveSuffix	服务名使用了保留后缀。
InvalidParameter.TooManyPortMappingRules	PortMapping规则数量超过限制。
InvalidParameterValue.ApplicationServiceNotFound	没有找到该访问方式。
InvalidParameterValue.InvalidEksServiceType	访问类型不支持。
InvalidParameterValue.PortIsReserved	服务端口是保留端口。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。

配置文件相关接口

创建配置

最近更新时间：2025-04-25 02:00:09

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

创建配置

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：CreateConfigData。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：abc-xxxx
Data.N	是	Array of Pair	配置信息 示例值：[{"Key":"key-xxx","Value":"val-xxx"}]
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 创建配置

创建配置

输入示例

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

<公共请求参数>

```
{
  "EnvironmentId": "en-xxxxxx",
  "Name": "config-name-xxx",
  "SourceChannel": 0,
  "Data": [
    {
      "Key": "key-xxx",
      "Value": "val-xxx"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.CreateConfigDataError	创建配置失败。
InvalidParameterValue.ConfigDataAlreadyExist	配置已存在。
InvalidParameterValue.ConfigDataInvalid	配置不合法。

错误码	描述
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

查询配置详情

最近更新时间：2025-04-25 02:00:09

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询配置详情

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：DescribeConfigData。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：abc-xxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	ConfigData	配置 示例值：{"Name":"config-test","CreateTime":"2024-12-19 23:55:05","RelatedApplications":[],"Data":[{"Key":"key-xxx","Value":"val-xxx","Type":"","Config":"","Secret":""}]}
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 查询配置详情

查询配置详情

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
  "Name": "abc-xxxx",
```

```
"SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8b94500c-06cf-42cf-a228-dfea478c3239",
    "Result": {
      "Name": "config-test",
      "CreateTime": "2024-12-19 23:55:05",
      "RelatedApplications": [],
      "Data": [
        {
          "Key": "key-xxx",
          "Value": "val-xxx",
          "Type": "",
          "Config": "",
          "Secret": ""
        }
      ]
    }
  }
}
```

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.DescribeConfigDataError	查询配置详情失败。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。

错误码	描述
ResourceNotFound.ConfigDataNotFound	配置不存在。
UnauthorizedOperation.UnauthorizedOperation	未授权。

查询配置列表

最近更新时间：2025-04-25 02:00:09

-

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询配置列表

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：DescribeConfigDataList。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0
ContinueToken	否	String	查询游标 示例值：token-xxxxxx
Limit	否	Integer	分页 limit 示例值：10

3. 输出参数

参数名称	类型	描述
Result	DescribeConfigDataListPage	配置列表
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 查询配置列表

查询配置列表

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
  "ContinueToken": "token-xxxxxx",
  "Limit": 0,
```

```
"SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8b94500c-06cf-42cf-a228-dfea478c3239",
    "Result": {
      "Records": [
        {
          "Name": "config-test",
          "CreateTime": "2024-12-19 23:55:05",
          "RelatedApplications": [],
          "Data": [
            {
              "Key": "key-xxx",
              "Value": "val-xxx",
              "Type": "",
              "Config": "",
              "Secret": ""
            }
          ]
        },
        {
          "Name": "pk-test",
          "CreateTime": "2024-12-19 11:04:13",
          "RelatedApplications": [],
          "Data": [
            {
              "Key": "name",
              "Value": "pk",
              "Type": "",
              "Config": "",
              "Secret": ""
            }
          ]
        }
      ],
      "RemainingCount": null,
      "ContinueToken": ""
    }
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DescribeConfigDataListError	查询配置列表失败。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

销毁配置

最近更新时间：2025-04-25 02:00:09

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

销毁配置

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：DestroyConfigData。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：abc-xxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 销毁配置

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
  "Name": "abc-xx",
```

```
"SourceChannel": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

编辑配置

最近更新时间：2025-04-25 02:00:08

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

编辑配置

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：ModifyConfigData。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：abc-xxxx
Data.N	是	Array of Pair	配置信息 示例值：[{"Key":"key-xxx","Value":"val-xxx","Type":"","Config":"","Secret":""}]
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 编辑配置

编辑配置

输入示例

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

```
{
  "EnvironmentId": "en-xxxxxx",
  "Name": "config-test",
  "SourceChannel": 0,
  "Data": [
    {
      "Key": "key-xxx",
      "Value": "val-xxx",
      "Type": "",
      "Config": "",
      "Secret": ""
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.ModifyConfigDataError	修改配置失败。
InvalidParameterValue.ConfigDataInvalid	配置不合法。
UnauthorizedOperation.UnauthorizedOperation	未授权。

日志采集相关接口

创建日志收集配置

最近更新时间：2025-04-25 02:00:13

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

创建日志收集配置

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：CreateLogConfig。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：abc-xxx
InputType	是	String	收集类型，container_stdout 为标准输出；container_file 为文件； 示例值：container_file
ApplicationId	是	String	应用 ID 示例值：app-xxxxxx
LogsetId	否	String	日志集 ID 示例值：abc-xxxxxx
TopicId	否	String	日志主题 ID 示例值：abc-xxxxxx
LogType	否	String	日志提取模式，minimalist_log 为单行全文；multiline_log 为多行全文；json_log 为 json 格式；fullregex_log 为单行正则；multiline_fullregex_log 为多行正则 示例值：minimalist_log
BeginningRegex	否	String	首行正则表达式，当LogType=multiline_log 时生效 示例值：/abc/
LogPath	否	String	收集文件目录，当 InputType=container_file 时生效 示例值：/abc
FilePattern	否	String	收集文件名模式，当 InputType=container_file 时生效 示例值：app_*.log
ExtractRule	否	LogConfigExtractRule	导出规则 示例值：{"FilterKeys":[],"FilterRegex": [],"UnMatchUpload":"true","UnMatchedKey":"LogParseFailure"}

3. 输出参数

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

4. 示例

示例1 创建日志收集配置

创建日志收集配置

输入示例

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

{
  "EnvironmentId": "en-dpxydz5",
  "ApplicationId": "app-5vaz8x85",
  "LogsetId": "7a126551-62d1-4e12-b426-84bd5b2fdbca",
  "TopicId": "6c69fdb7-5e4c-4e52-8437-32c2c729b85a",
  "InputType": "container_stdout",
  "LogType": "json_log",
  "Name": "log-cfg-xxx",
  "ExtractRule": {
    "FilterKeys": [],
    "FilterRegex": [],
    "UnMatchUpload": "true",
    "UnMatchedKey": "LogParseFailure"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalError.CreateConfigDataError	创建配置失败。
InternalError.CreateLogConfigError	创建 LogConfig 错误。
InternalError.DefaultInternalError	服务器繁忙,请稍后再试。
InvalidParameterValue.ConfigDataAlreadyExist	配置已存在。
InvalidParameterValue.LogConfigAlreadyExist	LogConfig 配置已存在。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

查询日志收集配置详情

最近更新时间：2025-04-25 02:00:13

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询日志收集配置详情

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：DescribeLogConfig。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：abc-xxx
ApplicationId	是	String	应用 ID 示例值：app-xxx

3. 输出参数

参数名称	类型	描述
Result	LogConfig	配置
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 查询日志收集配置详情

查询日志收集配置详情

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
  "Name": "log-name-xxx",
```

```
"ApplicationId": "app-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e6fedfb8-1bc9-414d-b69b-5e24fdda6416",
    "Result": {
      "Name": "log-name-xxx",
      "ApplicationName": "app-name-xxx",
      "ApplicationId": "",
      "InputType": "container_stdout",
      "TopicId": "3ae33b64-d3d1-4a2c-96ff-a2e044dcexxx",
      "LogsetId": "5bf3beeb-1f64-40f9-9672-b596248d7xxx",
      "LogType": "minimalist_log",
      "BeginningRegex": "",
      "ExtractRule": {
        "Backtracking": "-1",
        "BeginningRegex": "",
        "Delimiter": "",
        "FilterKeys": [],
        "FilterRegex": [],
        "Keys": [],
        "LogRegex": "",
        "TimeFormat": "",
        "TimeKey": "",
        "UnMatchUpload": "false",
        "UnMatchedKey": ""
      },
      "LogPath": "",
      "FilePattern": "",
      "CreateDate": "2022-05-28T18:02:55.000+08:00",
      "ModifyDate": "2022-05-28T18:02:56.000+08:00"
    }
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DescribeLogConfigError	查询 LogConfig 错误。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.LogConfigNotFound	日志配置不存在。
ResourceNotFound.ServiceNotFound	找不到应用。
UnauthorizedOperation.UnauthorizedOperation	未授权。

查询分页的日志收集配置列表

最近更新时间：2025-04-25 02:00:13

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

查询分页的日志收集配置列表

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：DescribePagedLogConfigList。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxx
ApplicationId	否	String	应用 ID 示例值：app-xxx
ApplicationName	否	String	应用名 示例值：app-name-xxx
Name	否	String	规则名 示例值：rule-name
Limit	否	Integer	分页大小，默认 20 示例值：20
ContinueToken	否	String	翻页游标 示例值：some-token-value

3. 输出参数

参数名称	类型	描述
Result	LogConfigListPage	日志收集配置列表
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 查询分页的日志收集配置列表

查询分页的日志收集配置列表

输入示例

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

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

```
{
  "EnvironmentId": "en-xxxxxx",
  "ApplicationId": "app-xxxxxx",
  "ApplicationName": "name-xxx",
  "Name": "name-xxx",
  "Limit": 0,
  "ContinueToken": "some-token-value"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "0793bb3c-6de8-4816-9ab0-5f006cF7656c",
    "Result": {
      "ContinueToken": "",
      "Records": [
        {
          "ApplicationId": "app-5vaz8x85",
          "ApplicationName": "app0925",
          "BeginningRegex": "",
          "CreateDate": "2024-10-31T15:25:07.000+08:00",
          "ExtractRule": {
            "Backtracking": "",
            "BeginningRegex": "",
            "Delimiter": "",
            "FilterKeys": [],
            "FilterRegex": [],
            "Keys": [],
            "LogRegex": "",
            "TimeFormat": "",
            "TimeKey": "",
            "UnMatchUpload": "true",
            "UnMatchedKey": "LogParseFailure"
          },
          "FilePattern": "",
          "InputType": "container_stdout",
          "LogPath": "",
          "LogType": "",
          "LogsetId": "7a126551-62d1-4e12-b426-84bd5b2fdxxx",
          "ModifyDate": "2024-10-31T15:25:08.000+08:00",
          "Name": "log-cfg-name",
          "TopicId": "6c69fdb7-5e4c-4e52-8437-32c2c729b85a"
        }
      ]
    }
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.DescribeConfigDataListError	查询配置列表失败。
InternalServerError.DescribeLogConfigListError	查询 LogConfig 列表错误。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

