

批量计算 API 文档



腾讯云

【版权声明】

©2013–2025 腾讯云版权所有

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

【商标声明】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

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

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

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

文档目录

API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

签名方法 v3

签名方法

返回结果

参数类型

计算环境相关接口

添加实例到计算环境

创建计算环境

删除计算环境

获取计算环境详情

查看计算环境活动信息

查看计算环境的创建信息

查看计算环境创建信息列表

获取计算环境列表

从计算环境移出实例

修改计算环境

销毁计算节点

批量销毁计算节点

任务模板相关接口

创建任务模板

删除任务模板

获取任务模板

修改任务模板

作业相关接口

删除作业

查看作业详情

获取作业的提交信息

查看作业列表

查询任务详情

获取任务日志详情

重试作业

提交作业

终止作业

终止任务实例

查看配置相关接口

获取批量计算可用的CVM机型配置信息

获取批量计算可用区机型配置信息

查询实例分类信息

数据结构

API 文档

更新历史

最近更新时间：2025-04-02 01:08:37

第 57 次发布

发布时间：2025-04-02 01:08:33

本次发布包含了以下内容：

改善已有的文档。

删除接口：

- CreateCpmComputeEnv
- DescribeCpmOsInfo

删除数据结构：

- CpmVirtualPrivateCloud
- EnvDataCpm
- NamedCpmComputeEnv
- OsInfo

第 56 次发布

发布时间：2024-12-23 01:10:47

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ComputeNode](#)
 - 修改成员：ComputeNodeInstanceId, ResourceCreatedTime, AgentVersion

第 55 次发布

发布时间：2024-12-20 01:06:29

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Activity](#)
 - 修改成员：EndTime
- [JobView](#)
 - 修改成员：Placement, EndTime
- [TaskInstanceView](#)
 - 修改成员：ExitCode, ComputeNodeInstanceId, LaunchTime, RunningTime, EndTime
- [TaskView](#)
 - 修改成员：EndTime

第 54 次发布

发布时间：2024-12-10 01:11:02

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Docker](#)
 - 修改成员：User, Password

第 53 次发布

发布时间：2024-12-02 01:08:07

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DataDisk](#)
 - 新增成员：DiskName
- [SystemDisk](#)
 - 新增成员：DiskName

第 52 次发布

发布时间：2023-12-01 01:07:51

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DataDisk](#)
 - 新增成员：BurstPerformance

第 51 次发布

发布时间：2023-10-09 01:07:08

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InstanceTypeQuotaItem](#)
 - 新增成员：StatusCategory

第 50 次发布

发布时间：2023-09-15 15:00:54

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Placement](#)
 - 删除成员：HostIps

第 49 次发布

发布时间：2023-08-25 01:09:47

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [CommandLine](#)

修改数据结构：

- [Application](#)
 - 新增成员：Commands

第 48 次发布

发布时间：2023-08-17 01:18:33

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Application](#)
 - 修改成员：Command

第 47 次发布

发布时间：2023-07-19 01:07:48

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [StorageBlock](#)
 - 修改成员：Type, MinSize, MaxSize

第 46 次发布

发布时间：2023-07-18 01:08:15

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ItemPrice](#)
 - 修改成员：UnitPrice, ChargeUnit, OriginalPrice, DiscountPrice, Discount, UnitPriceDiscount, UnitPriceSecondStep, UnitPriceDiscountSecondStep, UnitPriceThirdStep, UnitPriceDiscountThirdStep, OriginalPriceThreeYear, DiscountPriceThreeYear, DiscountThreeYear, OriginalPriceFiveYear, DiscountPriceFiveYear, DiscountFiveYear, OriginalPriceOneYear, DiscountPriceOneYear, DiscountOneYear

第 45 次发布

发布时间：2022-12-09 01:11:02

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [OutputMappingOption](#)

修改数据结构：

- [OutputMapping](#)
 - 新增成员：OutputMappingOption

第 44 次发布

发布时间：2022-12-07 06:17:30

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InstanceTypeQuotaItem](#)
 - 新增成员：GpuCount, Frequency

第 43 次发布

发布时间：2022-12-05 06:18:51

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Docker](#)
 - 新增成员：DockerRunOption

第 42 次发布

发布时间：2022-08-25 06:10:07

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Docker](#)
 - 新增成员：MaxRetryCount, DelayOnRetry

第 41 次发布

发布时间：2022-03-31 06:03:07

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Job](#)
 - 新增成员：NotificationTarget
- [NamedComputeEnv](#)
 - 新增成员：NotificationTarget
- [NamedCpmComputeEnv](#)
 - 新增成员：NotificationTarget

第 40 次发布

发布时间：2022-03-29 08:04:27

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [NamedComputeEnv](#)
 - 修改成员：Notifications

第 39 次发布

发布时间：2022-02-18 08:04:38

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ItemPrice](#)
 - 修改成员：Discount, UnitPriceDiscount, UnitPriceSecondStep, UnitPriceDiscountSecondStep, UnitPriceThirdStep, UnitPriceDiscountThirdStep, OriginalPriceThreeYear, DiscountPriceThreeYear, DiscountThreeYear, OriginalPriceFiveYear, DiscountPriceFiveYear, DiscountFiveYear, OriginalPriceOneYear, DiscountPriceOneYear, DiscountOneYear

第 38 次发布

发布时间：2021-08-25 08:03:12

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RunAutomationServiceEnabled](#)
 - 新增成员：Enabled

第 37 次发布

发布时间：2021-08-18 08:03:13

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [RunAutomationServiceEnabled](#)

修改数据结构：

- [EnhancedService](#)
 - 新增成员：AutomationService

第 36 次发布

发布时间：2021-02-24 08:01:57

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DataDisk](#)
 - 新增成员：Cdcd
- [SystemDisk](#)
 - 新增成员：Cdcd

第 35 次发布

发布时间：2021-01-15 08:01:53

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DataDisk](#)
 - 新增成员：ThroughputPerformance

第 34 次发布

发布时间：2020-10-27 08:02:00

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ItemPrice](#)
 - 新增成员：OriginalPriceThreeYear, DiscountPriceThreeYear, DiscountThreeYear, OriginalPriceFiveYear, DiscountPriceFiveYear, DiscountFiveYear, OriginalPriceOneYear, DiscountPriceOneYear, DiscountOneYear

第 33 次发布

发布时间：2020-10-20 08:01:43

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeJob](#)
 - 新增出参：NextAction
 - 修改出参：TaskInstanceMetrics

第 32 次发布

发布时间：2020-09-23 08:01:24

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateTaskTemplate](#)
 - 新增入参：Tags
- [DescribeComputeEnv](#)
 - 新增出参：Tags
- [DescribeComputeEnvCreateInfo](#)
 - 新增出参：Tags
- [DescribeJob](#)
 - 新增出参：Tags
- [DescribeJobSubmitInfo](#)
 - 新增出参：Tags

新增数据结构：

- [Tag](#)

修改数据结构：

- [ComputeEnvCreateInfo](#)
 - 新增成员：Tags
- [ComputeEnvView](#)
 - 新增成员：Tags
- [Job](#)
 - 新增成员：Tags
- [JobView](#)
 - 新增成员：Tags
- [NamedComputeEnv](#)
 - 新增成员：Tags
- [NamedCpmComputeEnv](#)
 - 新增成员：Tags
- [TaskTemplateView](#)
 - 新增成员：Tags

第 31 次发布

发布时间：2020-07-07 08:04:05

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InstanceTypeQuotalItem](#)
 - 新增成员：Gpu, Fpga, Remark

第 30 次发布

发布时间：2020-07-03 08:04:08

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DataDisk](#)
 - 新增成员：KmsKeyId

第 29 次发布

发布时间：2020-06-02 08:04:00

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InstanceTypeQuotalItem](#)
 - 新增成员：InstanceBandwidth, InstancePps, StorageBlockAmount, CpuType

第 28 次发布

发布时间：2020-04-29 08:04:08

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ItemPrice](#)
 - 修改成员：Discount

第 27 次发布

发布时间：2020-03-16 08:01:47

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ItemPrice](#)
 - 新增成员：Discount, UnitPriceDiscount, UnitPriceSecondStep, UnitPriceDiscountSecondStep, UnitPriceThirdStep, UnitPriceDiscountThirdStep

第 26 次发布

发布时间：2020-03-04 17:46:21

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InstanceTypeQuotaItem](#)
 - 新增成员：SoldOutReason

第 25 次发布

发布时间：2020-02-27 19:58:14

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AttachInstances](#)
- [DetachInstances](#)

修改接口：

- [DescribeComputeEnv](#)
 - 新增出参：AttachedComputeNodeCount

新增数据结构：

- [Instance](#)

修改数据结构：

- [ComputeEnvView](#)
 - 新增成员：AttachedComputeNodeCount
- [ComputeNode](#)
 - 新增成员：ResourceOrigin
- [NamedComputeEnv](#)
 - 新增成员：ResourceMaxRetryCount
- [NamedCpmComputeEnv](#)
 - 新增成员：ResourceMaxRetryCount

第 24 次发布

发布时间：2020-01-17 15:09:37

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeComputeEnv](#)
 - 新增出参：NextAction

修改数据结构：

- [ComputeEnvView](#)
 - 新增成员：NextAction
- [DataDisk](#)
 - 新增成员：Encrypt
- [Placement](#)
 - 新增成员：HostId

第 23 次发布

发布时间：2019-12-23 23:28:59

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- DescribeCpmOsInfo

修改接口：

- [DescribeComputeEnv](#)
 - 新增出参：ResourceType

新增数据结构：

- OsInfo

修改数据结构：

- [ComputeEnvView](#)
 - 新增成员：ResourceType
- [ComputeNode](#)
 - 新增成员：ResourceType
- [VirtualPrivateCloud](#)
 - 新增成员：Ipv6AddressCount

第 22 次发布

发布时间：2019-11-22 11:47:05

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- EnvDataCpm
 - 新增成员：InstanceTypes, TimeUnit, TimeSpan, RaidId, NeedSecurityAgent, NeedMonitorAgent, AutoRenewFlag, IsZoning, FileSystem, Password, ApplyEip, EipPayMode, EipBandwidth, ImageId, SysRootSpace, SysDataSpace, HyperThreading, LanIps
 - 删除成员：ComputeTypes, DeployType, SpotStrategy, Passwd
 - 修改成员：OsTypeId
- [LocalDiskType](#)
 - 新增成员：Required

第 21 次发布

发布时间：2019-09-20 15:39:29

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Task](#)
 - 新增成员：ResourceMaxRetryCount

第 20 次发布

发布时间：2019-08-23 11:24:11

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- CreateCpmComputeEnv

新增数据结构：

- CpmVirtualPrivateCloud
- EnvDataCpm
- NamedCpmComputeEnv

第 19 次发布

发布时间：2019-05-30 20:25:09

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [DataDisk](#)
 - 新增成员：SnapshotId
- [Placement](#)
 - 新增成员：HostIps
- [Task](#)
 - 新增成员：RestartComputeNode

第 18 次发布

发布时间：2019-05-09 19:41:49

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ComputeNodeMetrics](#)
 - 修改成员：SubmittedCount, CreatingCount, CreationFailedCount, CreatedCount, RunningCount, DeletingCount, AbnormalCount

第 17 次发布

发布时间：2019-04-29 17:18:16

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeInstanceCategories](#)

新增数据结构：

- [InstanceCategoryItem](#)

修改数据结构：

- [EnvData](#)
 - 新增成员：Zones, VirtualPrivateClouds

第 16 次发布

发布时间：2019-04-12 11:18:36

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- [InstanceTypeOptions](#)

修改数据结构：

- [EnvData](#)
 - 新增成员：InstanceTypeOptions
- [InstanceMarketOptionsRequest](#)
 - 修改成员：SpotOptions
- [InternetAccessible](#)
 - 新增成员：BandwidthPackageId
- [Task](#)
 - 新增成员：MaxConcurrentNum

第 15 次发布

发布时间：2018-12-20 20:36:52

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [RetryJobs](#)

第 14 次发布

发布时间：2018-11-22 19:46:59

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Activity](#)
 - 新增成员：InstanceId

第 13 次发布

发布时间：2018-11-08 19:12:59

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeTaskLogs](#)

新增数据结构：

- [TaskInstanceLog](#)

第 12 次发布

发布时间：2018-10-26 14:33:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyComputeEnv](#)
 - 新增入参：EnvName, EnvDescription, EnvData
 - 修改入参：DesiredComputeNodeCount

新增数据结构：

- [ComputeEnvData](#)

第 11 次发布

发布时间：2018-10-19 13:22:41

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeTask](#)
 - 新增入参：Offset, Limit, Filters

第 10 次发布

发布时间：2018-10-11 15:27:31

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeCvmZoneInstanceConfigInfos](#)
 - 修改入参：Filters

第 9 次发布

发布时间：2018-09-25 14:47:59

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [EnvData](#)
 - 新增成员：InstanceTypes

第 8 次发布

发布时间：2018-09-21 11:23:55

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeJobs](#)
 - 修改出参：JobSet

第 7 次发布

发布时间：2018-09-20 10:49:02

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [Task](#)
 - 修改成员：EnvVars, Authentications

第 6 次发布

发布时间：2018-09-14 14:37:34

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [AnonymousComputeEnv](#)
 - 修改成员： EnvType, EnvData
- [DataDisk](#)
 - 新增成员： DeleteWithInstance
- [EnvData](#)
 - 修改成员： InstanceType, ImageId
- [Job](#)
 - 修改成员： JobName, Priority
- [NamedComputeEnv](#)
 - 修改成员： EnvType, EnvData
- [Task](#)
 - 修改成员： TaskName, TaskInstanceNum, RedirectInfo, MaxRetryCount, Timeout

第 5 次发布

发布时间：2018-08-17 15:26:03

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [InstanceTypeConfig](#)
 - 新增成员： Mem, Cpu
 - 删除成员： GPU, CPU, Memory, CbsSupport, InstanceTypeState
 - 修改成员： InstanceType, Zone, InstanceFamily

第 4 次发布

发布时间：2018-08-02 15:48:13

本次发布包含了以下内容：

改善已有的文档。

第 3 次发布

发布时间：2018-07-26 15:15:22

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeCvmZoneInstanceConfigInfos](#)

新增数据结构：

- [Externals](#)
- [InstanceMarketOptionsRequest](#)
- [InstanceTypeQuotaItem](#)
- [ItemPrice](#)
- [LocalDiskType](#)
- [SpotMarketOptions](#)
- [StorageBlock](#)

修改数据结构：

- [EnvData](#)
 - 新增成员： InstanceChargeType, InstanceMarketOptions

第 2 次发布

发布时间：2018-05-24 17:08:50

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [TerminateComputeNodes](#)

修改接口：

- [DescribeJob](#)
 - 新增出参：StateReason

修改数据结构：

- [ComputeNode](#)
 - 新增成员：PrivateIpAddresses, PublicIpAddresses
- [Job](#)
 - 新增成员：StateIfCreateCvmFailed
- [NamedComputeEnv](#)
 - 新增成员：ActionIfComputeNodeInactive
- [TaskInstanceView](#)
 - 新增成员：StateDetailedReason

第 1 次发布

发布时间：2018-04-24

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateComputeEnv](#)
- [CreateTaskTemplate](#)
- [DeleteComputeEnv](#)
- [DeleteJob](#)
- [DeleteTaskTemplates](#)
- [DescribeAvailableCvmInstanceTypes](#)
- [DescribeComputeEnv](#)
- [DescribeComputeEnvActivities](#)
- [DescribeComputeEnvCreateInfo](#)
- [DescribeComputeEnvCreateInfos](#)
- [DescribeComputeEnvs](#)
- [DescribeJob](#)
- [DescribeJobSubmitInfo](#)
- [DescribeJobs](#)
- [DescribeTask](#)
- [DescribeTaskTemplates](#)
- [ModifyComputeEnv](#)
- [ModifyTaskTemplate](#)
- [SubmitJob](#)
- [TerminateComputeNode](#)
- [TerminateJob](#)
- [TerminateTaskInstance](#)

新增数据结构：

- [Activity](#)

- [AgentRunningMode](#)
- [AnonymousComputeEnv](#)
- [Application](#)
- [Authentication](#)
- [ComputeEnvCreateInfo](#)
- [ComputeEnvView](#)
- [ComputeNode](#)
- [ComputeNodeMetrics](#)
- [DataDisk](#)
- [Dependence](#)
- [Docker](#)
- [EnhancedService](#)
- [EnvData](#)
- [EnvVar](#)
- [EventConfig](#)
- [EventVar](#)
- [Filter](#)
- [InputMapping](#)
- [InstanceTypeConfig](#)
- [InternetAccessible](#)
- [Job](#)
- [JobView](#)
- [LoginSettings](#)
- [MountDataDisk](#)
- [NamedComputeEnv](#)
- [Notification](#)
- [OutputMapping](#)
- [OutputMappingConfig](#)
- [Placement](#)
- [RedirectInfo](#)
- [RedirectLocalInfo](#)
- [RunMonitorServiceEnabled](#)
- [RunSecurityServiceEnabled](#)
- [SystemDisk](#)
- [Task](#)
- [TaskInstanceMetrics](#)
- [TaskInstanceView](#)
- [TaskMetrics](#)
- [TaskTemplateView](#)
- [TaskView](#)
- [VirtualPrivateCloud](#)

简介

最近更新时间：2025-04-25 01:09:26

欢迎使用批量计算（BatchCompute，简称 Batch）。

批量计算（Batch）是应对海量批处理作业的分布式计算平台，目的是给有大数据计算业务的企业、科研单位等提供高性价比且易用的计算服务。Batch 会智能的对任意规模批处理作业进行资源分配和作业执行管理，您可以将精力集中在如何分析和处理数据上。Batch 非常适合基因测序、医疗分析、科学研究、视频渲染、金融大数据、精准广告等领域。

作为一项完全安全的托管服务，Batch 使开发人员、科学家和工程师能够运行任意规模的批处理计算作业。利用 Batch，您仅提供计算执行的环境、命令以及对应输入输出数据存放地址，Batch 会根据工作负载去获取腾讯云的弹性资源，并自动调度作业执行流程以完成成千上万的批处理作业。Batch 可以降低运营的复杂性、节省时间和减少成本，从而使开发人员、科学家和工程师能够在腾讯云中轻松运行其批处理作业，也使企业、科研机构等不再需要预支巨额的费用去建设批处理数据中心。

注意：

本章节批量计算 API 接口均为最新 API 3.0 接口，后续 Batch 相关新增功能都会在此章节更新。我们强烈推荐您使用最新 API 3.0 接口。

API 概览

最近更新时间：2025-05-21 01:08:42

计算环境相关接口

接口名称	接口功能	频率限制（次/秒）
AttachInstances	添加实例到计算环境	20
CreateComputeEnv	创建计算环境	2
DeleteComputeEnv	删除计算环境	2
DescribeComputeEnv	获取计算环境详情	2
DescribeComputeEnvActivities	查看计算环境活动信息	2
DescribeComputeEnvCreateInfo	查看计算环境的创建信息	2
DescribeComputeEnvCreateInfos	查看计算环境创建信息列表	20
DescribeComputeEnvs	获取计算环境列表	10
DetachInstances	从计算环境移出实例	20
ModifyComputeEnv	修改计算环境	2
TerminateComputeNode	销毁计算节点	20
TerminateComputeNodes	批量销毁计算节点	20

任务模板相关接口

接口名称	接口功能	频率限制（次/秒）
CreateTaskTemplate	创建任务模板	2
DeleteTaskTemplates	删除任务模板	2
DescribeTaskTemplates	获取任务模板	10
ModifyTaskTemplate	修改任务模板	2

作业相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteJob	删除作业	30
DescribeJob	查看作业详情	50
DescribeJobSubmitInfo	获取作业的提交信息	20
DescribeJobs	查看作业列表	10
DescribeTask	查询任务详情	20
DescribeTaskLogs	获取任务日志详情	20
RetryJobs	重试作业	10
SubmitJob	提交作业	20
TerminateJob	终止作业	10
TerminateTaskInstance	终止任务实例	2

查看配置相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeAvailableCvmInstanceTypes	获取批量计算可用的CVM机型配置信息	2
DescribeCvmZoneInstanceConfigInfos	获取批量计算可用区机型配置信息	5
DescribeInstanceCategories	查询实例分类信息	20

⚠ 注意：

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

调用方式

请求结构

最近更新时间：2025-05-21 01:08:42

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

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

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

2. 通信协议

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

3. 请求方法

支持的 HTTP 请求方法:

- POST (推荐)
- GET

POST 请求支持的 Content-Type 类型:

- application/json (推荐), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。
- application/x-www-form-urlencoded, 必须使用签名方法 v1 (HmacSHA1 或 HmacSHA256)。
- multipart/form-data (仅部分接口支持), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。

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

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2024-11-15 01:10:16

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

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

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

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

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

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKID***/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKID*** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 batch；tc3_request 为固定字符串； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要，计算过程详见 文档 。
Token	String	否	HTTP 请求头：X-TC-Token。即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表中的前十个，接口参数设置为偏移量 Offset=0，返回数量 Limit=10，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例:

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

Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```


HTTP POST（application/json）请求结构示例：

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

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

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

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

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

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

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

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

10
--58731222010402--
```

使用签名方法 v1 的公共参数

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

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	—	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 注意： 某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。

参数名称	类型	必选	描述
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbe224158d66e7ae5fcadb70b2d181d02&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=37ac2f4fde00b0ac9bd9eadeb459b1bbe224158d66e7ae5fcadb70b2d181d02&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
西南地区（重庆）	ap-chongqing
华南地区（广州）	ap-guangzhou
港澳台地区（中国香港）	ap-hongkong
华东地区（南京）	ap-nanjing
亚太东北（首尔）	ap-seoul
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
美国东部（弗吉尼亚）	na-ashburn
美国西部（硅谷）	na-siliconvalley

签名方法 v3

最近更新时间：2024-12-25 01:11:15

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区分例列表作为例子，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, 否则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]\"}";

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

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

NodeJS

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

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

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

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

function main(){
```

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

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

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

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

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

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

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

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

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

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

C++

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

using namespace std;

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

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

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

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

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

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

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

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

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

    return (ss.str());
}

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

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

```
}
return output;
}

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

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

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

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

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

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

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

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

C

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

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

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

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

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

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

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

}

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

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

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

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

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

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

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

// ***** 步骤 2: 拼接待签名字符串 *****
const char* algorithm = "TC3-HMAC-SHA256";
char request_timestamp[16] = {0};
sprintf(request_timestamp, "%d", timestamp);
char credential_scope[64] = {0};
strcat(credential_scope, date);
sprintf(credential_scope, "%s/%s/tc3-request", date, service);
char hashed_canonical_request[100] = {0};
sha256_hex(canonical_request, hashed_canonical_request);
char string_to_sign[256] = {0};
sprintf(string_to_sign, "%s\n%s\n%s\n%s", algorithm, request_timestamp,
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s", "TC3", SECRET_KEY);
unsigned char k_date[64] = {0};
unsigned int output_len = 0;
hmac_sha256(k_key, strlen(k_key), date, strlen(date), k_date, &output_len);
unsigned char k_service[64] = {0};
hmac_sha256(k_date, output_len, service, strlen(service), k_service, &output_len);
unsigned char k_signing[64] = {0};
hmac_sha256(k_service, output_len, "tc3-request", strlen("tc3-request"), k_signing, &output_len);
unsigned char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, output_len, string_to_sign, strlen(string_to_sign), k_hmac_sha_sign, &output_len);

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



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

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

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

其他语言

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

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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

签名方法

最近更新時間：2024-12-25 01:11:16

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：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

有了安全凭证 SecretId 和 SecretKey 后，就可以生成签名串了。以下是使用签名方法 v1 生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

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

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

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

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

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

2.1. 对参数排序

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

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

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

2.2. 拼接请求字符串

此步骤生成请求字符串。

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

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

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

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

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

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

示例的拼接结果为：

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

2.4. 生成签名串

此步骤生成签名串。

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

具体代码如下，以 PHP 语言为例：

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

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

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

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

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

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

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

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

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

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

5. 签名演示

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

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

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

- [Signature Demo](#)

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

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

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

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

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

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

Java

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

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

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

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

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

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

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

Python

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

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

import requests

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

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

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

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

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

Golang

```
package main

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

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

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

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

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

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

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

PHP

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

ksort($params);

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

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

Ruby

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

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



```
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

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

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

DotNet

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

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

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

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

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

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

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

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

NodeJS

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

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

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

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

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

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

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

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

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

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

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

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

返回结果

最近更新时间：2024-03-12 01:09:53

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 06:14:52

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。

计算环境相关接口

添加实例到计算环境

最近更新时间：2025-05-28 01:08:19

1. 接口描述

接口请求域名：`batch.tencentcloudapi.com`。

此接口可将已存在实例添加到计算环境中。

实例需要满足如下条件：

- 1.实例不在批量计算系统中。
- 2.实例状态要求处于运行中。
- 3.支持预付费实例，按小时后付费实例，专享子机实例。不支持竞价实例。

此接口会将加入到计算环境中的实例重设UserData和重装操作系统。

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：AttachInstances。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：env-nuydksjj
Instances.N	是	Array of Instance	加入计算环境实例列表，每次请求的实例的上限为100。

3. 输出参数

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

4. 示例

示例1 添加实例到计算环境

输入示例

```
https://batch.tencentcloudapi.com/?Action=AttachInstances
&EnvId=env-nuydksjj
&Instances.0.InstanceId=ins-dus5t7tu
&Instances.0.ImageId=img-8toqc6s3
```

```
&Instances.0.LoginSettings.Password=xxxxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "a402b86a-5b45-4edf-b64c-d233c2984fbd"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalError	内部错误。
InternalError.CallCvm	调用CVM API返回错误。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
InvalidParameterValue.InstanceIdDuplicated	实例ID重复。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
UnsupportedOperation	操作不支持。
UnsupportedOperation.InstancesNotAllowedToAttach	不允许将该实例添加到计算环境中。

创建计算环境

最近更新时间：2025-05-15 01:10:02

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于创建计算环境

默认接口请求频率限制：2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：CreateComputeEnv。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ComputeEnv	是	NamedComputeEnv	计算环境信息
Placement	是	Placement	位置信息
ClientToken	否	String	用于保证请求幂等性的字符串。该字符串由用户生成，需保证不同请求之间唯一，最大值不超过64个ASCII字符。若不指定该参数，则无法保证请求的幂等性。 示例值：19c018a8-e17b-4073-b7fb-84b56cb890f9

3. 输出参数

参数名称	类型	描述
EnvId	String	计算环境ID 示例值：env-2u17q5bl
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 创建计算环境

输入示例

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

{
  "ComputeEnv": {
    "EnvName": "test compute env",
    "EnvDescription": "EnvDescription",
```

```
"EnvType": "MANAGED",
"EnvData": {
  "InstanceType": "S1.SMALL2",
  "ImageId": "img-bd78fy2t"
},
"DesiredComputeNodeCount": 1
},
"Placement": {
  "Zone": "ap-guangzhou-2"
}
}
```

输出示例

```
{
  "Response": {
    "EnvId": "env-3dhmblr3",
    "RequestId": "b2ac2379-6453-4eab-8f63-7ade00cb67b0"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallCam	调用CAM API返回错误。
InternalServerError.CallCvm	调用CVM API返回错误。
InternalServerError.CallTagAPI	获取Tag错误。
InvalidParameter.CvmParameters	非法的CVM参数。

错误码	描述
InvalidParameter.EnvDescriptionTooLong	计算环境描述过长。
InvalidParameter.EnvNameTooLong	计算环境名称过长。
InvalidParameter.ImageIdMalformed	镜像ID不正确。
InvalidParameter.InvalidParameterCombination	非法的参数组合。
InvalidParameter.NotificationEventNameDuplicate	重复的消息通知事件名称。
InvalidParameter.NotificationTopicName	非法的主题名称。
InvalidParameter.NotificationTopicNameTooLong	主题名称过长。
InvalidParameterValue.ComputeEnv	计算环境参数校验失败。
InvalidParameterValue.InstanceType	不支持指定的实例类型。
InvalidParameterValue.InvalidDataTypeAny	DataTypeAny不合法。
InvalidParameterValue.InvalidZoneMismatchRegion	可用区和地域不匹配。
InvalidParameterValue.LocalPath	非法的本地存储路径。
InvalidParameterValue.Negative	非法的负值参数。
InvalidParameterValue.RemoteStoragePath	非法的存储路径格式。
InvalidParameterValue.RemoteStorageSchemeType	非法的存储类型。
InvalidParameterValue.UnavailableZone	Zone不可用。
InvalidZone.MismatchRegion	指定的zone不存在。
LimitExceeded.ComputeEnvQuota	计算环境配额不足。
LimitExceeded.CpuQuota	CPU配额不足。
UnauthorizedOperation.UserNotAllowedToUseBatch	禁止使用批量计算服务。

删除计算环境

最近更新时间：2025-05-28 01:08:19

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于删除计算环境

默认接口请求频率限制：2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：DeleteComputeEnv。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取，不能对状态处于删除中或者存在计算实例未销毁的环境发起删除动作。 示例值：env-3dhmblr3