销毁日志收集配置

最近更新时间：2025-04-25 02:00:12

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

销毁日志收集配置

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：DestroyLogConfig。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：name-xxx
ApplicationId	否	String	应用 ID 示例值：app-xxxxxx

3. 输出参数

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

4. 示例

示例1 销毁日志收集配置

销毁日志收集配置

输入示例

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

{
  "EnvironmentId": "en-xxxxxx",
```

```
"Name": "name-xxx",
"ApplicationId": "app-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.DeleteLogConfigError	删除 LogConfig 错误。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

编辑日志收集配置

最近更新时间：2025-04-25 02:00:12

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

编辑日志收集配置

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：ModifyLogConfig。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境 ID 示例值：en-xxxxxx
Name	是	String	配置名 示例值：abc-xxxx
Data	是	LogConfig	日志收集配置信息
ApplicationId	是	String	应用 ID 示例值：app-xxxxxx

3. 输出参数

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

4. 示例

示例1 编辑日志收集配置

编辑日志收集配置

输入示例

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

{
```

```
"EnvironmentId": "en-xxxxxx",
"Name": "name-xxx",
"Data": {
  "LogsetId": "7a126551-62d1-4e12-b426-84bd5b2fdbca",
  "TopicId": "6c69fdb7-5e4c-4e52-8437-32c2c729b85a",
  "InputType": "container_stdout",
  "LogType": "json_log",
  "ExtractRule": {
    "FilterKeys": [],
    "FilterRegex": [],
    "UnMatchUpload": "true",
    "UnMatchedKey": "LogParseFailure"
  }
},
"ApplicationId": "app-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。

错误码	描述
InternalError.ModifyLogConfigError	修改 LogConfig 错误。
ResourceNotFound.ServiceNotFound	找不到应用。
UnauthorizedOperation.UnauthorizedOperation	未授权。

弹性规则相关接口

创建弹性伸缩策略组合

最近更新时间：2025-04-27 21:16:37

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

创建弹性伸缩策略组合

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：CreateApplicationAutoscaler。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxxx
Autoscaler	是	Autoscaler	弹性伸缩策略
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	String	弹性伸缩策略组合ID 示例值：scaler-xxxxxxx
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 创建弹性伸缩策略组合

创建弹性伸缩策略组合

输入示例

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

{
```

```
"ApplicationId": "app-xxxxxx",
"SourceChannel": 0,
"EnvironmentId": "en-xxxxxx",
"Autoscaler": {
  "MinReplicas": 1,
  "MaxReplicas": 3,
  "AutoscalerName": "pk-test-1",
  "Description": "这是一个描述",
  "Enabled": false,
  "HorizontalAutoscaler": [
    {
      "MinReplicas": 1,
      "MaxReplicas": 3,
      "Metrics": "CPU",
      "DoubleThreshold": 50,
      "Enabled": true
    }
  ],
  "CronHorizontalAutoscaler": [
    {
      "Name": "policy-test",
      "Period": "* * *",
      "Schedules": [
        {
          "StartAt": "00:00",
          "TargetReplicas": 1
        }
      ],
      "Enabled": true,
      "Priority": 0
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Result": "scaler-xxxxxx",
    "RequestId": "ab-xx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.AtLeastOneScalerRuleShouldBeApplied	请至少启用一种弹性规则。
InvalidParameterValue.AutoScalerLargerThanOne	只支持绑定一种弹性伸缩。
InvalidParameterValue.AutoScalerNameInvalid	弹性规则名称不合法, 长度必须小于20。
InvalidParameterValue.CronHpaReplicasInvalid	定时弹性伸缩目标实例数不合法。
InvalidParameterValue.HpaMetricsInvalid	弹性伸缩指标不合法。
InvalidParameterValue.HpaThresholdInvalid	弹性伸缩阈值不合法。
InvalidParameterValue.InvalidCronScalerPeriod	无效的定时伸缩周期。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.ScalerNameDuplicated	弹性伸缩名称已存在。
MissingParameter.AutoScalerNameNull	弹性规则名称不能为空。
MissingParameter.MinMaxNumNull	弹性伸缩配置最大值/最小值不能为空。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceUnavailable.ApplicationStopped	应用已停止。
UnauthorizedOperation.UnauthorizedOperation	未授权。

删除应用弹性策略组合

最近更新时间：2025-04-27 21:16:37

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

删除应用弹性策略组合

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：DeleteApplicationAutoscaler。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxxx
AutoscalerId	是	String	弹性伸缩策略ID 示例值：scaler-xxxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 删除应用弹性策略组合

删除应用弹性策略组合

输入示例

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

```
{
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0,
  "EnvironmentId": "en-xxxxxx",
  "AutoscalerId": "scaler-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "ab-xx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.DisableScalerBeforeDelete	弹性伸缩启用中，请停用后再删除。
MissingParameter.ScalerIdNull	弹性规则ID不能为空。
ResourceUnavailable.ApplicationStopped	应用已停止。
UnauthorizedOperation.UnauthorizedOperation	未授权。

获取应用弹性策略组合

最近更新时间：2025-04-27 21:16:37

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

获取应用弹性策略组合

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：DescribeApplicationAutoscalerList。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	Array of Autoscaler	弹性伸缩策略组合
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 获取应用弹性策略组合

获取应用弹性策略组合

输入示例

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

{
  "SourceChannel": 0,
```

```
"ApplicationId": "app-xxxxxx",
"EnvironmentId": "en-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": [
      {
        "AutoscalerId": "scaler-3j2o6ypg",
        "AutoscalerName": "elastic-test",
        "MinReplicas": 0,
        "MaxReplicas": 4,
        "CreateDate": "2024-12-05 11:10:19",
        "ModifyDate": "2024-12-20 17:00:23",
        "EnableDate": "2024-12-20 17:00:23",
        "Description": "这是一个描述",
        "Enabled": true,
        "HorizontalAutoscaler": [
          {
            "MinReplicas": 0,
            "MaxReplicas": 4,
            "Metrics": "CPU",
            "Threshold": 0,
            "DoubleThreshold": 60,
            "Enabled": true
          }
        ],
        "CronHorizontalAutoscaler": []
      },
      {
        "AutoscalerId": "scaler-b5q33njo",
        "AutoscalerName": "pk-test-clb",
        "MinReplicas": 0,
        "MaxReplicas": 4,
        "CreateDate": "2024-10-25 16:36:10",
        "ModifyDate": "2024-11-29 10:35:52",
        "EnableDate": "2024-11-29 10:35:52",
        "Description": "这是一个描述",
        "Enabled": false,
        "HorizontalAutoscaler": [
          {
            "MinReplicas": 0,
            "MaxReplicas": 4,
            "Metrics": "CPU",
            "Threshold": 0,
            "DoubleThreshold": 50,
            "Enabled": true
          }
        ],
        "CronHorizontalAutoscaler": [
          {
            "Name": "pk-test",
            "Period": "* * * *",
            "Schedules": [
```

```
{
  "StartAt": "02:00",
  "TargetReplicas": 1
},
{
  "Enabled": false,
  "Priority": 0
}
],
{
  "AutoscalerId": "scaler-ojr3nlj9",
  "AutoscalerName": "pk-test-1",
  "MinReplicas": 1,
  "MaxReplicas": 3,
  "CreateDate": "2024-12-19 23:38:45",
  "ModifyDate": "2024-12-19 23:38:45",
  "EnableDate": null,
  "Description": "",
  "Enabled": false,
  "HorizontalAutoscaler": [
    {
      "MinReplicas": 1,
      "MaxReplicas": 3,
      "Metrics": "CPU",
      "Threshold": 0,
      "DoubleThreshold": 50,
      "Enabled": true
    }
  ],
  "CronHorizontalAutoscaler": [
    {
      "Name": "policy-test",
      "Period": "* * *",
      "Schedules": [
        {
          "StartAt": "00:00",
          "TargetReplicas": 1
        }
      ],
      "Enabled": true,
      "Priority": 0
    }
  ],
  "RequestId": "c1d58f12-420f-464c-ac3e-f4f6a77284b0"
}
```

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
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

关闭应用弹性策略组合

最近更新时间：2025-04-25 02:00:13

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

关闭应用弹性策略组合

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：DisableApplicationAutoscaler。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
AutoscalerId	是	String	弹性伸缩策略ID 示例值：scaler-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 关闭应用弹性策略组合

关闭应用弹性策略组合

输入示例

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

```
{
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0,
  "EnvironmentId": "en-xxxxxx",
  "AutoscalerId": "scaler-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "ab-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
ResourceUnavailable.ApplicationStopped	应用已停止。
UnauthorizedOperation.UnauthorizedOperation	未授权。

启用应用弹性策略组合

最近更新时间：2025-04-25 02:00:13

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

启用应用弹性策略组合

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：EnableApplicationAutoscaler。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
AutoscalerId	是	String	弹性伸缩策略ID 示例值：scaler-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 启用应用弹性策略组合

启用应用弹性策略组合

输入示例

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

```
{
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0,
  "EnvironmentId": "en-xxx",
  "AutoscalerId": "scaler-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "abc-xxx-xxx-xxx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
ResourceUnavailable.ApplicationStopped	应用已停止。
UnauthorizedOperation.UnauthorizedOperation	未授权。

修改弹性伸缩策略组合

最近更新时间：2025-04-25 02:00:13

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

修改弹性伸缩策略组合

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：ModifyApplicationAutoscaler。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
AutoscalerId	是	String	弹性伸缩策略ID 示例值：scaler-xxxxxx
Autoscaler	是	Autoscaler	弹性伸缩策略
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 修改弹性伸缩策略组合