3. 输出参数

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

4. 示例

示例1 删除计算环境

输入示例

```
https://batch.tencentcloudapi.com/?Action=DeleteComputeEnv
&EnvId=env-3dhmblr3
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8b71050a-55c1-4f3b-bf66-5e4f8510a5a0"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
UnsupportedOperation	操作不支持。
UnsupportedOperation.AcceptOtherRequest	处理其他请求，禁止删除操作。
UnsupportedOperation.ComputeEnvAcceptOtherRequest	计算环境正在接受请求。
UnsupportedOperation.ComputeEnvOperation	禁止删除操作。
UnsupportedOperation.ComputeNodeForbidTerminate	禁止终止计算节点。

获取计算环境详情

最近更新时间：2025-05-28 01:08:19

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查询计算环境的详细信息

默认接口请求频率限制：2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：DescribeComputeEnv。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：env-lcpcej85

3. 输出参数

参数名称	类型	描述
EnvId	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnv 获取。 示例值：env-lcpcej85
EnvName	String	计算环境名称 示例值：test compute env
Placement	Placement	位置信息
CreateTime	String	计算环境创建时间 示例值：2018-03-08T11:48:43Z
ComputeNodeSet	Array of ComputeNode	计算节点列表信息
ComputeNodeMetrics	ComputeNodeMetrics	计算节点统计指标
DesiredComputeNodeCount	Integer	计算节点期望个数 示例值：2
EnvType	String	计算环境管理类型，枚举如下：MANAGED: 由客户在Batch平台主动创建；THPC_QUEUE: 由thpc平台创建，关联thpc平台集群队列。 示例值：MANAGED
ResourceType	String	计算环境资源类型，当前为CVM和CPM（黑石） 示例值：CVM
NextAction	String	下一步的动作，枚举如下：DELETING: 删除中 示例值：DELETING
AttachedComputeNodeCount	Integer	用户添加到计算环境中的计算节点个数 示例值：0

参数名称	类型	描述
Tags	Array of Tag	计算环境绑定的标签列表。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查看计算环境详细信息

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeComputeEnv
&EnvId=env-lcpcej85
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "EnvId": "env-lcpcej85",
    "ComputeNodeMetrics": {
      "CreatedCount": 0,
      "DeletingCount": 0,
      "CreationFailedCount": 0,
      "SubmittedCount": 0,
      "CreatingCount": 0,
      "AbnormalCount": 1,
      "RunningCount": 1
    },
    "ResourceType": "CVM",
    "EnvType": "MANAGED",
    "CreateTime": "2018-03-08T11:48:43Z",
    "EnvName": "test compute env",
    "NextAction": "",
    "Placement": {
      "ProjectId": 0,
      "Zone": "ap-guangzhou-2"
    },
    "ComputeNodeSet": [
      {
        "ComputeNodeId": "node-noa8vefu",
        "ComputeNodeState": "ABNORMAL",
        "Mem": 2,
        "ResourceCreatedTime": "2018-03-08T11:49:17Z",
        "ResourceType": "CVM",
        "ResourceOrigin": "BATCH_CREATED",
        "ComputeNodeInstanceId": "ins-0kj3gz6s",
        "AgentVersion": "1.1.9",
        "TaskInstanceNumAvailable": 1,
        "Cpu": 1,
        "PublicIpAddresses": [
          "132.20.19.18"
        ],
        "PrivateIpAddresses": [
```

```
"10.10.8.3"
]
},
{
  "ComputeNodeId": "node-9yzd5jei",
  "ComputeNodeState": "RUNNING",
  "Mem": 2,
  "ResourceCreatedTime": "2018-03-09T13:15:44Z",
  "ResourceType": "CVM",
  "ResourceOrigin": "BATCH_CREATED",
  "ComputeNodeInstanceId": "ins-jyafz2sk",
  "AgentVersion": "1.1.9",
  "TaskInstanceNumAvailable": 1,
  "Cpu": 1,
  "PublicIpAddresses": [
    "132.20.19.18"
  ],
  "PrivateIpAddresses": [
    "10.10.8.3"
  ]
},
],
"DesiredComputeNodeCount": 2,
"AttachedComputeNodeCount": 0,
"Tags": [
  {
    "Key": "batch-test-tag-env-key",
    "Value": "batch-test-tag-env-value1"
  }
],
"RequestId": "274cbd4f-77d2-4bf6-9984-643318ed3ef8"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

- - - - -

6. 错误码

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

错误码	描述
InternalError	内部错误。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
ResourceNotFound.ComputeEnv	指定计算环境不存在。

查看计算环境活动信息

最近更新时间：2025-05-28 01:08:19

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查询计算环境的活动信息

默认接口请求频率限制：2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：DescribeComputeEnvActivities。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：env-lcpcej85
Offset	否	Integer	偏移量，默认为0。 示例值：0
Limit	否	Integer	返回数量，默认值20，最大值100。 示例值：100
Filters	否	Filter	过滤条件 • compute-node-id – String – 是否必填：否 –（过滤条件）按照计算节点ID过滤，节点ID通过调用接口 DescribeComputeEnv 获取。

3. 输出参数

参数名称	类型	描述
ActivitySet	Array of Activity	计算环境中的活动列表
TotalCount	Integer	活动数量 示例值：2
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 查询计算环境的活动列表

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeComputeEnvActivities
&EnvId=env-lcpcej85
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "ActivitySet": [
      {
        "ComputeNodeId": "node-9yzd5jei",
        "EnvId": "env-lcpcej85",
        "InstanceId": "ins-x23sd8xk",
        "ActivityState": "SUCCEED",
        "ActivityId": "act-83xvnt5p",
        "StartTime": "2018-03-09T13:14:34Z",
        "StateReason": "",
        "EndTime": "2018-03-09T13:15:44Z",
        "Cause": "ModifyComputeEnv: increase the capacity from 1 to 2",
        "ComputeNodeActivityType": "CREATE_COMPUTE_NODE"
      },
      {
        "ComputeNodeId": "node-noa8vefu",
        "EnvId": "env-lcpcej85",
        "InstanceId": "ins-x23sdiwd",
        "ActivityState": "SUCCEED",
        "ActivityId": "act-ezle9e7v",
        "StartTime": "2018-03-08T11:48:43Z",
        "StateReason": "",
        "EndTime": "2018-03-08T11:49:17Z",
        "Cause": "CreateComputeEnv: the capacity is 1",
        "ComputeNodeActivityType": "CREATE_COMPUTE_NODE"
      }
    ],
    "RequestId": "ecfeec77-6ec1-4daf-8730-1dd00c742252"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalError	内部错误。
InvalidParameter.ComputeNodeIdMalformed	非法的计算节点ID格式。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
UnknownParameter	未知参数错误。

查看计算环境的创建信息

最近更新时间：2025-05-30 01:10:49

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

查看计算环境的创建信息。

默认接口请求频率限制：2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：DescribeComputeEnvCreateInfo。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：env-lcpcej85

3. 输出参数

参数名称	类型	描述
EnvId	String	计算环境 ID 示例值：env-qleawxzt
EnvName	String	计算环境名称 示例值：test compute env
EnvDescription	String	计算环境描述 示例值：test compute env
EnvType	String	计算环境类型，仅支持 “MANAGED” 类型 示例值：MANAGED
EnvData	EnvData	计算环境参数
MountDataDisks	Array of MountDataDisk	数据盘挂载选项
InputMappings	Array of InputMapping	输入映射
Authentications	Array of Authentication	授权信息
Notifications	Array of Notification	通知信息
DesiredComputeNodeCount	Integer	计算节点期望个数 示例值：2
Tags	Array of Tag	计算环境绑定的标签列表 注意：此字段可能返回 null，表示取不到有效值。

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

4. 示例

示例1 查看计算环境的创建信息

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeComputeEnvCreateInfo
&EnvId=env-lcpcej85
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "578f3fae-6908-4521-ac07-17c2cffcd5ae",
    "EnvId": "env-lcx7qbbr",
    "Notifications": [
      {
        "EventConfigs": [
          {
            "EventName": "COMPUTE_ENV_CREATED_SUCCESS",
            "EventVars": [
              {
                "Name": "COMPUTE_ENV_CREATED_SUCCESS",
                "Value": "COMPUTE_ENV_CREATED_SUCCESS"
              }
            ]
          }
        ],
        "TopicName": "compute-env-topic"
      }
    ],
    "Tags": [
      {
        "Value": "demo-tag",
        "Key": "demo-env1"
      }
    ],
    "Authentications": [
      {
        "SecretKey": "AKIDUdi1*****9ThAzpuXIjNx",
        "SecretId": "Udi16*****Q9ThAzpuXIjN",
        "Scene": "COS"
      }
    ],
    "EnvData": {
      "VirtualPrivateCloud": {
        "SubnetId": "subnet-23syenxk",
        "AsVpcGateway": true,
        "Ipv6AddressCount": 1,
        "VpcId": "vpc-1xnwrwx2",
        "PrivateIpAddresses": [
```

```
"10.10.2.29"
]
},
"LoginSettings": {
  "Password": "Aa12**5678",
  "KeepImageLogin": "True",
  "KeyIds": [
    "Qa123"
  ]
},
"InstanceMarketOptions": {
  "SpotOptions": {
    "SpotInstanceType": "one-time",
    "MaxPrice": "1"
  },
  "MarketType": "spot"
},
"SystemDisk": {
  "DiskSize": 0,
  "CdcId": "1",
  "DiskId": "disk-sdfk23nw",
  "DiskType": "CLOUD_SSD"
},
"InstanceTypes": [
  "S5.SMALL2"
],
"Zones": [
  "ap-guangzhou-2"
],
"InstanceChargeType": "POSTPAID_BY_HOUR",
"EnhancedService": {
  "SecurityService": {
    "Enabled": true
  },
  "MonitorService": {
    "Enabled": true
  },
  "AutomationService": {
    "Enabled": true
  }
},
"SecurityGroupIds": [
  "sg-1234t1"
],
"InternetAccessible": {
  "PublicIpAssigned": true,
  "InternetChargeType": "POSTPAID_BY_HOUR",
  "BandwidthPackageId": "dcaew",
  "InternetMaxBandwidthOut": 0
},
"VirtualPrivateClouds": [
  {
    "SubnetId": "subnet-23syenxk",
    "AsVpcGateway": true,
    "Ipv6AddressCount": 1,
    "VpcId": "vpc-1xnwrwxw2",
    "PrivateIpAddresses": [
```

```
"10.10.2.29"
]
}
],
"ImageId": "img-3sdke3x2",
"InstanceName": "test-env",
"InstanceType": "S5.SMALL2",
"InstanceTypeOptions": {
  "Memory": 1,
  "CPU": 1,
  "InstanceCategories": [
    "ALL"
  ]
},
"DataDisks": [
{
  "DeleteWithInstance": true,
  "Encrypt": true,
  "CdcId": "2",
  "DiskType": "CLOUD_SSD",
  "ThroughputPerformance": 0,
  "KmsKeyId": "kms-213k73kx",
  "DiskSize": 0,
  "SnapshotId": "snap-fsdekxe2",
  "DiskId": "disk-ecqwbck"
}
],
"EnvDescription": "test-env",
"MountDataDisks": [
{
  "FileSystemType": "EXT4",
  "LocalPath": "/mnt"
}
],
"DesiredComputeNodeCount": 1,
"EnvName": "test-env",
"InputMappings": [
{
  "SourcePath": "cos://test-1111123424.cos.ap-guangzhou.myqcloud.com/batchtest/input/",
  "DestinationPath": "/opt/data",
  "MountOptionParameter": ""
}
],
"EnvType": "MANAGED"
}
```

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
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
ResourceNotFound.ComputeEnv	指定计算环境不存在。

查看计算环境创建信息列表

最近更新时间：2025-05-28 01:08:19

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查看计算环境创建信息列表，包括名称、描述、类型、环境参数、通知及期望节点数等。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeComputeEnvCreateInfos。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvIds.N	否	Array of String	计算环境ID列表，与Filters参数不能同时指定，最大限制100，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：["env-5o4gphir"]
Filters.N	否	Array of Filter	过滤条件 - zone - String - 是否必填：否 - （过滤条件）按照可用区过滤，可用区通过调用接口 DescribeZones获取。 - env-id - String - 是否必填：否 - （过滤条件）按照计算环境ID过滤。 - env-name - String - 是否必填：否 - （过滤条件）按照计算环境名称过滤。 与EnvIds参数不能同时指定。
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	返回数量，默认值20，最大值100。 示例值：100

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	计算环境数量 示例值：2
ComputeEnvCreateInfoSet	Array of ComputeEnvCreateInfo	计算环境创建信息列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查看计算环境创建信息列表

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeComputeEnvCreateInfos
&EnvIds.0=env-lcx7qbbr
```

&<公共请求参数>

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "RequestId": "bb976df4-42a0-41cc-82d4-69bea1007083",
    "ComputeEnvCreateInfoSet": [
      {
        "EnvId": "env-lcx7qbbr",
        "Notifications": [
          {
            "EventConfigs": [
              {
                "EventName": "COMPUTE_ENV_CREATED_SUCCESS",
                "EventVars": [
                  {
                    "Name": "COMPUTE_ENV_CREATED_SUCCESS",
                    "Value": "COMPUTE_ENV_CREATED_SUCCESS"
                  }
                ]
              }
            ],
            "TopicName": "compute-env-topic"
          }
        ],
        "Tags": [
          {
            "Value": "demo-tag",
            "Key": "demo-env1"
          }
        ],
        "Authentications": [
          {
            "SecretKey": "AKIDUdi1*****9ThAzpuXIjNx",
            "SecretId": "Udi16*****Q9ThAzpuXIjN",
            "Scene": "COS"
          }
        ],
        "EnvData": {
          "VirtualPrivateCloud": {
            "SubnetId": "subnet-23syenxk",
            "AsVpcGateway": true,
            "Ipv6AddressCount": 1,
            "VpcId": "vpc-1xnwrwxw2",
            "PrivateIpAddresses": [
              "10.10.2.29"
            ]
          },
          "LoginSettings": {
            "Password": "Aa12**5678",
            "KeepImageLogin": "True",
            "KeyIds": [
              "Qa123"
            ]
          }
        }
      }
    ]
  }
}
```

```
},
"InstanceMarketOptions": {
  "SpotOptions": {
    "SpotInstanceType": "one-time",
    "MaxPrice": "1"
  },
  "MarketType": "spot"
},
"SystemDisk": {
  "DiskSize": 0,
  "CdcId": "1",
  "DiskId": "disk-sdfk23nw",
  "DiskType": "CLOUD_SSD"
},
"InstanceTypes": [
  "S5.SMALL2"
],
"Zones": [
  "ap-guangzhou-2"
],
"InstanceChargeType": "POSTPAID_BY_HOUR",
"EnhancedService": {
  "SecurityService": {
    "Enabled": true
  },
  "MonitorService": {
    "Enabled": true
  },
  "AutomationService": {
    "Enabled": true
  }
},
"SecurityGroupIds": [
  "sg-1234t1"
],
"InternetAccessible": {
  "PublicIpAssigned": true,
  "InternetChargeType": "POSTPAID_BY_HOUR",
  "BandwidthPackageId": "dcaew",
  "InternetMaxBandwidthOut": 0
},
"VirtualPrivateClouds": [
  {
    "SubnetId": "subnet-23syenxk",
    "AsVpcGateway": true,
    "Ipv6AddressCount": 1,
    "VpcId": "vpc-1xnwrwxw2",
    "PrivateIpAddresses": [
      "10.10.2.29"
    ]
  }
],
"ImageId": "img-3sdke3x2",
"InstanceName": "test-env",
"InstanceType": "S5.SMALL2",
"InstanceTypeOptions": {
  "Memory": 1,
```

```
"CPU": 1,
"InstanceCategories": [
  "ALL"
],
},
"DataDisks": [
{
  "DeleteWithInstance": true,
  "Encrypt": true,
  "CdcId": "2",
  "DiskType": "CLOUD_SSD",
  "ThroughputPerformance": 0,
  "KmsKeyId": "kms-213k73kx",
  "DiskSize": 0,
  "SnapshotId": "snap-fsdekxe2",
  "DiskId": "disk-ecqwbck"
}
],
},
"EnvDescription": "test-env",
"MountDataDisks": [
{
  "FileSystemType": "EXT4",
  "LocalPath": "/mnt"
}
],
"DesiredComputeNodeCount": 1,
"EnvName": "test-env",
"InputMappings": [
{
  "SourcePath": "cos://test-1111123424.cos.ap-guangzhou.myqcloud.com/batchtest/input/",
  "DestinationPath": "/opt/data",
  "MountOptionParameter": ""
}
],
"EnvType": "MANAGED"
}
}
}
}
```