修改弹性伸缩策略组合

输入示例

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

<公共请求参数>

```
{
  "ApplicationId": "app-xxxxxx",
  "EnvironmentId": "en-xxxxxx",
  "AutoscalerId": "scaler-xxxxxx",
  "Autoscaler": {
    "MinReplicas": 0,
    "MaxReplicas": 4,
    "AutoscalerName": "pk-test-clb",
    "Description": "",
    "Enabled": false,
    "HorizontalAutoscaler": [
      {
        "MinReplicas": 0,
        "MaxReplicas": 4,
        "Metrics": "CPU",
        "DoubleThreshold": 50,
        "Enabled": true
      }
    ],
    "CronHorizontalAutoscaler": [
      {
        "Name": "pk-test",
        "Period": "* * *",
        "Schedules": [
          {
            "StartAt": "02:00",
            "TargetReplicas": 1
          }
        ],
        "Enabled": false,
        "Priority": 0
      }
    ]
  }
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "abbd6b63-xxx-xxx-xxx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.AtLeastOneScalerRuleShouldBeApplied	请至少启用一种弹性规则。
InvalidParameterValue.AutoScalerLargerThanOne	只支持绑定一种弹性伸缩。
InvalidParameterValue.CronHpaReplicasInvalid	定时弹性伸缩目标实例数不合法。
InvalidParameterValue.HpaMinMaxInvalid	弹性伸缩最小值/最大值不合法。
InvalidParameterValue.HpaThresholdInvalid	弹性伸缩阈值不合法。
MissingParameter.MinMaxNumNull	弹性伸缩配置最大值/最小值不能为空。
MissingParameter.ScalerIdNull	弹性规则ID不能为空。
ResourceUnavailable.ApplicationStopped	应用已停止。
UnauthorizedOperation.UnauthorizedOperation	未授权。

服务版本相关接口

服务删除

最近更新时间：2025-04-25 02:00:12

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

服务删除

- 停止当前运行服务
- 删除服务相关资源
- 删除服务

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：DeleteApplication。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务Id 示例值：app-xxxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxxx
SourceChannel	否	Integer	来源渠道(用户不需要关心此参数) 示例值：0
DeleteApplicationIfNoRunningVersion	否	Boolean	当服务没有任何运行版本时，是否删除此服务 示例值：false

3. 输出参数

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

4. 示例

示例1 服务版本删除

服务版本删除

输入示例

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

{
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0,
  "EnvironmentId": "en-xxxxxx",
  "DeleteApplicationIfNoRunningVersion": true
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": true
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation.DefaultInternalError	服务器繁忙,请稍后再试。
FailedOperation.DeleteServiceError	删除应用失败。
FailedOperation.DescribeIngressListError	查询 ingress 列表失败。

错误码	描述
InternalError.ActionReadTimeout	请求响应超时。
InternalError.DefaultInternalError	服务器繁忙,请稍后再试。
InternalError.DeleteServiceError	删除应用失败。
InternalError.DescribeIngressListError	查询 ingress 列表失败。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.ServiceFoundRunningVersion	应用存在正在运行的实例。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
InvalidParameterValue.VersionRouteRateNotZero	版本的路由流量不为0。
ResourceInUse.ResourceAlreadyLocked	你操作的资源已被其他操作占用，请稍后重试。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
ResourceNotFound.VersionServiceNotFound	找不到版本对应的应用。
ResourceUnavailable.ApplicationNotDeletable	有资源依赖，无法直接删除应用。
UnauthorizedOperation.UnauthorizedOperation	未授权。

单环境下所有应用状态查看

最近更新时间：2025-04-25 02:00:12

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

单环境下所有应用状态查看

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：DescribeApplicationsStatus。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvironmentId	是	String	环境ID 示例值：en-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

参数名称	类型	描述
Result	Array of ServiceVersionBrief	返回结果 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 单环境下所有应用状态查看

单环境下所有应用状态查看

输入示例

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

{
  "SourceChannel": 0,
  "EnvironmentId": "en-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "dda3b8c7-b536-4842-92a2-6756688a337e",
    "Result": [
      {
        "VersionName": "bj",
        "VersionId": "revision-jbzmeq7j",
        "Status": "normal",
        "BatchDeployStatus": "",
        "EnableEs": -1,
        "LogOutputConf": null,
        "ExpectedInstances": 1,
        "CurrentInstances": 1,
        "DeployMode": "JAR",
        "BuildTaskId": "build-5enzamz5",
        "EnvironmentId": "en-dpxydm15",
        "EnvironmentName": "bugfix-bj",
        "ApplicationId": "app-5vaz8x85",
        "ApplicationName": "app0925",
        "Zones": [
          "ap-beijing-3"
        ],
        "NodeInfos": [
          {
            "Name": "eklet-subnet-1ebj4rp9-766511",
            "Zone": "ap-beijing-3",
            "SubnetId": "subnet-1ebj4rp9",
            "AvailableIpCount": "232",
            "Cidr": "10.0.0.0-24"
          }
        ],
        "PodList": {
          "Offset": -1,
          "Limit": -1,
          "TotalCount": 1,
          "PodList": [
            {
              "Zone": "ap-beijing-3",
              "NodeInfo": {
                "Name": "eklet-subnet-1ebj4rp9-766511",
                "Zone": "ap-beijing-3",
                "SubnetId": "subnet-1ebj4rp9",
                "AvailableIpCount": "232",
                "Cidr": "10.0.0.0-24"
              },
              "DeployVersion": "",
              "RestartCount": 0,
              "Ready": true,
              "StartTime": "2024-09-25T03:39:59Z",
              "ContainerState": "running",
              "Unhealthy": null,
              "UnhealthyWarningMsg": "",
              "VersionId": "revision-jbzmeq7j",
              "ApplicationName": "app0925",
              "Status": "Running",
```

```
"CreateTime": "2024-09-25T03:39:08Z",
"Webshell": "https://tkecache.cloud.tencent.com/xxx",
"PodId": "app0925-p728w",
"PodIp": "10.0.0.7"
},
],
"RequestId": ""
},
"WorkloadInfo": {
  "ApplicationName": "app0925",
  "ClusterId": "cls-1trz6x51",
  "VersionName": "bj",
  "ReadyReplicas": 1,
  "Replicas": 1,
  "UpdatedReplicas": 1,
  "UpdatedReadyReplicas": 1,
  "UpdateRevision": "app0925-868b68987",
  "CurrentRevision": "app0925-868b68987"
},
"CreateDate": null,
"UnderDeploying": false,
"RegionId": "8"
},
{
  "VersionName": "hello-world",
  "VersionId": "revision-jordnl7j",
  "Status": "normal",
  "BatchDeployStatus": "",
  "EnableEs": -1,
  "LogOutputConf": null,
  "ExpectedInstances": 1,
  "CurrentInstances": 1,
  "DeployMode": "IMAGE",
  "BuildTaskId": "",
  "EnvironmentId": "en-dpxydm15",
  "EnvironmentName": "bugfix-bj",
  "ApplicationId": "app-57477ag5",
  "ApplicationName": "bugfix-test",
  "Zones": [
    "ap-beijing-3"
  ],
  "NodeInfos": [
    {
      "Name": "eklet-subnet-1ebj4rp9-766511",
      "Zone": "ap-beijing-3",
      "SubnetId": "subnet-1ebj4rp9",
      "AvailableIpCount": "232",
      "Cidr": "10.0.0.0-24"
    }
  ],
  "PodList": {
    "Offset": -1,
    "Limit": -1,
    "TotalCount": 1,
    "PodList": [
      {
        "Zone": "ap-beijing-3",
```

```
"NodeInfo": {
  "Name": "eklet-subnet-1ebj4rp9-766511",
  "Zone": "ap-beijing-3",
  "SubnetId": "subnet-1ebj4rp9",
  "AvailableIpCount": "232",
  "Cidr": "10.0.0.0-24"
},
"DeployVersion": "",
"RestartCount": 0,
"Ready": true,
"StartTime": "2024-08-05T08:15:26Z",
"ContainerState": "running",
"Unhealthy": null,
"UnhealthyWarningMsg": "",
"VersionId": "revision-jordn17j",
"ApplicationName": "bugfix-test",
"Status": "Running",
"CreateTime": "2024-08-05T08:14:28Z",
"Webshell": "https://tkecache.cloud.tencent.com/xxx",
"PodId": "bugfix-test-lvn56",
"PodIp": "10.0.0.16"
},
"RequestId": ""
},
"WorkloadInfo": {
  "ApplicationName": "bugfix-test",
  "ClusterId": "cls-1trz6x51",
  "VersionName": "hello-world",
  "ReadyReplicas": 1,
  "Replicas": 1,
  "UpdatedReplicas": 1,
  "UpdatedReadyReplicas": 1,
  "UpdateRevision": "bugfix-test-648f6b456d",
  "CurrentRevision": "bugfix-test-648f6b456d"
},
"CreateDate": null,
"UnderDeploying": false,
"RegionId": "8"
}
]
}
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

修改应用实例数量

最近更新时间：2025-04-25 02:00:12

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

修改应用实例数量

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：ModifyApplicationReplicas。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	应用id 示例值：app-xxxxxxx
EnvironmentId	是	String	环境ID 示例值：en-xxxxxxx
Replicas	是	Integer	实例数量 示例值：1
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 修改版本规格

修改版本规格

输入示例

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

```
{
  "ApplicationId": "app-xxxxxx",
  "SourceChannel": 0,
  "EnvironmentId": "en-xxxxxx",
  "Replicas": 0
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "abc-xxx-xxx-xxx"
  }
}
```

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
OperationDenied.ResourceIsolated	账号欠费状态下不支持该操作，请冲正后重试。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
UnauthorizedOperation.UnauthorizedOperation	未授权。

服务重启

最近更新时间：2025-04-25 02:00:12

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

服务重启

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：RestartApplication。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID/命名空间ID 示例值：en-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 服务重启

服务重启

输入示例

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

{
  "ApplicationId": "app-xxxxxx",
```

```
"SourceChannel": 0,
"EnvironmentId": "en-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "abc-xxx-xxx"
  }
}
```

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.DefaultInternalServerError	服务器繁忙,请稍后再试。
InternalServerError.RestartApplicationError	重启失败。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
MissingParameter.NamespaceIdNull	环境ID不能为空。
OperationDenied.ResourceIsolated	账号欠费状态下不支持该操作，请冲正后重试。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
UnauthorizedOperation.UnauthorizedOperation	未授权。

服务停止

最近更新时间：2025-04-25 02:00:11

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

服务停止

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：StopApplication。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	服务id 示例值：app-xxxxxx
EnvironmentId	是	String	环境ID/命名空间ID 示例值：en-xxxxxx
SourceChannel	否	Integer	来源渠道 示例值：0

3. 输出参数

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

4. 示例

示例1 服务版本停止

服务版本停止

输入示例

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

{
  "SourceChannel": 0,
```

```
"ApplicationId": "app-xxxxxx",
"EnvironmentId": "en-xxxxxx"
}
```

输出示例

```
{
  "Response": {
    "Result": true,
    "RequestId": "abc-xxxx-xxxx"
  }
}
```

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.StopApplicationError	停止应用失败。
MissingParameter.NamespaceIdNull	环境ID不能为空。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
UnauthorizedOperation.UnauthorizedOperation	未授权。

其他相关接口

生成Cos临时密钥

最近更新时间：2025-04-27 21:16:39

1. 接口描述

接口请求域名：tem.tencentcloudapi.com。

生成Cos临时密钥

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：CreateCosToken。
Version	是	String	公共参数 ，本接口取值：2021-07-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	String	应用ID 示例值：1
PkgName	是	String	包名 示例值：1
OptType	是	Integer	optType 1上传 2查询 示例值：1
SourceChannel	否	Integer	来源 channel 示例值：0
TimeVersion	否	String	充当deployVersion入参 示例值：1

3. 输出参数

参数名称	类型	描述
Result	CosToken	成功时为CosToken对象，失败为null
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 生成Cos临时密钥

为上传jar包和war包，生成Cos临时密钥

输入示例

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