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
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
InvalidParameter.InvalidParameterCombination	非法的参数组合。
InvalidParameterValue	参数取值错误。
InvalidZone.MismatchRegion	指定的zone不存在。
UnknownParameter	未知参数错误。

获取计算环境列表

最近更新时间：2025-05-28 01:08:18

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查看计算环境列表

默认接口请求频率限制：10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：DescribeComputeEnvs。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvIds.N	否	Array of String	计算环境ID列表，与Filters参数不能同时指定。最大数量上限100。 示例值：["env-lcx7qbbr"]
Filters.N	否	Array of Filter	过滤条件 - zone - String - 是否必填：否 - （过滤条件）按照可用区过滤，可用区通过调用接口 DescribeZones 获取。 - env-id - String - 是否必填：否 - （过滤条件）按照计算环境ID过滤。 - env-name - String - 是否必填：否 - （过滤条件）按照计算环境名称过滤。 - resource-type - String - 是否必填：否 - （过滤条件）按照计算资源类型过滤，取值CVM或者CPM(黑石)。 - tag-key - String - 是否必填：否 - （过滤条件）按照标签键进行过滤。 - tag-value - String - 是否必填：否 - （过滤条件）按照标签值进行过滤。 - tag:tag-key - String - 是否必填：否 - （过滤条件）按照标签键值对进行过滤。tag-key使用具体的标签键进行替换。 与EnvIds参数不能同时指定。
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	返回数量，默认值20，最大值100。 示例值：20

3. 输出参数

参数名称	类型	描述
ComputeEnvSet	Array of ComputeEnvView	计算环境列表
TotalCount	Integer	计算环境数量 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查看计算环境列表

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeComputeEnvs
&EnvIds.0=env-lcx7qbbr
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "ComputeEnvSet": [
      {
        "EnvId": "env-lcx7qbbr",
        "ComputeNodeMetrics": {
          "CreatedCount": 0,
          "DeletingCount": 0,
          "CreationFailedCount": 0,
          "SubmittedCount": 0,
          "CreatingCount": 0,
          "AbnormalCount": 1,
          "RunningCount": 0
        },
        "Tags": [
          {
            "Key": "batch-test-tag-env-key",
            "Value": "batch-test-tag-env-value"
          }
        ],
        "ResourceType": "CVM",
        "EnvType": "MANAGED",
        "AttachedComputeNodeCount": 0,
        "DesiredComputeNodeCount": 1,
        "EnvName": "test-compute-env",
        "Placement": {
          "Zone": "ap-guangzhou-4",
          "ProjectId": 0
        },
        "NextAction": "",
        "CreateTime": "2020-09-22T06:51:42Z"
      }
    ],
    "RequestId": "bedb1750-a09f-4c4d-9f25-6afccda50dea"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidFilter	指定的Filter不被支持。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
InvalidParameter.InvalidParameterCombination	非法的参数组合。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidFilter	Filter参数不正确。
InvalidParameterValue.ResourceType	指定的ResourceType不合法。
InvalidZone.MismatchRegion	指定的zone不存在。
UnknownParameter	未知参数错误。

从计算环境移出实例

最近更新時間：2025-05-28 01:08:18

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

将添加到计算环境中的实例从计算环境中移出。若是由批量计算自动创建的计算节点实例则不允许移出。

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：DetachInstances。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：env-1mekuyxs
InstanceIds.N	是	Array of String	实例ID列表，实例ID通过调用接口 DescribeInstances 获取。 示例值：["ins-kjoossv5"]

3. 输出参数

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

4. 示例

示例1 从计算环境移出实例

输入示例

```
https://batch.tencentcloudapi.com/?Action=DetachInstances
&EnvId=env-1mekuyxs
&InstanceIds.0=ins-kjoossv5
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "bd757b4d-9797-493e-b421-89873457a071"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallCvm	调用CVM API返回错误。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
InvalidParameterValue.InstanceIdDuplicated	实例ID重复。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

修改计算环境

最近更新时间：2025-05-28 01:08:18

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于修改计算环境属性

默认接口请求频率限制：2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：ModifyComputeEnv。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：env-lcpcej85
DesiredComputeNodeCount	否	Integer	计算节点期望个数，最大上限2000。 示例值：2
EnvName	否	String	计算环境名称 示例值：test-env
EnvDescription	否	String	计算环境描述 示例值：test-env
EnvData	否	ComputeEnvData	计算环境属性数据

3. 输出参数

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

4. 示例

示例1 调整计算环境节点期望数量为2

输入示例

```
https://batch.tencentcloudapi.com/?Action=ModifyComputeEnv
&EnvId=env-lcpcej85
&DesiredComputeNodeCount=2
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "147fbf83-f817-4528-a547-f5f313b673d2"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallCpmAPI	调用Cpm API返回错误。
InvalidParameter.EnvDescriptionTooLong	计算环境描述过长。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
InvalidParameter.EnvNameTooLong	计算环境名称过长。
InvalidParameterValue.InstanceType	不支持指定的实例类型。
InvalidParameterValue.InstanceTypeDuplicate	实例类型不能有重复值。
InvalidParameterValue.InstanceTypesEmpty	实例类型列表不能为空。
InvalidParameterValue.LimitExceeded	Filter参数值数量超过限制。
InvalidParameterValue.Negative	非法的负值参数。
LimitExceeded.CpuQuota	CPU配额不足。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
UnsupportedOperation.AcceptOtherRequest	处理其他请求，禁止删除操作。
UnsupportedOperation.ComputeEnvAcceptOtherRequest	计算环境正在接受请求。

错误码	描述
UnsupportedOperation.ComputeNodeForbidTerminate	禁止终止计算节点。
UnsupportedOperation.NotEnoughComputeNodesToTerminate	缩容时，缩容数大于现有计算节点数。

销毁计算节点

最近更新时间：2025-05-28 01:08:18

-

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于销毁计算节点。

对于状态为CREATED、CREATION_FAILED、RUNNING和ABNORMAL的节点，允许销毁处理。

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：TerminateComputeNode。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnvs 获取。 示例值：env-lcx7qbbr
ComputeNodeId	是	String	计算节点ID，节点ID通过调用接口 DescribeComputeEnv 获取。 示例值：node-lcpcej85

3. 输出参数

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

4. 示例

示例1 销毁计算节点

输入示例

```
https://batch.tencentcloudapi.com/?Action=TerminateComputeNode
&EnvId=env-lcx7qbbr
&ComputeNodeId=node-lcpcej85
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "c50bfbf1-d214-4e41-90f6-39270dc9f071"
  }
}
```

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
InvalidParameter.ComputeNodeIdMalformed	非法的计算节点ID格式。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
ResourceNotFound.ComputeNode	指定计算节点不存在。
UnsupportedOperation.AcceptOtherRequest	处理其他请求，禁止删除操作。
UnsupportedOperation.ComputeEnvAcceptOtherRequest	计算环境正在接受请求。
UnsupportedOperation.ComputeEnvOperation	禁止删除操作。
UnsupportedOperation.ComputeNodeForbidTerminate	禁止终止计算节点。
UnsupportedOperation.ComputeNodeIsTerminating	计算节点停止中。

批量销毁计算节点

最近更新时间：2025-05-15 01:09:59

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于批量销毁计算节点，不允许重复销毁同一个节点。

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：TerminateComputeNodes。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
EnvId	是	String	计算环境ID，环境ID通过调用接口 DescribeComputeEnv 获取。 示例值：env-lcx7qbbr
ComputeNodeIds.N	是	Array of String	计算节点ID列表，最大数量上限100，节点ID通过调用接口 DescribeComputeEnv 获取。 示例值：["node-k9fwel23"]

3. 输出参数

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

4. 示例

示例1 批量销毁计算节点

输入示例

```
https://batch.tencentcloudapi.com/?Action=TerminateComputeNodes
&EnvId=env-lcx7qbbr
&ComputeNodeIds.0=node-lcpcej85
&ComputeNodeIds.1=node-lcz8jkvw
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "c50bfbf1-d214-4e41-90f6-39270dc9f071"
  }
}
```

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
InvalidParameter.ComputeNodeIdMalformed	非法的计算节点ID格式。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
ResourceNotFound.ComputeNode	指定计算节点不存在。
UnsupportedOperation	操作不支持。
UnsupportedOperation.AcceptOtherRequest	处理其他请求，禁止删除操作。
UnsupportedOperation.ComputeEnvAcceptOtherRequest	计算环境正在接受请求。
UnsupportedOperation.ComputeNodeForbidTerminate	禁止终止计算节点。
UnsupportedOperation.ComputeNodeIsTerminating	计算节点停止中。

任务模板相关接口

创建任务模板

最近更新时间：2025-05-15 01:10:05

1. 接口描述

接口请求域名：`batch.tencentcloudapi.com`。

用于创建任务模板

默认接口请求频率限制：2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：CreateTaskTemplate。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TaskTemplateName	是	String	任务模板名称，最大长度限制60个字符。 示例值：job1
TaskTemplateInfo	是	Task	任务模板内容，参数要求与任务一致
TaskTemplateDescription	否	String	任务模板描述，最大长度限制200个字符。 示例值：test template
Tags.N	否	Array of Tag	标签列表。通过指定该参数可以支持绑定标签到任务模板。每个任务模板最多绑定10个标签。

3. 输出参数

参数名称	类型	描述
TaskTemplateId	String	任务模板ID 示例值：task-tmpl-7vtx96n2
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 创建任务模板

输入示例

```
https://batch.tencentcloudapi.com/?Action=CreateTaskTemplate
&TaskTemplateName=A
&TaskTemplateDescription=demo_task_template
&TaskTemplateInfo.TaskName=A
&TaskTemplateInfo.TaskInstanceNum=1
&TaskTemplateInfo.Application.DeliveryForm=LOCAL
&TaskTemplateInfo.Application.Command=python-c"fib=lamdban:1ifn<=2elsefib(n-1)+fib(n-2);print(fib(20))"
```

```
&TaskTemplateInfo.ComputeEnv.EnvType=MANAGED
&TaskTemplateInfo.ComputeEnv.EnvData.InstanceType=S1.SMALL1
&TaskTemplateInfo.ComputeEnv.EnvData.ImageId=img-bd78fy2t
&TaskTemplateInfo.RedirectInfo.StdoutRedirectPath=cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/
&TaskTemplateInfo.RedirectInfo.StderrRedirectPath=cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TaskTemplateId": "task-tmpl-7vtx96n2",
    "RequestId": "83cd875b-7ac0-4e5b-b150-0e69c59e5e52"
  }
}
```

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
AllowedOneAttributeInEnvIdAndComputeEnv	ComputeEnv 和 EnvId 必须指定一个（且只有一个）参数。
InternalServerError	内部错误。
InternalServerError.CallTagAPI	获取Tag错误。
InvalidParameter.CvmParameters	非法的CVM参数。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
InvalidParameter.ImageIdMalformed	镜像ID不正确。
InvalidParameter.TaskName	非法的任务名称。
InvalidParameter.TaskNameTooLong	任务名称过长。

错误码	描述
InvalidParameter.TaskTemplateName	非法的任务模板名称。
InvalidParameter.TaskTemplateNameTooLong	任务模板名称过长。
InvalidParameterValue.ComputeEnv	计算环境参数校验失败。
InvalidParameterValue.DependenceNotFoundTaskName	找不到依赖任务定义。
InvalidParameterValue.DependenceUnfeasible	禁止环状任务依赖关系。
InvalidParameterValue.InvalidDataTypeAny	DataTypeAny不合法。
InvalidParameterValue.LocalPath	非法的本地存储路径。
InvalidParameterValue.MaxRetryCount	重试次数过大。
InvalidParameterValue.Negative	非法的负值参数。
InvalidParameterValue.RemoteStoragePath	非法的存储路径格式。
InvalidParameterValue.RemoteStorageSchemeType	非法的存储类型。
LimitExceeded.TaskTemplateQuota	任务模板配额不足。

删除任务模板

最近更新时间：2025-05-15 01:10:05

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于删除任务模板信息

默认接口请求频率限制：2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：DeleteTaskTemplates。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TaskTemplateIds.N	是	Array of String	用于删除任务模板信息，最大数量上限100，环境模版ID通过调用接口 DescribeTaskTemplates 获取。 示例值：["task-tmpl-7vtx96n2"]

3. 输出参数

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

4. 示例

示例1 删除任务模板

输入示例

```
https://batch.tencentcloudapi.com/?Action=DeleteTaskTemplates
&TaskTemplateIds.0=task-tmpl-7vtx96n2
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "1b68e632-c0eb-4e85-b9ef-a152aa3cc073"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.TaskTemplateIdMalformed	非法的任务模板ID格式。
ResourceNotFound.TaskTemplate	指定任务模板ID不存在。

获取任务模板

最近更新时间：2025-04-25 01:09:25

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查询任务模板信息

默认接口请求频率限制：10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：DescribeTaskTemplates。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TaskTemplateIds.N	否	Array of String	任务模板ID列表，与Filters参数不能同时指定。 示例值：["task-tmpl-pq24rq1e\n","task-tmpl-606i415o\n"]
Filters.N	否	Array of Filter	过滤条件 - task-template-name - String - 是否必填：否 -（过滤条件）按照任务模板名称过滤。 - tag-key - String - 是否必填：否 -（过滤条件）按照标签键进行过滤。 - tag-value - String - 是否必填：否 -（过滤条件）按照标签值进行过滤。 - tag:tag-key - String - 是否必填：否 -（过滤条件）按照标签键值对进行过滤。tag-key使用具体的标签键进行替换。 与TaskTemplateIds参数不能同时指定。
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	返回数量 示例值：10

3. 输出参数

参数名称	类型	描述
TaskTemplateSet	Array of TaskTemplateView	任务模板列表
TotalCount	Integer	任务模板数量 示例值：2
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 查询任务模板

输入示例