```
X-TC-Action: CreateCosToken
```

<公共请求参数>

```
{
  "PkgName": "test.jar",
  "TimeVersion": "1.0",
  "ApplicationId": "app-testid",
  "SourceChannel": 0,
  "OptType": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "81f74023-563c-437d-abf7-8139449ef178",
    "Result": {}
  }
}
```

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.ActionReadTimeout	请求响应超时。
InternalServerError.DefaultInternalServerError	服务器繁忙,请稍后再试。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。
UnauthorizedOperation.UnauthorizedOperation	未授权。

数据结构

最近更新时间：2025-04-24 01:27:22

Autoscaler

弹性伸缩策略组合

被如下接口引用：CreateApplicationAutoscaler, DescribeApplicationAutoscalerList, DescribeApplicationInfo, ModifyApplicationAutoscaler。

名称	类型	必选	描述
MinReplicas	Integer	是	弹性伸缩最小实例数 示例值：0
MaxReplicas	Integer	是	弹性伸缩最大实例数 示例值：2
HorizontalAutoscaler	Array of HorizontalAutoscaler	否	指标弹性伸缩策略(指标策略和定时策略必须填写一个) 示例值：{"HorizontalAutoscaler":[{"DoubleThreshold":50,"Enabled":true,"MaxReplicas":4,"Metrics":"CPU",
CronHorizontalAutoscaler	Array of CronHorizontalAutoscaler	否	定时弹性伸缩策略(指标策略和定时策略必须填写一个) 示例值：{"CronHorizontalAutoscaler":[{"Enabled":false,"Name":"pk-test","I *","Priority":0,"Schedules":[{"StartAt":"02:00","TargetReplicas":1}]]}
AutoscalerId	String	否	弹性伸缩ID 示例值：scaler-xxxxxxx
AutoscalerName	String	否	弹性伸缩名称 示例值：scaler-name
Description	String	否	弹性伸缩描述 示例值：这是一个描述
CreateDate	String	否	创建日期 示例值：2024-10-25 16:36:10
ModifyDate	String	否	修改时间 示例值：2024-10-25 16:36:10
EnableDate	String	否	启用时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2024-10-25 16:36:10
Enabled	Boolean	否	是否启用 示例值：true

ConfigData

配置

被如下接口引用：DescribeConfigData, DescribeConfigDataList。

名称	类型	描述
Name	String	配置名称 示例值：name-xxx
CreateTime	String	创建时间 示例值：2012-12-10 09:00:00
RelatedApplications	Array of TemService	关联的服务列表
Data	Array of Pair	配置条目

CosToken

Cos token

被如下接口引用：CreateCosToken。

名称	类型	描述
RequestId	String	唯一请求 ID 示例值：1
Bucket	String	存储桶桶名 示例值：1
Region	String	存储桶所在区域 示例值：1
TmpSecretId	String	临时密钥的SecretId 示例值：1
TmpSecretKey	String	临时密钥的SecretKey 示例值：1
SessionToken	String	临时密钥的 sessionToken 示例值：1
StartTime	String	临时密钥获取的开始时间 示例值：1
ExpiredTime	String	临时密钥的 expiredTime 示例值：1
FullPath	String	包完整路径 示例值：1

CronHorizontalAutoscaler

定时伸缩策略

被如下接口引用：CreateApplicationAutoscaler, DeployApplication, DescribeApplicationAutoscalerList, DescribeApplicationInfo, ModifyApplicationAutoscaler。

名称	类型	必选	描述
Name	String	否	定时伸缩策略名称 示例值：1
Period	String	否	策略周期 **，三个范围，第一个是天，第二个月，第三个是周，中间用空格隔开 例子： **（每天） **0-3（每周日到周三） 1,11,21*（每个月1号，11号，21号） 示例值：**
Schedules	Array of CronHorizontalAutoscalerSchedule	否	定时伸缩策略明细 示例值：1
Enabled	Boolean	否	是否启用 示例值：1
Priority	Integer	否	策略优先级，值越大优先级越高，0为最小值 示例值：1

CronHorizontalAutoscalerSchedule

定时伸缩策略明细

被如下接口引用：CreateApplicationAutoscaler, DeployApplication, DescribeApplicationAutoscalerList, DescribeApplicationInfo, ModifyApplicationAutoscaler。

名称	类型	必选	描述
StartAt	String	是	触发事件，小时分钟，用:分割 例如 00:00（零点零分触发） 示例值：00:00
TargetReplicas	Integer	否	目标实例数（不大于50） 示例值：1

DeployServiceBatchDetail

分批发布单批次详情

被如下接口引用：DescribeDeployApplicationDetail。

名称	类型	必选	描述
OldPodList	DeployServicePodDetail	否	旧实例列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[1]
NewPodList	DeployServicePodDetail	否	新实例列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[1]
BatchStatus	String	否	当前批次状态："WaitForTimeExceed", "WaitForResume", "Deploying", "Finish", "NotStart" 示例值：Finish
PodNum	Integer	否	该批次预计旧实例数量 示例值：3
BatchIndex	Integer	否	批次id 示例值：0
OldPods	Array of DeployServicePodDetail	否	旧实例列表 示例值：[1]
NewPods	Array of DeployServicePodDetail	否	新实例列表 示例值：[1]
NextBatchStartTime	Integer	否	=0：手动确认批次；>0：下一批次开始时间戳 示例值：0

DeployServicePodDetail

分批发布单批次详情

被如下接口引用：DescribeDeployApplicationDetail。

名称	类型	必选	描述
PodId	String	否	pod id 示例值：pod-123
PodStatus	Array of String	否	pod状态 示例值：Pending
PodVersion	String	否	pod版本 示例值：v1.1
CreateTime	String	否	pod创建时间 示例值：2020-01-01 11: 11: 11
Zone	String	否	pod所在可用区 示例值：shanghai-zone-4
Webshell	String	否	webshell地址

名称	类型	必选	描述
			示例值：webshell
Status	String	否	状态 示例值：1

DeployStrategyConf

分批发布策略配置

被如下接口引用：DeployApplication, DescribeApplicationInfo, DescribeDeployApplicationDetail。

名称	类型	必选	描述
TotalBatchCount	Integer	否	总分批数 示例值：10
BetaBatchNum	Integer	否	beta分批实例数 示例值：5
DeployStrategyType	Integer	否	分批策略：0-全自动，1-全手动，2-beta分批，beta批一定是手动的，3-首次发布 示例值：1
BatchInterval	Integer	否	每批暂停间隔 示例值：200
MinAvailable	Integer	否	最小可用实例数 示例值：1
Force	Boolean	否	是否强制发布 示例值：false

DescribeConfigDataListPage

配置信息的分页列表

被如下接口引用：DescribeConfigDataList。

名称	类型	描述
Records	Array of ConfigData	记录
ContinueToken	String	分页游标，用以查询下一页 示例值：token-xxx
RemainingCount	Integer	剩余数目 注意：此字段可能返回 null，表示取不到有效值。 示例值：22

DescribeRunPodPage

版本pod列表

被如下接口引用：DescribeApplicationInfo, DescribeApplicationPods, DescribeApplicationsStatus, DescribeDeployApplicationDetail。

名称	类型	必选	描述
Offset	Integer	否	分页下标 示例值：0
Limit	Integer	否	单页条数 示例值：10
TotalCount	Integer	否	总数 示例值：1
RequestId	String	否	请求id 示例值：1ce4bb30-202a-403a-8552-1a38b21xxxx

名称	类型	必选	描述
PodList	Array of RunVersionPod	否	条目 示例值： [{"ApplicationName":"app0925","Webshell":"https://tkecache.cloud.tencent.com/xxxxxx","PodId":"xxxxxx"}]

EksService

eks service info

被如下接口引用：DeployApplication, DescribeApplicationInfo, DescribeApplicationServiceList, ModifyApplicationService。

名称	类型	必选	描述
Name	String	否	service name 示例值：name-xxx
Ports	Array of Integer	否	可用端口 示例值：[8080]
Yaml	String	否	yaml 内容 示例值：apiVersion: v1-xxxxxx
ApplicationName	String	否	服务名 示例值：app-name-xxx
VersionName	String	否	版本名 示例值：version-xxxx
ClusterIp	Array of String	否	内网ip 示例值：172.xx.xx.xx
ExternalIp	String	否	外网ip 示例值：10.xx.xx.xx
Type	String	否	访问类型，可选值： - EXTERNAL（公网访问） - VPC（vpc内访问） - CLUSTER（集群内访问） 示例值：VPC
SubnetId	String	否	子网ID，只在类型为vpc访问时才有值 示例值：subnet-xxxxxx
LoadBalancerId	String	否	负载均衡ID，只在外网访问和vpc内访问才有值，默认自动创建 示例值：10.xx.xx.xx
PortMappings	Array of PortMapping	否	端口映射
ServicePortMappingList	Array of ServicePortMapping	否	每种类型访问配置详情
FlushAll	Boolean	否	刷新复写所有类型 示例值：true
EnableRegistryNextDeploy	Integer	否	1: 下次部署自动注入注册中心信息；0: 不注入 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
ApplicationId	String	否	返回应用id 示例值：app-xxxxxx
AllIpDone	Boolean	否	所有服务IP是否已经ready 示例值：false
ExternalDomain	String	否	clb 域名 示例值：lb-xxxx.com

EnablePrometheusConf

开启prometheus监控配置

被如下接口引用：DeployApplication, DescribeApplicationInfo。

名称	类型	必选	描述
Port	Integer	否	应用开放的监听端口 示例值：9090
Path	String	否	业务指标暴露的url path 示例值：/metrics

EsInfo

弹性伸缩配置

被如下接口引用：DeployApplication, DescribeApplicationInfo。

名称	类型	必选	描述
MinAliveInstances	Integer	是	最小实例数 示例值：1
MaxAliveInstances	Integer	是	最大实例数 示例值：5
EsStrategy	Integer	是	弹性策略,1:cpu, 2:内存 示例值：1
Threshold	Integer	是	弹性扩缩容条件值 示例值：50
VersionId	String	否	版本Id 示例值：version-test

HealthCheckConfig

健康检查配置

被如下接口引用：DeployApplication, DescribeApplicationInfo。

名称	类型	必选	描述
Type	String	是	支持的健康检查类型，如 HttpGet, TcpSocket, Exec 示例值：HttpGet
Protocol	String	否	仅当健康检查类型为 HttpGet 时有效，表示协议类型，如 HTTP，HTTPS 示例值：HTTP
Path	String	否	仅当健康检查类型为 HttpGet 时有效，表示请求路径 示例值：/xxx
Exec	String	否	仅当健康检查类型为 Exec 时有效，表示执行的脚本内容 示例值：shell cmd
Port	Integer	否	仅当健康检查类型为 HttpGet\TcpSocket 时有效，表示请求路径 示例值：80