```
https://batch.tencentcloudapi.com/?Action=DescribeTaskTemplates
&TaskTemplateIds.0=task-tmpl-pq24rq1e
&TaskTemplateIds.1=task-tmpl-606i415o
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "TaskTemplateSet": [
      {
        "TaskTemplateName": "rendering",
        "TaskTemplateDescription": "3ds Max 2018 Demo",
        "TaskTemplateId": "task-tmpl-pq24rq1e",
        "CreateTime": "2018-03-08T03:27:43Z",
        "Tags": [
          {
            "Key": "batch-test-tag-tp-key",
            "Value": "batch-test-tag-tp-value"
          }
        ],
        "TaskTemplateInfo": {
          "OutputMappings": [
            {
              "SourcePath": "C:\\\\render\\\\",
              "DestinationPath": "cos://bucket-appid.cos.ap-guangzhou.myqcloud.com/"
            }
          ],
          "TaskInstanceNum": 1,
          "ComputeEnv": {
            "EnvType": "MANAGED",
            "EnvData": {
              "SystemDisk": {
                "DiskSize": "50",
                "DiskType": "CLOUD_BASIC"
              },
              "LoginSettings": {
                "Password": "B1habcdB1habcd"
              },
              "InternetAccessible": {
                "InternetMaxBandwidthOut": "10",
                "InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR"
              },
              "ImageId": "img-i64lx84h",
              "InstanceType": "S1.LARGE8",
              "DataDisks": [
                {
                  "DiskSize": "0",
                  "DiskType": "CLOUD_BASIC"
                }
              ]
            }
          },
          "RedirectInfo": {
            "StdoutRedirectPath": "cos://bucket-appid.cos.ap-guangzhou.myqcloud.com/wonderflet/logs/",
```

```
"StderrRedirectPath": "cos://bucket-appid.cos.ap-guangzhou.myqcloud.com/wonderflet/logs/"
},
"Application": {
  "Command": "3dsmaxcmd Demo.max -outputName:c:\\\\render\\\\\\image.jpg",
  "PackagePath": "cos://bucket-appid.cos.ap-guangzhou.myqcloud.com/render/max.tar.gz",
  "DeliveryForm": "PACKAGE"
},
"MaxRetryCount": 0,
"Timeout": 259200,
"FailedAction": "TERMINATE",
"TaskName": "rendering"
}
},
{
  "TaskTemplateName": "A",
  "TaskTemplateDescription": "",
  "TaskTemplateId": "task-tmpl-606i415o",
  "CreateTime": "2018-01-25T07:46:12Z",
  "Tags": [
    {
      "Key": "batch-test-tag-tp-key",
      "Value": "batch-test-tag-tp-value"
    }
  ],
  "TaskTemplateInfo": {
    "TaskInstanceNum": 2,
    "ComputeEnv": {
      "EnvType": "MANAGED",
      "EnvData": {
        "InstanceType": "S1.SMALL1",
        "ImageId": "img-bq1gnde2"
      }
    },
    "RedirectInfo": {
      "StdoutRedirectPath": "cos://bucket-appid.cos.ap-guangzhou.myqcloud.com/hello2/logs/",
      "StderrRedirectPath": "cos://bucket-appid.cos.ap-guangzhou.myqcloud.com/hello2/logs/"
    },
    "Application": {
      "Command": "python -c \"fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2); print(fib(20))\" ",
      "DeliveryForm": "LOCAL"
    },
    "MaxRetryCount": 0,
    "FailedAction": "TERMINATE",
    "TaskName": "A"
  }
}
],
"RequestId": "23f7d202-1cd4-4658-a989-88ad4ed77aac"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.InvalidParameterCombination	非法的参数组合。
InvalidParameter.TaskTemplateIdMalformed	非法的任务模板ID格式。
InvalidParameterValue	参数取值错误。
UnknownParameter	未知参数错误。

修改任务模板

最近更新时间：2025-05-15 01:10:04

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于修改任务模板

默认接口请求频率限制：2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：ModifyTaskTemplate。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TaskTemplateId	是	String	任务模板ID；详见 任务模版 。 示例值：task-tmpl-606i415o
TaskTemplateName	否	String	任务模板名称；字节长度限制60。 示例值：A
TaskTemplateDescription	否	String	任务模板描述；字节长度限制200。 示例值：first template
TaskTemplateInfo	否	Task	任务模板信息

3. 输出参数

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

4. 示例

示例1 更改任务模板

输入示例

```
https://batch.tencentcloudapi.com/?Action=ModifyTaskTemplate
&TaskTemplateId=task-tmpl-606i415o
&TaskTemplateName=A
&TaskTemplateDescription=demo_task
&TaskTemplateInfo.TaskName=A
&TaskTemplateInfo.TaskInstanceNum=1
&TaskTemplateInfo.Application.DeliveryForm=LOCAL
&TaskTemplateInfo.Application.Command=python -c "fib=lambda n:1 if n<=2 else fib(n-1)+fib(n-2); print(fib(20))"
&TaskTemplateInfo.ComputeEnv.EnvType=MANAGED
&TaskTemplateInfo.ComputeEnv.EnvData.InstanceType=S1.SMALL1
```

```
&TaskTemplateInfo.ComputeEnv.EnvData.ImageId=img-bd78fy2t
&TaskTemplateInfo.RedirectInfo.StdoutRedirectPath=cos://bucket-appid.cos.ap-guangzhou.myqcloud.com/hello2/logs/
&TaskTemplateInfo.RedirectInfo.StderrRedirectPath=cos://bucket-appid.ap-guangzhou.myqcloud.com/hello2/logs/
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "8a56bd93-688e-41c3-bc8b-f9867ffb0dd0"
  }
}
```

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
AllowedOneAttributeInEnvIdAndComputeEnv	ComputeEnv 和 EnvId 必须指定一个（且只有一个）参数。
InternalServerError	内部错误。
InvalidParameter.CvmParameters	非法的CVM参数。
InvalidParameter.TaskName	非法的任务名称。
InvalidParameter.TaskNameTooLong	任务名称过长。
InvalidParameter.TaskTemplateDescriptionTooLong	任务模板描述过长。
InvalidParameter.TaskTemplateIdMalformed	非法的任务模板ID格式。
InvalidParameter.TaskTemplateName	非法的任务模板名称。
InvalidParameter.TaskTemplateNameTooLong	任务模板名称过长。

错误码	描述
InvalidParameterAtLeastOneAttribute	TaskTemplateName、TaskTemplateDescription和TaskTemplateInfo至少包含一项。
InvalidParameterValue.ComputeEnv	计算环境参数校验失败。
InvalidParameterValue.DependenceNotFoundTaskName	找不到依赖任务定义。
InvalidParameterValue.DependenceUnfeasible	禁止环状任务依赖关系。
InvalidParameterValue.LocalPath	非法的本地存储路径。
InvalidParameterValue.MaxRetryCount	重试次数过大。
InvalidParameterValue.Negative	非法的负值参数。
InvalidParameterValue.RemoteStoragePath	非法的存储路径格式。
InvalidParameterValue.RemoteStorageSchemeType	非法的存储类型。
ResourceNotFound.TaskTemplate	指定任务模板ID不存在。

作业相关接口

删除作业

最近更新时间：2025-05-15 01:10:04

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于删除作业记录。

删除作业的效果相当于删除作业相关的所有信息。删除成功后，作业相关的所有信息都无法查询。

待删除的作业必须处于完结状态，且其内部包含的所有任务实例也必须处于完结状态，否则会禁止操作。完结状态，是指处于 SUCCEED 或 FAILED 状态。

默认接口请求频率限制：30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：DeleteJob。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID；JobId详见 作业列表 示例值：job-f5da1ed1