InitialDelaySeconds	Integer	否	检查延迟开始时间，单位为秒，默认为 0 示例值：0
TimeoutSeconds	Integer	否	超时时间，单位为秒，默认为 1 示例值：1
PeriodSeconds	Integer	否	间隔时间，单位为秒，默认为 10 示例值：10

HorizontalAutoscaler

弹性伸缩策略

被如下接口引用：CreateApplicationAutoscaler, DeployApplication, DescribeApplicationAutoscalerList, DescribeApplicationInfo, ModifyApplicationAutoscaler。

名称	类型	必选	描述
MinReplicas	Integer	否	最小实例数（可以不传） 示例值：1
MaxReplicas	Integer	否	最大实例数（可以不传） 示例值：1
Metrics	String	否	指标度量 CPU（CPU使用率，%） MEMORY（内存使用率，%） CPU_CORE_USED（CPU使用量，core） MEMORY_SIZE_USED(内存使用量，MiB) NETWORK_BANDWIDTH_RECEIVE(网络入带宽，Mbps) NETWORK_BANDWIDTH_TRANSMIT(网络出带宽，Mbps) NETWORK_TRAFFIC_RECEIVE(网络入流量，MiB/s) NETWORK_TRAFFIC_TRANSMIT(网络出流量，MiB/s) NETWORK_PACKETS_RECEIVE(网络入包量，Count/s) NETWORK_PACKETS_TRANSMIT(网络出包量，Count/s) FS_IOPS_WRITE(磁盘写次数，Count/s) FS_IOPS_READ(磁盘读次数，Count/s) FS_SIZE_WRITE(磁盘写大小，MiB/s) FS_SIZE_READ(磁盘读大小，MiB/s) 示例值：1
Threshold	Integer	否	阈值（整数） 示例值：1
Enabled	Boolean	否	是否启用 示例值：1
DoubleThreshold	Float	否	阈值（小数，优先使用） 示例值：1

IngressInfo

Ingress 配置

被如下接口引用：DescribeIngress, DescribeIngresses, DescribeRelatedIngresses, ModifyIngress。

名称	类型	必选	描述
EnvironmentId	String	是	环境ID 示例值：en-xxxxxxx
AddressIPVersion	String	是	ip version 示例值：IPV4
IngressName	String	是	ingress name 示例值：ingress name
Rules	Array of IngressRule	是	rules 配置 示例值：rules 配置
ClbId	String	是	clb ID 示例值：clb-xxx
ClusterNamespace	String	否	环境namespace 示例值：default
Tls	Array of IngressTls	否	tls 配置 注意：此字段可能返回 null，表示取不到有效值。 示例值：tls 配置

名称	类型	必选	描述
ClusterId	String	否	环境集群ID 示例值：cls-xxx
Vip	String	否	clb ip 示例值：10.10.10.10
CreateTime	String	否	创建时间 示例值：2010-11-11 22:21:11
Mixed	Boolean	否	是否混合 https，默认 false，可选值 true 代表有 https 协议监听 示例值：false
RewriteType	String	否	重定向模式，可选值： - AUTO（自动重定向http到https） - NONE（不使用重定向） 示例值：1
Domain	String	否	clb 域名 示例值：lb-xxx

IngressRule

ingress rule 配置

被如下接口引用：DescribeIngress, DescribeIngresses, DescribeRelatedIngresses, ModifyIngress。

名称	类型	必选	描述
Http	IngressRuleValue	是	ingress rule value 示例值：ingress rule value
Host	String	否	host 地址 示例值：xxx.com
Protocol	String	否	协议，选项为 http，https，默认为 http 示例值：protocol

IngressRuleBackend

Ingress 规则 backend 配置

被如下接口引用：DescribeIngress, DescribeIngresses, DescribeRelatedIngresses, ModifyIngress。

名称	类型	必选	描述
ServiceName	String	是	eks service 名 示例值：serviceA
ServicePort	Integer	是	eks service 端口 示例值：8080

IngressRulePath

Ingress Rule Path 配置

被如下接口引用：DescribeIngress, DescribeIngresses, DescribeRelatedIngresses, ModifyIngress。

名称	类型	必选	描述
Path	String	是	path 信息 示例值：/xxx
Backend	IngressRuleBackend	是	backend 配置 示例值：{"ServiceName": "serviceA", "ServicePort": 8080}

IngressRuleValue

Ingress Rule Value 配置

被如下接口引用：DescribeIngress, DescribeIngresses, DescribeRelatedIngresses, ModifyIngress。

名称	类型	必选	描述
Paths	Array of IngressRulePath	是	rule 整体配置 示例值：rulePaths 数组

IngressTls

ingress tls 配置

被如下接口引用：DescribeIngress, DescribeIngresses, DescribeRelatedIngresses, ModifyIngress。

名称	类型	必选	描述
Hosts	Array of String	是	host 数组, 空数组表示全部域名的默认证书 示例值：["xx.com", "yyy.com"]
SecretName	String	是	secret name, 如使用证书, 则填空字符串 示例值：secret
CertificateId	String	否	SSL Certificate Id 示例值：CertificateId

LogConfig

日志收集配置

被如下接口引用：DescribeLogConfig, DescribePagedLogConfigList, ModifyLogConfig。

名称	类型	必选	描述
Name	String	否	名称 示例值：ab-xxx
InputType	String	否	收集类型, container_stdout 为标准输出; container_file 为文件; 示例值：container_stdout
LogsetId	String	否	日志集 ID 示例值：4d43fac5-fc0a-42bf-92a9-ef5b5b245e6a
TopicId	String	否	日志主题 ID 示例值：dbebda3e-0a9a-4888-9d5c-642bf2ffa183
LogType	String	否	日志提取模式, minimalist_log 为单行全文; multiline_log 为多行全文; fullregex_log 为单行正则; multiline_fullregex_log 为多行正则; json_log 为 json; 示例值：json_log
BeginningRegex	String	否	首行正则表达式, 当 LogType 为多行全文、多行正则时生效 示例值：^ERROR
LogPath	String	否	收集文件目录, 当 InputType=container_file 时生效 示例值：/abc
FilePattern	String	否	收集文件名模式, 当 InputType=container_file 时生效 示例值：app_*.log
CreateDate	String	否	创建时间 示例值：2022-05-20T14:42:16Z
ModifyDate	String	否	更新时间 示例值：2022-05-20T14:42:16Z
ApplicationId	String	否	应用 ID 示例值：app-xxx

名称	类型	必选	描述
ApplicationName	String	否	应用名 示例值：abc-xxx
ExtractRule	LogConfigExtractRule	否	导出规则

LogConfigExtractRule

日志采集的导出规则配置

被如下接口引用：CreateLogConfig, DescribeLogConfig, DescribePagedLogConfigList, ModifyLogConfig。

名称	类型	必选	描述
BeginningRegex	String	否	首行正则表达式 示例值：^ERROR
Keys	Array of String	否	提取结果 示例值：["k1", "k2"]
FilterKeys	Array of String	否	过滤键 示例值：["k1"]
FilterRegex	Array of String	否	过滤值 示例值：["v1*"]
LogRegex	String	否	日志正则表达式 示例值：^Info
TimeKey	String	否	时间字段 示例值：time
TimeFormat	String	否	时间格式 示例值：%Y-%m-%d %H:%M:%S
UnMatchUpload	String	否	是否上传解析失败日志 示例值：true
UnMatchedKey	String	否	解析失败日志的键名称 示例值：LogParseFailure
Backtracking	String	否	tracking 示例值：tracking-xxx
Delimiter	String	否	分隔符 示例值：:

LogConfigListPage

LogConfig 列表结果

被如下接口引用：DescribePagedLogConfigList。

名称	类型	描述
Records	Array of LogConfig	记录
ContinueToken	String	翻页游标 示例值：token-xxx

LogOutputConf

日志输出配置

被如下接口引用：DeployApplication, DescribeApplicationInfo, DescribeApplicationsStatus。

名称	类型	必选	描述
OutputType	String	是	日志消费端类型 示例值：cls、kafka
ClsLogsetName	String	否	cls日志集 示例值：1
ClsLogTopicId	String	否	cls日志主题 示例值：1
ClsLogsetId	String	否	cls日志集id 示例值：1
ClsLogTopicName	String	否	cls日志名称 示例值：1

MountedSettingConf

挂载配置信息

被如下接口引用：DeployApplication, DescribeApplicationInfo。

名称	类型	必选	描述
ConfigDataName	String	是	配置名称 示例值：1
MountedPath	String	是	挂载路径 示例值：1
Data	Array of Pair	否	配置内容
SecretDataName	String	否	加密配置名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

NamespaceInfo

Namespace 基础信息

被如下接口引用：DescribeEnvironment。

名称	类型	描述
EnvironmentId	String	ID 信息 示例值：env-xxx
NamespaceName	String	名字（已弃用） 示例值：name-test
Region	String	地域 示例值：ap-shanghai
VpcId	String	vpc id 示例值：vpc-k1ae4nhe
SubnetIds	Array of String	subnet id 数组 示例值：["subnet-xxx"]
Description	String	描述 示例值：这是一个描述
CreateDate	String	创建时间 示例值：2012-08-08
EnvironmentName	String	环境名称 示例值：1

名称	类型	描述
ApmInstanceId	String	APM 资源 ID 示例值：instance-xxx
Locked	Integer	环境是否上锁，1为上锁，0则未上锁 示例值：1
Tags	Array of Tag	标签 示例值：1
EnvType	String	环境类型：test、pre、prod 示例值：prod

NamespacePage

命名空间分页

被如下接口引用：DescribeEnvironments。

名称	类型	描述
Records	Array of TemNamespaceInfo	分页内容 示例值：1
Total	Integer	总数 示例值：1
Size	Integer	条目数 示例值：1
Pages	Integer	页数 示例值：1
Current	Integer	当前条目 示例值：1

NamespaceStatusInfo

命名空间状态

被如下接口引用：DescribeEnvironmentStatus。

名称	类型	必选	描述
EnvironmentId	String	是	命名空间id 示例值：1
EnvironmentName	String	是	命名空间名称 示例值：1
ClusterId	String	否	TCB envId EKS clusterId 示例值：env-xxx cls-xxx
ClusterStatus	String	否	环境状态 示例值：normal unavailable
EnvironmentStartingStatus	TemEnvironmentStartingStatus	否	环境启动状态（不在启动中为null） 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
EnvironmentStoppingStatus	TemEnvironmentStoppingStatus	否	环境停止状态（不在停止中为null） 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

NodeInfo

node信息

被如下接口引用：DescribeApplicationInfo, DescribeApplicationPods, DescribeApplicationsStatus。

名称	类型	必选	描述
Name	String	否	node名字 示例值：eklet-subnet-kctzw7nj-xxxxxx
Zone	String	否	node可用区 示例值：ap-shanghai-1
SubnetId	String	否	node子网ID 示例值：subnet-xxxxxxx
AvailableIpCount	String	否	可用IP数 示例值：128
Cidr	String	否	cidr块 示例值：10.0.10.0-24

Pair

键值对

被如下接口引用：CreateConfigData, DeployApplication, DescribeApplicationInfo, DescribeConfigData, ModifyConfigData。

名称	类型	必选	描述
Key	String	是	键 示例值：1
Value	String	是	值 示例值：1
Type	String	否	类型，default 为自定义，reserved 为系统变量，referenced 为引用配置项 示例值：default
Config	String	否	配置名称 示例值：configName
Secret	String	否	加密配置名称 示例值：1

PortMapping

服务端口映射

被如下接口引用：DeployApplication, DescribeApplicationInfo, DescribeApplicationServiceList, ModifyApplicationService。

名称	类型	必选	描述
Port	Integer	是	端口 示例值：1
TargetPort	Integer	是	映射端口 示例值：1
Protocol	String	是	协议栈 TCP/UDP 示例值：1
ServiceName	String	否	k8s service名称 示例值：service123

QueryFilter

查询过滤器

被如下接口引用：DescribeApplications, DescribeEnvironments。

名称	类型	必选	描述
Name	String	否	查询字段名称 示例值：1
Value	Array of String	否	查询字段值 示例值：1

RunVersionPod

应用实例

被如下接口引用：DescribeApplicationInfo, DescribeApplicationPods。

名称	类型	必选	描述
Webshell	String	否	shell地址 示例值：https://tkecache.cloud.tencent.com/xxxx
PodId	String	否	pod的id 示例值：pod-jt5bn
Status	String	否	状态 示例值：Running
CreateTime	String	否	创建时间 示例值：2010-11-11 22:21:11
PodIp	String	否	实例的ip 示例值：127.0.0.1
Zone	String	否	可用区 示例值：ap-shanghai-1
DeployVersion	String	否	部署版本 示例值：20241202165113
RestartCount	Integer	否	重启次数 示例值：2
Ready	Boolean	否	pod是否就绪 示例值：true
ContainerState	String	否	容器状态 示例值：running
NodeInfo	NodeInfo	否	实例所在节点信息
StartTime	String	否	启动时间 示例值：2024-12-20T09:01:01Z
Unhealthy	Boolean	否	是否健康 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
UnhealthyWarningMsg	String	否	不健康时的提示信息 示例值：message
VersionId	String	否	版本ID 示例值：revision-xxxxxx
ApplicationName	String	否	应用名称 示例值：app-name-xxx

ServicePage

服务分页

被如下接口引用：DescribeApplications。

名称	类型	必选	描述
Records	Array of TemService	是	条目 示例值：1
Total	Integer	是	总数 示例值：1
Size	Integer	是	条目 示例值：1
Pages	Integer	是	页数 示例值：1
Current	Integer	否	当前条数 示例值：1

ServicePortMapping

端口映射详细信息结构体

被如下接口引用：CreateApplicationService, DeployApplication, DescribeApplicationInfo, DescribeApplicationServiceList, ModifyApplicationService。

名称	类型	必选	描述
Type	String	否	服务类型：如：EXTERNAL，VPC，CLUSTER 示例值：ClusterIP
ServiceName	String	否	服务名称 示例值：consumer
ClusterIp	String	否	集群内访问vip 示例值：10.0.0.2
ExternalIp	String	否	集群外方位vip 示例值：202.0.0.2
SubnetId	String	否	子网id 示例值：subnet-123
VpcId	String	否	vpc id 示例值：vpc-123
LoadBalanceId	String	否	LoadBalance Id 示例值：clb-123
Yaml	String	否	yaml 内容 示例值：yaml
Ports	Array of Integer	否	暴露端口列表 示例值：[100,2000,3000]
PortMappingItemList	Array of ServicePortMappingItem	否	端口映射数组
ExternalDomain	String	否	clb domain 示例值：lb-xxxx.com

ServicePortMappingItem

服务端口映射条目

被如下接口引用：CreateApplicationService, DeployApplication, DescribeApplicationInfo, DescribeApplicationServiceList, ModifyApplicationService。

名称	类型	必选	描述
Port	Integer	否	应用访问端口 示例值：1000

名称	类型	必选	描述
TargetPort	Integer	否	应用监听端口 示例值：2000
Protocol	String	否	协议类型 示例值：TCP

ServiceVersionBrief

服务版本信息列表

被如下接口引用：DescribeApplications, DescribeApplicationsStatus, DescribeConfigData。

名称	类型	必选	描述
VersionName	String	否	版本名称 示例值：1
Status	String	否	状态 示例值：1
EnableEs	Integer	否	是否启动弹性 -- 已废弃 示例值：1
CurrentInstances	Integer	否	当前实例 示例值：10
VersionId	String	否	version的id 示例值：10
LogOutputConf	LogOutputConf	否	日志输出配置 -- 已废弃 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
ExpectedInstances	Integer	否	期望实例 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
DeployMode	String	否	部署方式 示例值：1
BuildTaskId	String	否	建构任务ID 示例值：1
EnvironmentId	String	否	环境ID 示例值：1
EnvironmentName	String	否	环境name 示例值：1
ApplicationId	String	否	服务ID 示例值：1
ApplicationName	String	否	服务name 示例值：1
UnderDeploying	Boolean	否	是否正在发布中 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
BatchDeployStatus	String	否	分批次部署状态 示例值：1
Zones	Array of String	否	可用区 示例值：1
NodeInfos	Array of NodeInfo	否	节点信息 示例值：1

名称	类型	必选	描述
PodList	DescribeRunPodPage	否	实例信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
WorkloadInfo	WorkloadInfo	否	工作负载信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
CreateDate	String	否	创建日期 示例值：1
RegionId	String	否	地域id 示例值：4

SortType

查询过滤器

被如下接口引用：DescribeApplications, DescribeEnvironments。

名称	类型	必选	描述
Key	String	否	排序字段名称 示例值：1
Type	Integer	否	0：升序，1：倒序 示例值：0

StorageConf

存储卷配置

被如下接口引用：DeployApplication, DescribeApplicationInfo。

名称	类型	必选	描述
StorageVolName	String	是	存储卷名称 示例值：1
StorageVolPath	String	是	存储卷路径 示例值：1
StorageVollp	String	否	存储卷IP 示例值：1

StorageMountConf

数据卷挂载信息

被如下接口引用：DeployApplication, DescribeApplicationInfo。

名称	类型	必选	描述
VolumeName	String	是	数据卷名 示例值：1
MountPath	String	是	数据卷绑定路径 示例值：1

Tag

标签

被如下接口引用：CreateApplication, CreateEnvironment, DescribeApplicationInfo, DescribeApplications, DescribeConfigData, DescribeEnvironment, DescribeEnvironments。

名称	类型	必选	描述
TagKey	String	否	标签键 示例值：1
TagValue	String	否	标签值 示例值：1

TemDeployApplicationDetailInfo

分批发布详情

被如下接口引用：DescribeDeployApplicationDetail。

名称	类型	描述
DeployStrategyConf	DeployStrategyConf	分批发布策略 注意：此字段可能返回 null，表示取不到有效值。
StartTime	String	开始时间 示例值：2025-02-18 18:23:09
EndTime	String	结束时间 示例值：2025-02-18 18:23:09
Status	String	当前状态 示例值：Running
BetaBatchDetail	DeployServiceBatchDetail	beta分批详情 注意：此字段可能返回 null，表示取不到有效值。
OtherBatchDetail	Array of DeployServiceBatchDetail	其他分批详情
OldVersionPodList	DescribeRunPodPage	老版本pod列表 注意：此字段可能返回 null，表示取不到有效值。
CurrentBatchIndex	Integer	当前批次id 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
ErrorMessage	String	错误原因 示例值：errormesage
CurrentBatchStatus	String	当前批次状态 示例值：revert
NewDeployVersion	String	新版本version 示例值：2018222
OldDeployVersion	String	旧版本version 示例值：2018221
NewVersionPackageInfo	String	包名称 示例值：hello-world.jar
NextBatchStartTime	Integer	下一批次开始时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：12342423452

TemEnvironmentStartingStatus

环境启动进程（只统计由环境启动操作触发的应用数量）

被如下接口引用：DescribeEnvironmentStatus。

名称	类型	必选	描述
ApplicationNumNeedToStart	Integer	否	需要启动的应用数量 示例值：1

名称	类型	必选	描述
StartedApplicationNum	Integer	否	已经启动的应用数量 示例值：1
StartFailedApplicationNum	Integer	否	启动失败的应用数量 示例值：1

TemEnvironmentStoppingStatus

环境停止进程（只统计由环境停止操作触发的应用数量）
被如下接口引用：DescribeEnvironmentStatus。

名称	类型	必选	描述
ApplicationNumNeedToStop	Integer	否	需要停止的应用数量 示例值：1
StoppedApplicationNum	Integer	否	已经停止的应用数量 示例值：1
StopFailedApplicationNum	Integer	否	停止失败的应用数量 示例值：1

TemNamespaceInfo

命名空间对象
被如下接口引用：DescribeEnvironments。

名称	类型	描述
EnvironmentId	String	环境id 示例值：en-xxxxxx
Channel	String	渠道 示例值：0
EnvironmentName	String	环境名称 示例值：en-name-xxx
Region	String	区域名称 示例值：ap-shanghai
Description	String	环境描述 示例值：这是一个描述
Status	Integer	状态,1:已销毁;0:正常 示例值：1
Vpc	String	vpc网络 示例值：vpc-xxxxxx
CreateDate	String	创建时间 示例值：2024-10-29 18:53:24
ModifyDate	String	修改时间 示例值：2024-10-29 18:53:24
Modifier	String	修改人 示例值：10001xxxxx
Creator	String	创建人 示例值：10001xxxxx
ApplicationNum	Integer	应用数 示例值：2

名称	类型	描述
RunInstancesNum	Integer	运行实例数 示例值：2
SubnetId	String	子网络 示例值：subnet-xxxxxx
ClusterStatus	String	环境集群 status 示例值：NORMAL
EnableTswTraceService	Boolean	是否开启tsw 示例值：true
Locked	Integer	环境锁，1为上锁，0则为上锁 示例值：1
Appld	String	用户Appld 示例值：1305xxxxxx
Uin	String	用户Uin 示例值：100019xxxxxx
SubAccountUin	String	用户SubAccountUin 示例值：100039xxxxxx
ClusterId	String	集群ID 示例值：cls-xxxxxx
Tags	Array of Tag	标签
HasAuthority	Boolean	资源是否有权限 示例值：false