3. 输出参数

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

4. 示例

示例1 删除作业

输入示例

```
https://batch.tencentcloudapi.com/?Action=DeleteJob
&JobId=job-f5da1ed1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0ffc9347-a47a-43a6-a64a-aa77fda7ea13"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallTagAPI	获取Tag错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
ResourceInUse.Job	Job在使用中。
ResourceNotFound.Job	指定作业不存在。
UnsupportedOperation	操作不支持。

查看作业详情

最近更新时间：2025-05-21 01:08:41

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查看一个作业的详细信息，包括内部任务（Task）和依赖（Dependence）信息。

默认接口请求频率限制：50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：DescribeJob。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID；JobId详见 作业列表 示例值：job-97zcl3wt

3. 输出参数

参数名称	类型	描述
JobId	String	作业ID 示例值：job-97zcl3wt
JobName	String	作业名称 示例值：test job
Zone	String	可用区信息 示例值：ap-guangzhou-2
Priority	Integer	作业优先级 示例值：1
JobState	String	作业状态 示例值：SUCCEED
CreateTime	String	创建时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 示例值：2018-02-07T09:29:09Z
EndTime	String	结束时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 示例值：2018-02-07T09:33:01Z
TaskSet	Array of TaskView	任务视图信息
DependenceSet	Array of Dependence	任务间依赖信息
TaskMetrics	TaskMetrics	任务统计指标
TaskInstanceMetrics	TaskInstanceMetrics	任务实例统计指标
StateReason	String	作业失败原因 示例值：RUNNING

参数名称	类型	描述
Tags	Array of Tag	作业绑定的标签列表。 注意：此字段可能返回 null，表示取不到有效值。
NextAction	String	下一步动作 示例值：TERMINATING
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 查询作业详细信息

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeJob
&JobId=job-97zcl3wt
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "JobState": "SUCCEED",
    "TaskSet": [
      {
        "EndTime": "2018-02-07T09:30:31Z",
        "TaskName": "A",
        "TaskState": "SUCCEED",
        "CreateTime": "2018-02-07T09:29:09Z"
      },
      {
        "EndTime": "2018-02-07T09:31:49Z",
        "TaskName": "B",
        "TaskState": "SUCCEED",
        "CreateTime": "2018-02-07T09:29:09Z"
      },
      {
        "EndTime": "2018-02-07T09:31:48Z",
        "TaskName": "C",
        "TaskState": "SUCCEED",
        "CreateTime": "2018-02-07T09:29:09Z"
      },
      {
        "EndTime": "2018-02-07T09:33:01Z",
        "TaskName": "D",
        "TaskState": "SUCCEED",
        "CreateTime": "2018-02-07T09:29:09Z"
      }
    ],
    "Zone": "ap-guangzhou-2",
    "TaskMetrics": {
      "PendingCount": 0,
      "FailedCount": 0,
      "StartingCount": 0,
      "SucceedCount": 4,
    }
  }
}
```

```
"FailedInterruptedCount": 0,
"SubmittedCount": 0,
"RunnableCount": 0,
"RunningCount": 0
},
"JobName": "test job",
"JobId": "job-97zcl3wt",
"Priority": 1,
"StateReason": "",
"TaskInstanceMetrics": {
  "PendingCount": 0,
  "FailedCount": 0,
  "StartingCount": 0,
  "SucceedCount": 4,
  "FailedInterruptedCount": 0,
  "SubmittedCount": 0,
  "RunnableCount": 0,
  "RunningCount": 0
},
"EndTime": "2018-02-07T09:33:01Z",
"DependenceSet": [
  {
    "StartTask": "A",
    "EndTask": "B"
  },
  {
    "StartTask": "A",
    "EndTask": "C"
  },
  {
    "StartTask": "B",
    "EndTask": "D"
  },
  {
    "StartTask": "C",
    "EndTask": "D"
  }
],
"CreateTime": "2018-02-07T09:29:09Z",
"Tags": [
  {
    "Key": "batch-test-tag-job-key",
    "Value": "batch-test-tag-job-value"
  }
],
"NextAction": "",
"RequestId": "d1b08863-b8ee-49d4-aa08-f464499f97a0"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
ResourceNotFound.Job	指定作业不存在。

获取作业的提交信息

最近更新时间：2025-04-25 01:09:25

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查询指定作业的提交信息，其返回内容包括 JobId 和 SubmitJob 接口中作为输入参数的作业提交信息

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：DescribeJobSubmitInfo。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID 示例值：job-97zcl3wt

3. 输出参数

参数名称	类型	描述
JobId	String	作业ID 示例值：job-97zcl3wt
JobName	String	作业名称 示例值：test job
JobDescription	String	作业描述 示例值：xxx
Priority	Integer	作业优先级，任务（Task）和任务实例（TaskInstance）会继承作业优先级 示例值：1
Tasks	Array of Task	任务信息
Dependences	Array of Dependence	依赖信息
Tags	Array of Tag	作业绑定的标签列表。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取作业的提交信息

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeJobSubmitInfo
&&JobId=job-97zcl3wt
&&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Tasks": [
      {
        "TaskInstanceNum": 2,
        "ComputeEnv": {
          "EnvType": "MANAGED",
          "EnvData": {
            "InstanceType": "S1.SMALL1",
            "ImageId": "img-bd78fy2t"
          }
        },
        "RedirectInfo": {
          "StdoutRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/",
          "StderrRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/"
        },
        "Application": {
          "Command": "python -c \"fib=lambda n:1 if n<2 else fib(n-1)+fib(n-2); print(fib(20))\" ",
          "DeliveryForm": "LOCAL"
        },
        "MaxRetryCount": 0,
        "FailedAction": "TERMINATE",
        "TaskName": "A"
      },
      {
        "TaskInstanceNum": 2,
        "ComputeEnv": {
          "EnvType": "MANAGED",
          "EnvData": {
            "InstanceType": "S1.SMALL1",
            "ImageId": "img-bd78fy2t"
          }
        },
        "RedirectInfo": {
          "StdoutRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/",
          "StderrRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/"
        },
        "Application": {
          "Command": "python -c \"fib=lambda n:1 if n<2 else fib(n-1)+fib(n-2); print(fib(20))\" ",
          "DeliveryForm": "LOCAL"
        },
        "MaxRetryCount": 0,
        "FailedAction": "TERMINATE",
        "TaskName": "B"
      },
      {
        "TaskInstanceNum": 2,
        "ComputeEnv": {
          "EnvType": "MANAGED",
          "EnvData": {
```

```
"InstanceType": "S1.SMALL1",
"ImageId": "img-bd78fy2t"
},
},
"RedirectInfo": {
"StdoutRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/",
"StderrRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/"
},
"Application": {
"Command": "python -c \"fib=lambda n:1 if n<2 else fib(n-1)+fib(n-2); print(fib(20))\"",
"DeliveryForm": "LOCAL"
},
"MaxRetryCount": 0,
"FailedAction": "TERMINATE",
"TaskName": "C"
},
{
"TaskInstanceNum": 2,
"ComputeEnv": {
"EnvType": "MANAGED",
"EnvData": {
"InstanceType": "S1.SMALL1",
"ImageId": "img-bd78fy2t"
}
},
"RedirectInfo": {
"StdoutRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/",
"StderrRedirectPath": "cos://bucket-appid.cosgz.myqcloud.com/hello2/logs/"
},
"Application": {
"Command": "python -c \"fib=lambda n:1 if n<2 else fib(n-1)+fib(n-2); print(fib(20))\"",
"DeliveryForm": "LOCAL"
},
"MaxRetryCount": 0,
"FailedAction": "TERMINATE",
"TaskName": "D"
}
},
"Dependences": [
{
"StartTask": "A",
"EndTask": "B"
},
{
"StartTask": "A",
"EndTask": "C"
},
{
"StartTask": "B",
"EndTask": "D"
},
{
"StartTask": "C",
"EndTask": "D"
}
],
"JobName": "test_job",
```

```
"Priority": 1,
"JobDescription": "xxx",
"JobId": "job-97zcl3wt",
"Tags": [
{
"Key": "batch-test-tag-job-key",
"Value": "batch-test-tag-job-value"
}
],
"RequestId": "1a69db97-95ff-4bf7-9eb2-875ea6e5737b"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
ResourceNotFound.Job	指定作业不存在。

查看作业列表

最近更新时间：2025-05-15 01:10:04

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查询若干个作业的概览信息

默认接口请求频率限制：10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：DescribeJobs。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobIds.N	否	Array of String	作业ID列表，与Filters参数不能同时指定。 示例值：["job-7e7evk2x","job-3n2weaqt"]
Filters.N	否	Array of Filter	过滤条件 - job-id - String - 是否必填：否 - （过滤条件）按照作业ID过滤。 - job-name - String - 是否必填：否 - （过滤条件）按照作业名称过滤。 - job-state - String - 是否必填：否 - （过滤条件）按照作业状态过滤。 - SUBMITTED：已提交； - PENDING：等待中； - RUNNABLE：可运行； - STARTING：启动中； - RUNNING：运行中； - SUCCEED：成功； - FAILED：失败； - FAILED_INTERRUPTED：失败后保留实例。 - zone - String - 是否必填：否 - （过滤条件）按照可用区过滤。 - tag-key - String - 是否必填：否 - （过滤条件）按照标签键进行过滤。 - tag-value - String - 是否必填：否 - （过滤条件）按照标签值进行过滤。 - tag:tag-key - String - 是否必填：否 - （过滤条件）按照标签键值对进行过滤。tag-key使用具体的标签键进行替换。 与JobIds参数不能同时指定。
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	返回job数量限制，最大值：100，默认值：20。 示例值：100

3. 输出参数

参数名称	类型	描述
JobSet	Array of JobView	作业列表

参数名称	类型	描述
TotalCount	Integer	符合条件的作业数量 示例值：2
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 查询作业信息

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeJobs
&JobIds.0=job-7e7evk2x
&JobIds.1=job-3n2weaqt
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "JobSet": [
      {
        "EndTime": "2020-09-21T08:59:45Z",
        "JobState": "STARTING",
        "Tags": [
          {
            "Key": "batch-test-tag-job-key",
            "Value": "batch-test-tag-job-value"
          }
        ],
        "TaskMetrics": {
          "PendingCount": 0,
          "FailedCount": 0,
          "StartingCount": 1,
          "SucceedCount": 0,
          "FailedInterruptedCount": 0,
          "SubmittedCount": 0,
          "RunnableCount": 0,
          "RunningCount": 0
        },
        "JobId": "job-3n2weaqt",
        "Priority": 1,
        "Placement": {
          "Zone": "ap-guangzhou-4",
          "ProjectId": 0,
          "HostIds": [
            "host-o5i742go"
          ],
          "HostId": "host-o5i742go"
        },
        "JobName": "hello-test-job",
        "CreateTime": "2020-09-22T07:14:54Z"
      },
    ],
  },
}
```

```
{
  "EndTime": "2020-09-21T08:59:45Z",
  "JobState": "FAILED",
  "Tags": [
    {
      "Key": "job-key1",
      "Value": "job-value1"
    },
    {
      "Key": "job-key",
      "Value": "job-value"
    }
  ],
  "TaskMetrics": {
    "PendingCount": 0,
    "FailedCount": 1,
    "StartingCount": 0,
    "SucceedCount": 0,
    "FailedInterruptedCount": 0,
    "SubmittedCount": 0,
    "RunnableCount": 0,
    "RunningCount": 0
  },
  "JobId": "job-7e7evk2x",
  "Priority": 1,
  "Placement": {
    "Zone": "ap-guangzhou-4",
    "ProjectId": 1,
    "HostIds": [
      "host-o5i737go"
    ],
    "HostId": "host-o5i737go"
  },
  "JobName": "hello-test-job",
  "CreateTime": "2020-09-21T08:57:01Z"
}
],
"RequestId": "ba5df388-51bd-434d-81aa-d640f284b4e7"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidFilter	指定的Filter不被支持。
InvalidParameter.InvalidParameterCombination	非法的参数组合。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidFilter	Filter参数不正确。
InvalidParameterValue.LimitExceeded	Filter参数值数量超过限制。
InvalidZone.MismatchRegion	指定的zone不存在。
UnknownParameter	未知参数错误。

查询任务详情

最近更新时间：2025-05-15 01:10:04

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于查询指定任务的详细信息，包括任务内部的任务实例信息。

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：DescribeTask。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID；JobId详见 作业列表 示例值：job-97zcl3wt
TaskName	是	String	任务名称 示例值：A
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	返回数量。默认取值100，最大取值1000。 示例值：100
Filters.N	否	Array of Filter	过滤条件，详情如下： task-instance-state - String - 是否必填：否 - 按照任务实例状态进行过滤（ - SUBMITTED：已提交； - PENDING：等待中； - RUNNABLE：可运行； - STARTING：启动中； - RUNNING：运行中； - SUCCEED：成功； - FAILED：失败； - FAILED_INTERRUPTED：失败后保留实例）。

3. 输出参数

参数名称	类型	描述
JobId	String	作业ID 示例值：job-97zcl3wt
TaskName	String	任务名称 示例值：A
TaskState	String	任务状态 示例值：SUCCEED
CreateTime	String	创建时间

参数名称	类型	描述
		示例值：2018-02-07T09:29:09Z
EndTime	String	结束时间 示例值：2018-02-07T09:30:31Z
TaskInstanceTotalCount	Integer	任务实例总数 示例值：2
TaskInstanceSet	Array of TaskInstanceView	任务实例信息
TaskInstanceMetrics	TaskInstanceMetrics	任务实例统计指标
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 查询任务详情

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeTask
&JobId=job-97zcl3wt
&TaskName=A
&Filters.0.Name=task-instance-state
&Filters.0.Values.0=SUCCEED
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TaskInstanceSet": [
      {
        "TaskInstanceIndex": 1,
        "RedirectInfo": {
          "StderrRedirectFileName": "stderr.job-97zcl3wt.A.1.log",
          "StdoutRedirectFileName": "stdout.job-97zcl3wt.A.1.log",
          "StdoutRedirectPath": "cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/",
          "StderrRedirectPath": "cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/"
        },
        "RunningTime": "2018-02-07T09:29:55Z",
        "LaunchTime": "2018-02-07T09:29:09Z",
        "StateReason": "",
        "StateDetailedReason": "successful",
        "ComputeNodeInstanceId": "ins-9s04904p",
        "EndTime": "2018-02-07T09:30:23Z",
        "TaskInstanceState": "SUCCEED",
        "CreateTime": "2018-02-07T09:29:09Z",
        "ExitCode": 0
      },
      {
        "TaskInstanceIndex": 0,
        "RedirectInfo": {
          "StderrRedirectFileName": "stderr.job-97zcl3wt.A.0.log",
          "StdoutRedirectFileName": "stdout.job-97zcl3wt.A.0.log",
          "StdoutRedirectPath": "cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/",
          "StderrRedirectPath": "cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/"
        }
      }
    ]
  }
}
```

```
"StderrRedirectPath": "cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/"
},
"RunningTime": "2018-02-07T09:30:02Z",
"LaunchTime": "2018-02-07T09:29:09Z",
"StateReason": "",
"StateDetailedReason": "successful",
"ComputeNodeInstanceId": "ins-b4tgwxvn",
"EndTime": "2018-02-07T09:30:31Z",
"TaskInstanceState": "SUCCEED",
"CreateTime": "2018-02-07T09:29:09Z",
"ExitCode": 0
}
],
"TaskState": "SUCCEED",
"JobId": "job-97zcl3wt",
"TaskInstanceMetrics": {
  "PendingCount": 0,
  "FailedCount": 0,
  "StartingCount": 0,
  "SucceedCount": 2,
  "FailedInterruptedCount": 0,
  "SubmittedCount": 0,
  "RunnableCount": 0,
  "RunningCount": 0
},
"TaskInstanceTotalCount": 2,
"TaskName": "A",
"EndTime": "2018-02-07T09:30:31Z",
"CreateTime": "2018-02-07T09:29:09Z",
"RequestId": "f4723b24-e080-4599-b12d-b7eb657faefe"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalError	内部错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
ResourceNotFound.Task	指定作业任务不存在。

获取任务日志详情

最近更新时间：2025-05-16 01:10:15

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于获取任务多个实例标准输出和标准错误日志。

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：DescribeTaskLogs。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID。JobId详见 作业列表 。 示例值：job-mawzp3ja
TaskName	是	String	任务名称 示例值：task
TaskInstanceIndexes.N	否	Array of Integer	任务实例集合；与Offset不能同时指定。 示例值：[0, 1]
Offset	否	Integer	起始任务实例。与TaskInstanceIndexes参数不能同时指定。 示例值：0
Limit	否	Integer	最大任务实例数；默认值为5，最大值为10。 示例值：10

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	任务实例总数 示例值：1
TaskInstanceLogSet	Array of TaskInstanceLog	任务实例日志详情集合
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 获取任务日志详情

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeTaskLogs
&JobId=job-mawzp3ja
```

```
&TaskName=task
&TaskInstanceIndexes.0=0
&TaskInstanceIndexes.1=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "TaskInstanceLogSet": [
      {
        "StdoutLog": "data:text/plain;charset=utf-8;base64,bWFpbGpbJ2RlbW8ucHknLCAnMyCsICcvbW50L2lucHV0JywgJy9tbnQvb3V0cHV0JywgJzEnXWZsdXNoIHRvIC9tbnQvb3V0cHV0LzE1MzU3MjYwMDQu0TEKZmx1c2ggdG8gL2ludC9vdXRwdXQvMTUzNTcyNjAwNi4zOQpmbHVzaCB0byAvbW50L291dHB1dC8xNTM1NzI2MDA3LjcK",
        "TaskInstanceIndex": 0,
        "StderrLog": "data:text/plain;charset=utf-8;base64,Zmx1c2ggdG8gL2ludC9vdXRwdXQvMTUzNTcyNjAwNC45MQpmbHVzaCB0byAvbW50L291dHB1dC8xNTM1NzI2MDA2LjM5CmZsdXNoIHRvIC9tbnQvb3V0cHV0LzE1MzU3MjYwMDcuNwo=",
        "StdoutRedirectPath": "",
        "StderrRedirectPath": "",
        "StdoutRedirectFileName": "",
        "StderrRedirectFileName": ""
      }
    ],
    "RequestId": "3b1ba3bf-a315-42ec-9e8e-af8def768f8a"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
InvalidParameter.TaskName	非法的任务名称。
InvalidParameter.TaskNameTooLong	任务名称过长。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.Negative	非法的负值参数。
ResourceNotFound.Job	指定作业不存在。
ResourceNotFound.Task	指定作业任务不存在。
ResourceNotFound.TaskInstance	指定任务实例不存在。
UnknownParameter	未知参数错误。

重试作业

最近更新时间：2025-05-15 01:10:03

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于重试作业中失败的任务实例。

仅当作业处于“FAILED”状态，支持重试操作。重试操作成功后，作业会按照有向无环图中指定的任务依赖关系，依次重试各个任务中失败的任务实例。任务实例的历史信息将被重置，如同首次运行一样，参与后续的调度和执行。

默认接口请求频率限制：10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：RetryJobs。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobIds.N	是	Array of String	作业ID列表。最大重试作业数100；JobId详见 作业列表 。 示例值：["job-rybewp57","job-97zcl3wt"]

3. 输出参数

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

4. 示例

示例1 重试作业

任务中出现部分实例失败时，仅重试失败的任务实例，已成功任务不作处理。

输入示例

```
https://batch.tencentcloudapi.com/?Action=RetryJobs
&JobIds.0=job-rybewp57
&JobIds.1=job-97zcl3wt
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "970e5a9f-2963-436c-8dad-4847360676f7"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
ResourceNotFound.Job	指定作业不存在。
UnsupportedOperation	操作不支持。

提交作业

最近更新时间：2025-05-15 01:10:03

1. 接口描述

接口请求域名：`batch.tencentcloudapi.com`。

用于提交一个作业

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：SubmitJob。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Placement	是	Placement	作业所提交的位置信息。通过该参数可以指定作业关联CVM所属可用区等信息。
Job	是	Job	作业信息
ClientToken	否	String	用于保证请求幂等性的字符串。该字符串由用户生成，需保证不同请求之间唯一，最大值不超过64个ASCII字符。若不指定该参数，则无法保证请求的幂等性。 示例值：fbad0aef-8441-4615-b904-c16779e33b

3. 输出参数

参数名称	类型	描述
JobId	String	当通过本接口来提交作业时返回该参数，表示一个作业ID。返回作业ID列表并不代表作业解析/运行成功，可根据 DescribeJob 接口查询其状态。 示例值：job-4yn0we13
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 提交作业

输入示例

```
https://batch.tencentcloudapi.com/?Action=SubmitJob
&Placement.Zone=ap-guangzhou-2
&Job.JobName=normal_job
&Job.JobDescription=normal_job
&Job.Priority=1
&Job.Tasks.0.TaskName=hello2
&Job.Tasks.0.TaskInstanceNum=1
&Job.Tasks.0.Application.DeliveryForm=LOCAL
&Job.Tasks.0.Application.Command=sleep 10
```

```
&Job.Tasks.0.ComputeEnv.EnvType=MANAGED
&Job.Tasks.0.ComputeEnv.EnvData.InstanceType=SA5.MEDIUM4
&Job.Tasks.0.ComputeEnv.EnvData.ImageId=img-bd78fy2t
&Job.Tasks.0.RedirectInfo.StdoutRedirectPath=cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/
&Job.Tasks.0.RedirectInfo.StderrRedirectPath=cos://dondonbatch-1251783334.cosgz.myqcloud.com/hello2/logs/
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "JobId": "job-4yn0we13",
    "RequestId": "bf2a9734-e485-423f-9d73-a93f5e6c6a0c"
  }
}
```

示例2 提交DAG型作业，具有依赖关系

输入示例

```
https://batch.tencentcloudapi.com/?Action=SubmitJob
&Placement.Zone=ap-beijing-2
&Job.JobName=dag
&Job.JobDescription=dag_demo
&Job.Tasks.0.TaskName=pre_task
&Job.Tasks.0.TaskInstanceNum=1
&Job.Tasks.0.Application.DeliveryForm=LOCAL
&Job.Tasks.0.Application.Command=echo pre_task
&Job.Tasks.0.ComputeEnv.EnvData.InstanceType=S2.SMALL1
&Job.Tasks.0.RedirectInfo.StdoutRedirectPath=cos://batch-demo-1251783334.cosbj.myqcloud.com/logs/
&Job.Tasks.0.RedirectInfo.StderrRedirectPath=cos://batch-demo-1251783334.cosbj.myqcloud.com/logs/
&Job.Tasks.1.TaskName=post_task
&Job.Tasks.1.TaskInstanceNum=1
&Job.Tasks.1.Application.DeliveryForm=LOCAL
&Job.Tasks.1.Application.Command=echo post_task
&Job.Tasks.1.ComputeEnv.EnvData.InstanceType=S2.SMALL1
&Job.Tasks.1.RedirectInfo.StdoutRedirectPath=cos://batch-demo-1251783334.cosbj.myqcloud.com/logs/
&Job.Tasks.1.RedirectInfo.StderrRedirectPath=cos://batch-demo-1251783334.cosbj.myqcloud.com/logs/
&Job.Dependences.0.StartTask=pre_task
&Job.Dependences.0.EndTask=post_task
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "JobId": "job-2bmV4p7o",
    "RequestId": "165dc518-52a4-4363-8ea4-14f1ee8dea1a"
  }
}
```