EnvType	String	环境类型：test、pre、prod 示例值：prod
RegionId	String	地域码 示例值：4

TemService

服务

被如下接口引用：DescribeApplications, DescribeConfigData。

名称	类型	必选	描述
ApplicationId	String	否	主键 示例值：1
ApplicationName	String	否	服务名 示例值：10
Description	String	否	描述 示例值：10
EnvironmentId	String	否	命名空间id 示例值：1
CreateDate	Timestamp	否	创建时间 示例值：10
ModifyDate	Timestamp	否	修改时间 示例值：10
Modifier	String	否	修改人 示例值：1
Creator	String	否	创建者 示例值：1

名称	类型	必选	描述
RepoType	Integer	否	tcr个人版or企业版 示例值：0
InstanceId	String	否	企业版实例id 示例值：id123
RepoName	String	否	镜像仓库名 示例值：tem-ns/repo
CodingLanguage	String	否	编程语言 示例值：1
DeployMode	String	否	部署方式 示例值：1
EnvironmentName	String	否	环境名称 示例值：1
ActiveVersions	Array of ServiceVersionBrief	否	服务当前运行环境的实例信息 示例值：1
EnableTracing	Integer	否	是否启用链路追踪 示例值：1
Tags	Array of Tag	否	标签 示例值：1
HasAuthority	Boolean	否	是否有资源权限 示例值：1

TemServiceVersionInfo

版本信息

被如下接口引用：DescribeApplicationInfo。

名称	类型	必选	描述
VersionId	String	是	主键 示例值：revision-xxxxxxx
ApplicationId	String	是	服务id 示例值：app-xxxxxxx
DeployMode	String	是	部署方式 示例值：JAR
JdkVersion	String	是	jdk版本 示例值：KONA:11
Description	String	是	描述 示例值：这是一个描述
DeployVersion	String	是	部署版本 示例值：20241016164535
PublishMode	String	是	发布方式 示例值：RollingUpdate
JvmOpts	String	是	启动参数 示例值：-Xms128m -XX:MetaspaceSize=128m
InitPodNum	Integer	是	初始实例 示例值：2
CpuSpec	Float	是	cpu规格 示例值：2

名称	类型	必选	描述
MemorySpec	Float	是	内存规格 示例值：4
ImgRepo	String	是	镜像路径 示例值：img-repo-xxx
ImgName	String	是	镜像名称 示例值：tem-xxxx/image-name
ImgVersion	String	是	镜像版本 示例值：20241016164535
EsInfo	EsInfo	是	弹性配置 注意：此字段可能返回 null，表示取不到有效值。
EnvConf	Array of Pair	是	环境配置
StorageConfs	Array of StorageConf	是	存储配置
Status	String	是	运行状态 示例值：normal
Vpc	String	是	私有网络 示例值：vpc-xxxxxxx
SubnetId	String	是	子网网络 示例值：subnet-xxxxxxx
CreateDate	String	是	创建时间 示例值：2024-10-16 16:45:17
ModifyDate	String	是	修改时间 示例值：2024-10-16 16:45:17
StorageMountConfs	Array of StorageMountConf	是	挂载配置
VersionName	String	是	版本名称 示例值：version-name-xxx
LogOutputConf	LogOutputConf	否	日志输出配置 注意：此字段可能返回 null，表示取不到有效值。
ApplicationName	String	否	服务名称 示例值：app-name-xxx
ApplicationDescription	String	否	服务描述 示例值：这是一个描述
EnvironmentName	String	否	环境名称 示例值：env-name-xxx
EnvironmentId	String	否	环境ID 示例值：en-xxxxxxx
PublicDomain	String	否	公网地址 示例值：pub-domain-xxx.com
EnablePublicAccess	Boolean	否	是否开通公网访问 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
CurrentInstances	Integer	否	现有的实例 示例值：2
ExpectedInstances	Integer	否	期望的实例 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
CodingLanguage	String	否	编程语言 示例值：JAVA

名称	类型	必选	描述
PkgName	String	否	程序包名 示例值：tem/pkg/xxx/app-xxx/xxxxx/helloworld.jar
EsEnable	Integer	否	是否启用弹性伸缩 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
EsStrategy	Integer	否	弹性策略 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
ImageTag	String	否	镜像tag 示例值：20241016164535
LogEnable	Integer	否	是否启用log 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
MinAliveInstances	String	否	最小实例数 示例值：2
SecurityGroupIds	Array of String	否	安全组 示例值：sg-xxxxxx
ImageCommand	String	否	镜像命令 示例值：ls
ImageArgs	Array of String	否	镜像命令参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：["-c", "100"]
UseRegistryDefaultConfig	Boolean	否	是否使用默认注册中心配置 示例值：false
Service	EksService	否	eks 访问设置
SettingConfs	Array of MountedSettingConf	否	挂载配置信息
LogConfs	Array of String	否	log path数组信息 示例值：["/path"]
PostStart	String	否	启动后立即执行的脚本 示例值：post-start-shell-xxx
PreStop	String	否	停止前执行的脚本 示例值：pre-stop-shell-xxx
Liveness	HealthCheckConfig	否	存活探针配置 注意：此字段可能返回 null，表示取不到有效值。
Readiness	HealthCheckConfig	否	就绪探针配置 注意：此字段可能返回 null，表示取不到有效值。
HorizontalAutoscaler	Array of HorizontalAutoscaler	否	弹性策略
CronHorizontalAutoscaler	Array of CronHorizontalAutoscaler	否	定时弹性策略
Zones	Array of String	否	应用实际可用区 示例值：["ap-shanghai-2"]
LastDeployDate	String	否	最新部署时间 示例值：2024-12-09 10:40:31
LastDeploySuccessDate	String	否	最新部署成功时间 示例值：LastDeploySuccessDate
NodeInfos	Array of NodeInfo	否	应用所在node信息
ImageType	Integer	否	image类型 -0 为demo -1为正常image

名称	类型	必选	描述
			示例值：0
EnableTracing	Integer	否	是否启用调用链组件 示例值：1
EnableTracingReport	Integer	否	是否开启调用链上报，只有 EnableTracing=1 时生效（参数已弃用） 示例值：1
RepoType	Integer	否	镜像类型：0-个人镜像、1-企业镜像、2-公有镜像 示例值：2
BatchDeployStatus	String	否	分批发布子状态：batch_updating、batch_updating_waiting_confirm 示例值：batch_updating
ApmInstanceld	String	否	APM 资源 ID 示例值：apm-xxx
WorkloadInfo	WorkloadInfo	否	工作负载信息
SpeedUp	Boolean	否	是否启用应用加速 示例值：true
StartupProbe	HealthCheckConfig	否	启动检测探针配置 注意：此字段可能返回 null，表示取不到有效值。
OsFlavour	String	否	操作系统版本，可选参数： - ALPINE - CENTOS 示例值：ALPINE
RepoServer	String	否	镜像仓库server 示例值：tem-sh-custom-xxxxxx.tencentcloudcr.com
UnderDeploying	Boolean	否	是否正在发布中 示例值：true
EnablePrometheusConf	EnablePrometheusConf	否	监控业务指标监控 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Port":9090,"Path":"/data"}
StoppedManually	Boolean	否	是否为手动停止 示例值：true
TcrInstanceld	String	否	tcr实例ID 示例值：tcr-xxxxxx
EnableMetrics	Integer	否	1：开始自动metrics采集（open-telemetry）； 0：关闭metrics采集； 示例值：1
Appld	String	否	用户Appld 示例值：130xxxxxx
SubAccountUin	String	否	用户SubAccountUin 示例值：10000xxxxx
Uin	String	否	用户Uin 示例值：10001xxxxx
Region	String	否	地域 示例值：ap-shanghai
Groupld	String	否	应用分组ID 示例值：group-xxxxxx
EnableRegistry	Integer	否	是否启用注册中心 示例值：1
AutoscalerList	Array of Autoscaler	否	弹性伸缩数组

名称	类型	必选	描述
Modifier	String	否	修改人 示例值：100019xxxxxx
Creator	String	否	创建人 示例值：100019xxxxxx
DeployStrategyConf	DeployStrategyConf	否	部署策略 注意：此字段可能返回 null，表示取不到有效值。
PodList	DescribeRunPodPage	否	实例列表
ConfEdited	Boolean	否	发布时配置是否有修改 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
Tags	Array of Tag	否	标签
PreStopEncoded	String	否	是否编码 示例值：PreStopEncoded-xxx
PostStartEncoded	String	否	是否编码 示例值：PostStartEncoded-xxx

UseDefaultRepoParameters

创建应用，创建仓库参数

被如下接口引用：CreateApplication。

名称	类型	必选	描述
EnterpriseInstanceName	String	否	企业版实例名 示例值：tcr-demo-xxx
EnterpriseInstanceChargeType	Integer	否	企业版收费类型 0 按量收费 1 包年包月 示例值：0
EnterpriseInstanceType	String	否	企业版规格：basic-基础班，standard-标准版，premium-高级版 示例值：basic

WorkloadInfo

工作负载详情

被如下接口引用：DescribeApplicationInfo, DescribeApplicationsStatus。

名称	类型	必选	描述
ClusterId	String	否	资源 ID 示例值：cls-xxx
ApplicationName	String	否	应用名 示例值：appName-xxx
VersionName	String	否	版本名称 示例值：version-name-xxx
ReadyReplicas	Integer	否	Ready实例数 示例值：1
Replicas	Integer	否	实例数 示例值：2
UpdatedReplicas	Integer	否	Updated实例数 示例值：1
UpdatedReadyReplicas	Integer	否	UpdatedReady实例数 示例值：1

名称	类型	必选	描述
UpdateRevision	String	否	更新版本 示例值：revision-xxxxxx
CurrentRevision	String	否	当前版本 示例值：revision-xxxxxx

错误码

最近更新时间：2024-12-20 01:56:46

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。
NoSuchProduct	产品不存在

错误码	说明
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.ActionReadTimeout	请求响应超时。
FailedOperation.CreateServiceError	创建服务失败。
FailedOperation.DefaultInternalError	服务器繁忙,请稍后再试。
FailedOperation.DeleteServiceError	删除应用失败。
FailedOperation.DescribeIngressListError	查询 ingress 列表失败。
FailedOperation.DescribeRunPodListError	查询实例信息失败。
FailedOperation.DescribeServiceError	查询service失败。
FailedOperation.DescribeServiceListError	查询service列表失败。
FailedOperation.UpdateIngressError	更新 ingress 失败。
InternalError.ActionReadTimeout	请求响应超时。
InternalError.AddNewNodeError	添加子网/虚拟节点异常。