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
AllowedOneAttributeInEnvIdAndComputeEnv	ComputeEnv 和 EnvId 必须指定一个（且只有一个）参数。
InternalError	内部错误。
InternalError.CallCam	调用CAM API返回错误。
InternalError.CallCvm	调用CVM API返回错误。
InternalError.CallTagAPI	获取Tag错误。
InvalidParameter.CvmParameters	非法的CVM参数。
InvalidParameter.EnvIdMalformed	非法的计算环境ID格式。
InvalidParameter.ImageIdMalformed	镜像ID不正确。
InvalidParameter.InvalidParameterCombination	非法的参数组合。
InvalidParameter.JobDescriptionTooLong	作业描述过长。
InvalidParameter.JobNameTooLong	作业名称过长。
InvalidParameter.NotificationEventNameDuplicate	重复的消息通知事件名称。
InvalidParameter.NotificationTopicName	非法的主题名称。
InvalidParameter.NotificationTopicNameTooLong	主题名称过长。
InvalidParameter.TaskName	非法的任务名称。
InvalidParameter.TaskNameTooLong	任务名称过长。
InvalidParameterValue.ComputeEnv	计算环境参数校验失败。
InvalidParameterValue.DependenceNotFoundTaskName	找不到依赖任务定义。
InvalidParameterValue.DependenceUnfeasible	禁止环状任务依赖关系。
InvalidParameterValue.InstanceTypeDuplicate	实例类型不能有重复值。
InvalidParameterValue.InvalidDataTypeAny	DataTypeAny不合法。

错误码	描述
InvalidParameterValue.InvalidZoneMismatchRegion	可用区和地域不匹配。
InvalidParameterValue.LimitExceeded	Filter参数值数量超过限制。
InvalidParameterValue.LocalPath	非法的本地存储路径。
InvalidParameterValue.MaxRetryCount	重试次数过大。
InvalidParameterValue.Negative	非法的负值参数。
InvalidParameterValue.NotFloat	参数值不是浮点型。
InvalidParameterValue.RemoteStoragePath	非法的存储路径格式。
InvalidParameterValue.RemoteStorageSchemeType	非法的存储类型。
InvalidParameterValue.UnavailableZone	Zone不可用。
InvalidZone.MismatchRegion	指定的zone不存在。
LimitExceeded.JobQuota	作业配额不足。
ResourceNotFound.ComputeEnv	指定计算环境不存在。
UnauthorizedOperation.UserNotAllowedToUseBatch	禁止使用批量计算服务。

终止作业

最近更新時間：2025-05-15 01:10:03

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于终止作业。

当作业处于“SUBMITTED”状态时，禁止终止操作；当作业处于“SUCCEED”状态时，终止操作不会生效。

终止作业是一个异步过程。整个终止过程的耗时和任务总数成正比。终止的效果相当于所含的所有任务实例进行[TerminateTaskInstance](#)操作。具体效果和用法可参考[TerminateTaskInstance](#)。

默认接口请求频率限制：10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：TerminateJob。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID；JobId详见 作业列表 示例值：job-rybewp57

3. 输出参数

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

4. 示例

示例1 终止作业

输入示例

```
https://batch.tencentcloudapi.com/?Action=TerminateJob
&JobId=job-rybewp57
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "970e5a9f-2963-436c-8dad-4847360676f7"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
ResourceInUse.Job	Job在使用中。
ResourceNotFound.Job	指定作业不存在。
UnsupportedOperation	操作不支持。
UnsupportedOperation.TerminateOperationWithEnvId	指定计算环境的任务实例禁止该操作。

终止任务实例

最近更新时间：2025-05-15 01:10:03

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

用于终止任务实例。

对于状态已经为“SUCCEED”和“FAILED”的任务实例，不做处理。

对于状态为“SUBMITTED”、“PENDING”、“RUNNABLE”的任务实例，状态将置为“FAILED”状态。

对于状态为“STARTING”、“RUNNING”、“FAILED_INTERRUPTED”的任务实例，区分两种情况：如果未显示指定计算环境，会先销毁CVM服务器，然后将状态置为“FAILED”，具有一定耗时；如果指定了计算环境EnvId，任务实例状态置为“FAILED”，并重启执行该任务的CVM服务器，具有一定的耗时。

对于状态为“FAILED_INTERRUPTED”的任务实例，终止操作实际成功之后，相关资源和配额才会释放。

默认接口请求频率限制：2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：TerminateTaskInstance。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
JobId	是	String	作业ID；详见 作业列表 。 示例值：job-rybewp57
TaskName	是	String	任务名称；详见 作业提交信息 示例值：A
TaskInstanceId	是	Integer	任务实例索引 示例值：0

3. 输出参数

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

4. 示例

示例1 终止任务实例

输入示例

```
https://batch.tencentcloudapi.com/?Action=TerminateTaskInstance
&JobId=job-rybewp57
&TaskName=A
&TaskInstanceId=0
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "72e9a712-1fed-4b57-be72-62e71eefd4c3"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalError	内部错误。
InvalidParameter.JobIdMalformed	非法的作业ID格式。
ResourceNotFound.TaskInstance	指定任务实例不存在。
UnsupportedOperation.TerminateOperationWithEnvId	指定计算环境的任务实例禁止该操作。

查看配置相关接口

获取批量计算可用的CVM机型配置信息

最近更新时间：2025-05-15 01:10:02

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

查看可用的CVM机型配置信息

默认接口请求频率限制：2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：DescribeAvailableCvmInstanceTypes。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	过滤条件。 - zone – String – 是否必填：否 –（过滤条件）按照可用区过滤。 - instance-family String – 是否必填：否 –（过滤条件）按照机型系列过滤。实例机型系列形如：S1、I1、M1等。 示例值：[{"Name":"zone","Values":["ap-guangzhou-1"]}]

3. 输出参数

参数名称	类型	描述
InstanceTypeConfigSet	Array of InstanceTypeConfig	机型配置数组
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 查询广州三区 GN2 系列机型

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeAvailableCvmInstanceTypes
&Filters.0.Name=zone
&Filters.0.Values.0=ap-guangzhou-3
&Filters.1.Name=instance-family
&Filters.1.Values.0=GN2
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "InstanceTypeConfigSet": [
      {
        "Mem": 56,
        "Cpu": 28,
        "InstanceType": "GN2.7XLARGE56",
        "Zone": "ap-guangzhou-3",
        "InstanceFamily": "GN2"
      },
      {
        "Mem": 112,
        "Cpu": 56,
        "InstanceType": "GN2.14XLARGE112",
        "Zone": "ap-guangzhou-3",
        "InstanceFamily": "GN2"
      }
    ],
    "RequestId": "59f5b671-c492-4536-bbed-e5fbf78619dd"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallCam	调用CAM API返回错误。
InternalServerError.CallCvm	调用CVM API返回错误。
InvalidFilter	指定的Filter不被支持。

错误码	描述
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidFilter	Filter参数不正确。
InvalidParameterValue.InvalidZoneMismatchRegion	可用区和地域不匹配。
InvalidParameterValue.LimitExceeded	Filter参数值数量超过限制。
InvalidZone.MismatchRegion	指定的zone不存在。

获取批量计算可用区机型配置信息

最近更新时间：2025-05-15 01:10:02

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

获取批量计算可用区机型配置信息

默认接口请求频率限制：5次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeCvmZoneInstanceConfigInfos。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	过滤条件。 - zone - String - 是否必填：否 - （过滤条件）按照可用区过滤。 - instance-family String - 是否必填：否 - （过滤条件）按照机型系列过滤。实例机型系列形如：S1、I1、M1等。 - instance-type - String - 是否必填：否 - （过滤条件）按照机型过滤。实例机型形如：：S5.12XLARGE128、S5.12XLARGE96等。 - instance-charge-type - String - 是否必填：否 - （过滤条件）按照实例计费模式过滤。（POSTPAID_BY_HOUR：表示后付费，即按量计费机型 SPOTPAID：表示竞价付费机型。） 示例值：[{"Name":"zone","Values":["ap-guangzhou-1"]}]

3. 输出参数

参数名称	类型	描述
InstanceTypeQuotaSet	Array of InstanceTypeQuotaItem	可用区机型配置列表。
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 查看重庆一区按量计费机型配置信息

输入示例

```
https://batch.tencentcloudapi.com/?Action=DescribeCvmZoneInstanceConfigInfos
&Filters.0.Name=zone
&Filters.0.Values.0=ap-chongqing-1
&Filters.1.Name=instance-charge-type
&Filters.1.Values.0=POSTPAID_BY_HOUR
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "InstanceTypeQuotaSet": [
      {
        "Status": "SELL",
        "Zone": "ap-guangzhou-2",
        "NetworkCard": 0,
        "Price": {
          "DiscountPrice": 59,
          "OriginalPriceThreeYear": 0,
          "DiscountOneYear": 0,
          "UnitPrice": 0,
          "UnitPriceThirdStep": 0,
          "OriginalPriceFiveYear": 0,
          "Discount": 0,
          "DiscountFiveYear": 0,
          "UnitPriceDiscountThirdStep": 0,
          "UnitPriceSecondStep": 0,
          "OriginalPrice": 59,
          "DiscountThreeYear": 0,
          "UnitPriceDiscountSecondStep": 0,
          "UnitPriceDiscount": 0,
          "DiscountPriceFiveYear": 0,
          "OriginalPriceOneYear": 708,
          "ChargeUnit": "HOURL",
          "DiscountPriceThreeYear": 0,
          "DiscountPriceOneYear": 587.64
        },
        "InstanceFamily": "S2",
        "Externals": {},
        "InstanceType": "S2.SMALL1",
        "TypeName": "Standard S2",
        "InstanceChargeType": "POSTPAID_BY_HOUR",
        "Memory": 1,
        "LocalDiskTypeList": [],
        "Cpu": 1,
        "SoldOutReason": "",
        "StorageBlockAmount": 0,
        "CpuType": "Intel Xeon E5-2680 v4",
        "InstancePps": 20,
        "InstanceBandwidth": 1.5,
        "Gpu": 0,
        "GpuCount": 0,
        "Fpga": 0,
        "Remark": "",
        "Frequency": "2.4GHz",
        "StatusCategory": "EnoughStock"
      }
    ],
    "RequestId": "c28559ca-d3cf-40f0-9664-2ab303484efa"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalServerError	内部错误。
InternalServerError.CallCam	调用CAM API返回错误。
InternalServerError.CallCvm	调用CVM API返回错误。
InvalidFilter	指定的Filter不被支持。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidFilter	Filter参数不正确。
InvalidParameterValue.InvalidZoneMismatchRegion	可用区和地域不匹配。
InvalidParameterValue.LimitExceeded	Filter参数值数量超过限制。
InvalidParameterValue.UnsupportedBatchInstanceChargeType	批量计算不支持的机型付费类型。
InvalidZone.MismatchRegion	指定的zone不存在。

查询实例分类信息

最近更新时间：2025-05-15 01:10:02

1. 接口描述

接口请求域名：batch.tencentcloudapi.com。

目前对CVM现有实例族分类，每一类包含若干实例族。该接口用于查询实例分类信息。

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：DescribeInstanceCategories。
Version	是	String	公共参数 ，本接口取值：2017-03-12。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

3. 输出参数

参数名称	类型	描述
InstanceCategorySet	Array of InstanceCategoryItem	CVM实例分类列表 示例值：[{"InstanceFamilySet": ["C3","CN3"],"InstanceCategory": "COMPUTE_3"}]
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 查询实例分类信息

查询当前CVM实例族分类及分类详情。

输入示例

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

输出示例

```
{
  "Response": {
    "InstanceCategorySet": [
      {
        "InstanceFamilySet": [
          "C3",
          "CN3"
        ],
        "InstanceCategory": "COMPUTE_3"
      }
    ]
  }
}
```

```
{
  "InstanceFamilySet": [
    "C2"
  ],
  "InstanceCategory": "COMPUTE_2"
},
{
  "InstanceFamilySet": [
    "S2",
    "S2ne",
    "I2",
    "D1",
    "M2"
  ],
  "InstanceCategory": "GENERAL_2"
},
{
  "InstanceFamilySet": [
    "S3",
    "SN3ne",
    "S4",
    "I3",
    "M3",
    "M4"
  ],
  "InstanceCategory": "GENERAL_3"
}
],
"RequestId": "166a2a24-888f-4fd5-82ef-f41e4754d994"
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
InternalError	内部错误。
UnsupportedOperation	操作不支持。

最近更新时间：2025-05-30 01:10:50

Activity

计算环境的创建或销毁活动

被如下接口引用：DescribeComputeEnvActivities。

名称	类型	描述
ActivityId	String	活动ID 示例值：act-0akw6gb6
ComputeNodeId	String	计算节点ID 示例值：node-j8ybcrsh
ComputeNodeActivityType	String	计算节点活动类型，创建或者销毁 示例值：CREATE_COMPUTE_NODE
EnvId	String	计算环境ID 示例值：env-7iri4r84
Cause	String	起因 示例值：CreateComputeNode node-2bwng0f2
ActivityState	String	活动状态。取值范围： - SUBMITTED：已提交 - PROCESSING：处理中 - SUCCEED：成功 - FAILED：失败 示例值：SUCCEED
StateReason	String	状态原因 示例值：InternalError.CallCvm:CvmAPICode
StartTime	String	活动开始时间 示例值：2021-10-04T22:47:06Z
EndTime	String	活动结束时间 示例值：2021-10-05T01:02:14Z
InstanceId	String	云服务器实例ID 示例值：ins-qgks4wwf

AgentRunningMode

agent运行模式

被如下接口引用：CreateComputeEnv, CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Scene	String	是	场景类型，支持WINDOWS 示例值：WINDOWS
User	String	是	运行Agent的User 示例值：12348349197419
Session	String	是	运行Agent的Session 示例值：95af4dd1-c6cd-4294-9439-f55028895c62

AnonymousComputeEnv

计算环境

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
EnvType	String	否	计算环境管理类型 示例值：MANAGED
EnvData	EnvData	否	计算环境具体参数
MountDataDisks	Array of MountDataDisk	否	数据盘挂载选项
AgentRunningMode	AgentRunningMode	否	agent运行模式，适用于Windows系统

Application

应用程序信息

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
DeliveryForm	String	是	应用程序的交付方式，包括PACKAGE、LOCAL 两种取值，分别指远程存储的软件包、计算环境本地。 示例值：LOCAL
Command	String	否	松耦合任务执行命令。与Commands不能同时指定，一般使用Command字段提交任务。 示例值：sleep 10
PackagePath	String	否	应用程序软件包的远程存储路径 示例值：cos://kelvin-test-1251762227.cos.ap-nanjing.myqcloud.com/batch_test/
Docker	Docker	否	应用使用Docker的相关配置。在使用Docker配置的情况下，DeliveryForm 为 LOCAL 表示直接使用 Docker镜像内部的应用软件，通过Docker方式运行；DeliveryForm 为 PACKAGE，表示将远程应用包注入到Docker镜像后，通过Docker方式运行。为避免Docker不同版本的兼容性问题，Docker安装包及相关依赖由Batch统一负责，对于已安装Docker的自定义镜像，请卸载后再使用Docker特性。
Commands	Array of CommandLine	否	紧耦合任务执行命令信息。与Command不能同时指定。Command和Commands必须指定一个。