InternalError.CreateApmResourceError	创建apm资源失败。
InternalError.CreateConfigDataError	创建配置失败。
InternalError.CreateEksClusterError	底层集群创建失败。
InternalError.CreateLogConfigError	创建 LogConfig 错误。
InternalError.CreateServiceError	创建服务失败。
InternalError.DefaultInternalError	服务器繁忙,请稍后再试。

错误码	说明
InternalServerError.DeleteIngressError	删除 ingress 失败。
InternalServerError.DeleteLogConfigError	删除 LogConfig 错误。
InternalServerError.DeleteServiceError	删除应用失败。
InternalServerError.DeployVersionError	版本部署调用失败。
InternalServerError.DescribeConfigDataError	查询配置详情失败。
InternalServerError.DescribeConfigDataListError	查询配置列表失败。
InternalServerError.DescribeIngressError	查询 ingress 失败。
InternalServerError.DescribeIngressListError	查询 ingress 列表失败。
InternalServerError.DescribeLogConfigError	查询 LogConfig 错误。
InternalServerError.DescribeLogConfigListError	查询 LogConfig 列表错误。
InternalServerError.DescribeRunPodListError	查询实例信息失败。
InternalServerError.DescribeServiceError	查询service失败。
InternalServerError.DescribeServiceIngressError	查询服务关联的 ingress 失败。
InternalServerError.DescribeServiceListError	查询service列表失败。
InternalServerError.ModifyConfigDataError	修改配置失败。
InternalServerError.ModifyLogConfigError	修改 LogConfig 错误。
InternalServerError.RestartApplicationError	重启失败。
InternalServerError.StopApplicationError	停止应用失败。
InternalServerError.TagInterfaceError	调用标签接口异常。
InternalServerError.UpdateIngressError	更新 ingress 失败。
InvalidParameter.ApplicationAccessServiceReachMaximum	应用访问配置服务数量达到上限。
InvalidParameter.LBServiceCannotSupportTcpUdpSameTime	LB类型服务不能同时支持TCP和UDP。
InvalidParameter.MustProvidePortMappingRules	必须提供PortMapping规则。
InvalidParameter.ServiceNameNotValid	服务名不符合规范。
InvalidParameter.ServiceUseReserveSuffix	服务名使用了保留后缀。
InvalidParameter.TooManyPortMappingRules	PortMapping规则数量超过限制。
InvalidParameter.UnauthorizedOrMissingRole	权限不足或缺少相关角色
InvalidParameter.VpcOverQuota	自动创建将超过 vpc 限额。
InvalidParameterValue.ApmNotBind	APM 没有与当前环境绑定。
InvalidParameterValue.ApplicationAccessServiceReachMaximum	应用访问配置服务数量达到上限。
InvalidParameterValue.ApplicationServiceAlreadyExist	已存在同名访问方式。
InvalidParameterValue.ApplicationServiceNotFound	没有找到该访问方式。
InvalidParameterValue.AtLeastOneScalerRuleShouldBeApplied	请至少启用一种弹性规则。
InvalidParameterValue.AutoScalerLargerThanOne	只支持绑定一种弹性伸缩。
InvalidParameterValue.AutoScalerNameInvalid	弹性规则名称不合法, 长度必须小于20。
InvalidParameterValue.CannotOverWriteOtherApplicationService	不能覆盖其他应用的访问方式。
InvalidParameterValue.CannotUpdateServiceByBothMethods	不能同时使用全量以及单条更新。

错误码	说明
InvalidParameterValue.ConfigDataAlreadyExist	配置已存在。
InvalidParameterValue.ConfigDataInvalid	配置不合法。
InvalidParameterValue.CreateNamespaceUnsupportRegionError	创建环境createRegion参数错误。
InvalidParameterValue.CronHpaReplicasInvalid	定时弹性伸缩目标实例数不合法。
InvalidParameterValue.DailyCreateNamespaceReachMaximum	每日环境创建次数超过上限。
InvalidParameterValue.DisableScalerBeforeDelete	弹性伸缩启用中，请停用后再删除。
InvalidParameterValue.EnvironmentNameImmutable	环境名称不可变。
InvalidParameterValue.HpaMetricsInvalid	弹性伸缩指标不合法。
InvalidParameterValue.HpaMinMaxInvalid	弹性伸缩最小值/最大值不合法。
InvalidParameterValue.HpaThresholdInvalid	弹性伸缩阈值不合法。
InvalidParameterValue.IngressRewriteRequiredHttpsEnable	访问配置重定向需要启用 HTTPS。
InvalidParameterValue.InvalidCronScalerPeriod	无效的定时伸缩周期。
InvalidParameterValue.InvalidDeployVersion	版本号格式非法。
InvalidParameterValue.InvalidEksServiceType	访问类型不支持。
InvalidParameterValue.InvalidEnvName	环境变量名非法，要求有由字母、数字，".", "_" 和 "-" 组成，不能由数字开头。
InvalidParameterValue.InvalidEnvValue	环境变量长度不能超过1000。
InvalidParameterValue.InvalidMountPath	挂载路径不合法，不能为 /app。
InvalidParameterValue.InvalidServiceName	应用名格式非法。
InvalidParameterValue.InvalidTenantInfo	无法识别租户信息。
InvalidParameterValue.JdkVersionRequired	JDK 版本不能为空。
InvalidParameterValue.LogConfigAlreadyExist	LogConfig 配置已存在。
InvalidParameterValue.MustProvidePortMappingRules	必须提供PortMapping规则。
InvalidParameterValue.NamespaceDuplicateError	环境重复。
InvalidParameterValue.NamespaceNotBelongToAppid	命名空间不属于用户。
InvalidParameterValue.NamespaceNotFound	环境不存在。
InvalidParameterValue.NamespaceReachMaximum	环境创建失败，达到上限。
InvalidParameterValue.NamespaceResourceReachMaximum	环境资源创建失败，达到上限。
InvalidParameterValue.OsNotSupport	操作系统不支持。
InvalidParameterValue.PortDuplicateError	服务端口映射重复。
InvalidParameterValue.PortIsReserved	服务端口是保留端口。
InvalidParameterValue.PostStartNotValid	post start 不合法。
InvalidParameterValue.PreStopNotValid	pre stop 不合法。
InvalidParameterValue.PublicRepoTypeParameterError	公有镜像参数错误。
InvalidParameterValue.RegistryNotBind	注册中心没有与当前环境绑定。
InvalidParameterValueScalerNameDuplicated	弹性伸缩名称已存在。
InvalidParameterValue.ServiceFoundRunningVersion	应用存在正在运行的实例。
InvalidParameterValue.ServiceLowerCase	服务名必须小写。

错误码	说明
InvalidParameterValue.ServiceNameDuplicateError	应用名已存在。
InvalidParameterValue.ServiceNotBelongToAppid	应用不属于此账户。
InvalidParameterValue.ServicePodReachMaximum	实例创建失败，达到上限。
InvalidParameterValue.ServiceReachMaximum	应用创建失败，达到上限。
InvalidParameterValue.ServiceUseReserveSuffix	服务名使用了保留后缀。
InvalidParameterValue.TcrEntInstanceNameNotValid	企业版 TCR 实例名称不合法。
InvalidParameterValue.TemIdInvalid	不是合法的TEM ID。
InvalidParameterValue.TraitsTracingNotSupported	非 JAVA 应用不支持链路追踪特性。
InvalidParameterValue.VersionLengthLimit	version 不能超过128位。
InvalidParameterValue.VersionLowerCase	版本必须小写。
InvalidParameterValue.VersionRouteRateNotZero	版本的路由流量不为0。
InvalidParameterValue.VpcInvalid	vpc错误, 与环境vpc不一致。
MissingParameter.AutoScalerNameNull	弹性规则名称不能为空。
MissingParameter.DeployModeNull	部署方式不能为空。
MissingParameter.DeployVersionNull	部署版本不能为空。
MissingParameter.EnvironmentNameNull	环境名称不能为空。
MissingParameter.ImgRepoNull	镜像仓库不能为空。
MissingParameter.LogsetOrTopicNull	logset 和 topic 不能为空。
MissingParameter.MinMaxNumNull	弹性伸缩配置最大值/最小值不能为空。
MissingParameter.NamespaceIdNull	环境ID不能为空。
MissingParameter.PkgNameNull	包名不能为空。
MissingParameter.RepoServerNull	镜像仓库服务器不能为空。
MissingParameter.ScalerIdNull	弹性规则ID不能为空。
MissingParameter.ServiceIdNull	服务ID不能为空。
MissingParameter.SvcRepoNotReady	镜像仓库还未就绪。
MissingParameter.TcrEntInstanceNameNull	企业版 TCR 实例名称不能为空。
MissingParameter.VpcServiceSubnetNull	vpc内访问类型子网ID不能为空。
OperationDenied.BalanceNotEnough	账户余额不足。
OperationDenied.ResourceIsolated	账号欠费状态下不支持该操作，请冲正后重试。
ResourceInUse.EnvironmentAlreadyLocked	环境已锁定。
ResourceInUse.ResourceAlreadyLocked	你操作的资源已被其他操作占用，请稍后重试。
ResourceInUse.ResourceAlreadyUsed	资源已绑定。
ResourceInUse.ServiceDeploying	服务部署中。
ResourceNotFound.ConfigDataNotFound	配置不存在。
ResourceNotFound.InterfaceNotFound	未提供该接口。
ResourceNotFound.LogConfigNotFound	日志配置不存在。
ResourceNotFound.MicroserviceOffline	目标微服务已离线。

错误码	说明
ResourceNotFound.NamespaceNotFound	环境不存在。
ResourceNotFound.ServiceNotFound	找不到应用。
ResourceNotFound.ServiceRunningVersionNotFound	找不到运行的服务实例。
ResourceNotFound.VersionNamespaceNotFound	找不到版本对应的环境。
ResourceNotFound.VersionServiceNotFound	找不到版本对应的应用。
ResourceUnavailable.ApplicationNotDeletable	有资源依赖，无法直接删除应用。
ResourceUnavailable.ApplicationStopped	应用已停止。
ResourceUnavailable.WaitForKruise	等待组件安装。
UnauthorizedOperation.MissingEksLogRole	缺少容器服务的 CLS 日志角色，请打开控制台授权。
UnauthorizedOperation.UnauthorizedOperation	未授权。
UnsupportedOperation.UnsupportAction	不支持的ACTION。