Authentication

授权认证信息

被如下接口引用：CreateComputeEnv, CreateTaskTemplate, DescribeComputeEnvCreateInfo, DescribeComputeEnvCreateInfos, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Scene	String	是	授权场景，例如COS 示例值：COS
SecretId	String	是	SecretId 示例值：AKIDUdi16HNjSuTI027Q9ThAzpuXljNx
SecretKey	String	是	SecretKey 示例值：Udi16HNjSuTI027Q9ThAzpuXljN

CommandLine

任务执行信息描述。

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Command	String	是	任务执行命令。 示例值：echo hello world

ComputeEnvCreateInfo

计算环境创建信息。

被如下接口引用：DescribeComputeEnvCreateInfos。

名称	类型	描述
EnvId	String	计算环境 ID 示例值：env-5o4gphir
EnvName	String	计算环境名称 示例值：demo
EnvDescription	String	计算环境描述 示例值：it is a first job
EnvType	String	计算环境类型，仅支持“MANAGED”类型 示例值：MANAGED
EnvData	EnvData	计算环境参数
MountDataDisks	Array of MountDataDisk	数据盘挂载选项
InputMappings	Array of InputMapping	输入映射
Authentications	Array of Authentication	授权信息
Notifications	Array of Notification	通知信息
DesiredComputeNodeCount	Integer	计算节点期望个数 示例值：0
Tags	Array of Tag	计算环境标签列表

ComputeEnvData

计算环境属性数据

被如下接口引用：ModifyComputeEnv。

名称	类型	必选	描述
InstanceTypes	Array of String	是	CVM实例类型列表 示例值：["SA2.MEDIUM4","SA5.MEDIUM4"]

ComputeEnvView

计算环境信息

被如下接口引用：DescribeComputeEnvs。

名称	类型	描述
EnvId	String	计算环境ID 示例值：env-5o4gphir
EnvName	String	计算环境名称 示例值：first_env
Placement	Placement	位置信息
CreateTime	String	创建时间 示例值：2020-11-11 15:00:33
ComputeNodeMetrics	ComputeNodeMetrics	计算节点统计指标
EnvType	String	计算环境类型 示例值：MANAGED
DesiredComputeNodeCount	Integer	计算节点期望个数 示例值：0

名称	类型	描述
ResourceType	String	计算环境资源类型，当前为CVM和CPM（黑石） 示例值：CVM
NextAction	String	下一步动作 示例值：DELETING
AttachedComputeNodeCount	Integer	用户添加到计算环境中的计算节点个数 示例值：1
Tags	Array of Tag	计算环境绑定的标签列表。 注意：此字段可能返回 null，表示取不到有效值。

ComputeNode

计算节点

被如下接口引用：DescribeComputeEnv。

名称	类型	描述
ComputeNodeId	String	计算节点ID 示例值：node-99xnkwf0
ComputeNodeInstanceId	String	计算节点实例ID，对于CVM场景，即为CVM的InstanceId 示例值：ins-mhkm3smc
ComputeNodeState	String	计算节点状态。取值范围： - PENDING：表示创建中 - SUBMITTED：表示已提交创建 - CREATING：表示创建中 - CREATED：表示创建完成 - CREATION_FAILED：表示创建失败。 - RUNNING：表示运行中。 - ABNORMAL：表示节点异常。 - DELETING：表示删除中。 示例值：RUNNING
Cpu	Integer	CPU核数 示例值：2
Mem	Integer	内存容量，单位GiB 示例值：4
ResourceCreatedTime	String	资源创建完成时间 示例值：2022-12-20T09:11:15Z
TaskInstanceNumAvailable	Integer	计算节点运行 TaskInstance 可用容量。0表示计算节点忙碌。 示例值：1
AgentVersion	String	Batch Agent 版本 示例值：1.4.4
PrivateIpAddresses	Array of String	实例内网IP 示例值：172.16.18.127
PublicIpAddresses	Array of String	实例公网IP 示例值：119.1.1.1
ResourceType	String	计算环境资源类型，当前为CVM和CPM（黑石） 示例值：CVM
ResourceOrigin	String	计算环境资源来源。 BATCH_CREATED：由批量计算创建的实例资源。 USER_ATTACHED：用户添加到计算环境中的实例资源。 示例值：BATCH_CREATED

ComputeNodeMetrics

计算节点统计指标

被如下接口引用：DescribeComputeEnv, DescribeComputeEnvs。

名称	类型	描述
SubmittedCount	Integer	已经完成提交的计算节点数量 示例值：1
CreatingCount	Integer	创建中的计算节点数量 示例值：0
CreationFailedCount	Integer	创建失败的计算节点数量 示例值：1
CreatedCount	Integer	完成创建的计算节点数量 示例值：0
RunningCount	Integer	运行中的计算节点数量 示例值：2
DeletingCount	Integer	销毁中的计算节点数量 示例值：1
AbnormalCount	Integer	异常的计算节点数量 示例值：1

DataDisk

描述了数据盘的信息

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
DiskSize	Integer	是	数据盘大小，单位：GiB。最小调整步长为10GiB，不同数据盘类型取值范围不同，具体限制详见： 存储概述 。默认值为0，表示不购买数据盘。更多限制详见产品文档。 示例值：50
DiskType	String	否	数据盘类型。数据盘类型限制详见 存储概述 。取值范围： - LOCAL_BASIC：本地硬盘 - LOCAL_SSD：本地SSD硬盘 - LOCAL_NVME：本地NVME硬盘，与InstanceType强相关，不支持指定 - LOCAL_PRO：本地HDD硬盘，与InstanceType强相关，不支持指定 - CLOUD_BASIC：普通云硬盘 - CLOUD_PREMIUM：高性能云硬盘 - CLOUD_SSD：SSD云硬盘 - CLOUD_HSSD：增强型SSD云硬盘 - CLOUD_TSSD：极速型SSD云硬盘 - CLOUD_BSSD：通用型SSD云硬盘 默认取值：LOCAL_BASIC 该参数对ResizeInstanceDisk接口无效。 示例值：CLOUD_SSD
DiskId	String	否	数据盘ID。 该参数目前仅用于DescribeInstances等查询类接口的返回参数，不可用于RunInstances等写接口的入参。 示例值：disk-ciezoimt
DeleteWithInstance	Boolean	否	数据盘是否随子机销毁。取值范围： - true：子机销毁时，销毁数据盘，只支持按小时后付费云盘 - false：子机销毁时，保留数据盘 默认取值：true

名称	类型	必选	描述
			该参数目前仅用于 <code>RunInstances</code> 接口。 示例值：true
SnapshotId	String	否	数据盘快照ID。选择的数据盘快照大小需小于数据盘大小。 示例值：snap-6yczrj8x
Encrypt	Boolean	否	数据盘是否加密。取值范围： - true：加密 - false：不加密 默认取值：false 该参数目前仅用于 <code>RunInstances</code> 接口。 示例值：false
KmsKeyId	String	否	自定义CMK对应的ID，取值为UUID或者类似kms-abcd1234。用于加密云盘。 该参数目前仅用于 <code>RunInstances</code> 接口。 注意：此字段可能返回 null，表示取不到有效值。 示例值：kms-abcd1234
ThroughputPerformance	Integer	否	云硬盘性能，单位：MiB/s。使用此参数可给云硬盘购买额外的性能。 当前仅支持极速型云盘（CLOUD_TSSD）和增强型SSD云硬盘（CLOUD_HSSD） 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
CdId	String	否	所属的独享集群ID。 注意：此字段可能返回 null，表示取不到有效值。 示例值：cdc-1648zauv
BurstPerformance	Boolean	否	突发性能 注：内测中。 示例值：false
DiskName	String	否	磁盘名称，长度不超过128 个字符。 示例值：MyDiskName

Dependence

依赖关系

被如下接口引用：DescribeJob, DescribeJobSubmitInfo, SubmitJob。

名称	类型	必选	描述
StartTask	String	是	依赖关系的起点任务名称 示例值：TaskA
EndTask	String	是	依赖关系的终点任务名称 示例值：TaskB

Docker

Docker容器信息

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Image	String	是	Docker Hub填写 “[user/repo]:[tag]” ，Tencent Registry填写 “ccr.ccs.tencentyun.com/[namespace/re 示例值：kelvin-batch-test.tencentcloudcr.com/test/kelvin-test:v1
User	String	否	Docker Hub 用户名或 Tencent Registry 用户名；公共镜像可不填写此参数。 示例值：22372874874
Password	String	否	Docker Hub 密码或 Tencent Registry 密码；公共镜像可不填写此参数。 示例值： yJhbGciOiJSUzI1NiIsImtpZCI6IjJEQUg6UkNMUDpSSENTOkxNQ1g6UVBHTjpFWlhGOlBYTEo6Mlps

名称	类型	必选	描述
Server	String	否	Docker Hub 可以不填，但确保具有公网访问能力。或者是 Tencent Registry 服务地址 “ccr.ccs.tencentyun.com” 示例值：kelvin-batch-test.tencentcloudcr.com
MaxRetryCount	Integer	否	拉取Docker镜像重试次数。默认值：0。 示例值：0
DelayOnRetry	Integer	否	拉取Docker镜像失败时延迟时间。单位：秒。 示例值：1
DockerRunOption	String	否	Docker命令运行参数。 示例值：-u root

EnhancedService

描述了实例的增强服务启用情况与其设置，如云安全，云监控等实例 Agent

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
SecurityService	RunSecurityServiceEnabled	否	开启云安全服务。若不指定该参数，则默认开启云安全服务。 示例值：true
MonitorService	RunMonitorServiceEnabled	否	开启云监控服务。若不指定该参数，则默认开启云监控服务。 示例值：true
AutomationService	RunAutomationServiceEnabled	否	开启云自动化助手服务（TencentCloud Automation Tools，TAT）。若不指定该参数，则公共镜像默认开启云自动化助手服务，其他镜像默认不开启云自动化助手服务。 示例值：true

EnvData

计算环境数据

被如下接口引用：CreateComputeEnv, CreateTaskTemplate, DescribeComputeEnvCreateInfo, DescribeComputeEnvCreateInfos, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
InstanceType	String	否	CVM实例类型，不能与InstanceTypes和InstanceTypeOptions同时出现。 示例值：SA5.MEDIUM2
ImageId	String	否	CVM镜像ID 示例值：img-jsdt6r36
SystemDisk	SystemDisk	否	实例系统盘配置信息
DataDisks	Array of DataDisk	否	实例数据盘配置信息
VirtualPrivateCloud	VirtualPrivateCloud	否	私有网络相关信息配置，与Zones和VirtualPrivateClouds不能同时指定。
InternetAccessible	InternetAccessible	否	公网带宽相关信息设置
InstanceName	String	否	CVM实例显示名称 示例值：demo
LoginSettings	LoginSettings	否	实例登录设置
SecurityGroupIds	Array of String	否	实例所属安全组 示例值：["sg-dec9tgij"]
EnhancedService	EnhancedService	否	增强服务。通过该参数可以指定是否开启云安全、云监控等服务。若不指定该参数，则默认开启云监控、云安全服务。
InstanceChargeType	String	否	CVM实例计费类型 - POSTPAID_BY_HOUR：按小时后付费 - SPOTPAID：竞价付费

名称	类型	必选	描述
			默认值：POSTPAID_BY_HOUR。 示例值：POSTPAID_BY_HOUR
InstanceMarketOptions	InstanceMarketOptionsRequest	否	实例的市场相关选项，如竞价实例相关参数
InstanceTypes	Array of String	否	CVM实例类型列表，不能与InstanceType和InstanceTypeOptions同时出现。指定该字段后，计算节点按照机型先后顺序依次尝试创建，直到实例创建成功，结束遍历过程。最多支持10个机型。 示例值：["SA5.MEDIUM2"]
InstanceTypeOptions	InstanceTypeOptions	否	CVM实例机型配置。不能与InstanceType和InstanceTypes同时出现。
Zones	Array of String	否	可用区列表，支持跨可用区创建CVM实例。与VirtualPrivateCloud和VirtualPrivateClouds不能同时指定。 示例值：["ap-guangzhou-2"]
VirtualPrivateClouds	Array of VirtualPrivateCloud	否	私有网络列表，支持跨私有网络创建CVM实例。与VirtualPrivateCloud和Zones不能同时指定。

EnvVar

环境变量

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Name	String	是	环境变量名称 示例值：batch_demo
Value	String	是	环境变量取值 示例值：first_demo

EventConfig

事件配置

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo, SubmitJob。

名称	类型	必选	描述
EventName	String	是	事件类型，包括： “JOB_RUNNING”：作业运行，适用于"SubmitJob"。 “JOB_SUCCEED”：作业成功，适用于"SubmitJob"。 “JOB_FAILED”：作业失败，适用于"SubmitJob"。 “JOB_FAILED_INTERRUPTED”：作业失败，保留实例，适用于"SubmitJob"。 “TASK_RUNNING”：任务运行，适用于"SubmitJob"。 “TASK_SUCCEED”：任务成功，适用于"SubmitJob"。 “TASK_FAILED”：任务失败，适用于"SubmitJob"。 “TASK_FAILED_INTERRUPTED”：任务失败，保留实例，适用于"SubmitJob"。 “TASK_INSTANCE_RUNNING”：任务实例运行，适用于"SubmitJob"。 “TASK_INSTANCE_SUCCEED”：任务实例成功，适用于"SubmitJob"。 “TASK_INSTANCE_FAILED”：任务实例失败，适用于"SubmitJob"。 “TASK_INSTANCE_FAILED_INTERRUPTED”：任务实例失败，保留实例，适用于"SubmitJob"。 “COMPUTE_ENV_CREATED”：计算环境已创建，适用于"CreateComputeEnv"。 “COMPUTE_ENV_DELETED”：计算环境已删除，适用于"CreateComputeEnv"。 “COMPUTE_NODE_CREATED”：计算节点已创建，适用于"CreateComputeEnv"和"SubmitJob"。 “COMPUTE_NODE_CREATION_FAILED”：计算节点创建失败，适用于"CreateComputeEnv"和"SubmitJob"。 “COMPUTE_NODE_RUNNING”：计算节点运行中，适用于"CreateComputeEnv"和"SubmitJob"。 “COMPUTE_NODE_ABNORMAL”：计算节点异常，适用于"CreateComputeEnv"和"SubmitJob"。 “COMPUTE_NODE_DELETING”：计算节点已删除，适用于"CreateComputeEnv"和"SubmitJob"。

名称	类型	必选	描述
			示例值：JOB_RUNNING
EventVars	Array of EventVar	是	自定义键值对

EventVar

自定义键值对

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo, SubmitJob。

名称	类型	必选	描述
Name	String	是	自定义键 示例值：batch_demo
Value	String	是	自定义值 示例值：first-demo

Externals

扩展数据

被如下接口引用：DescribeCvmZoneInstanceConfigInfos。

名称	类型	必选	描述
ReleaseAddress	Boolean	否	释放地址 示例值：true
UnsupportNetworks	Array of String	否	不支持的网络类型，取值范围： - BASIC：基础网络 - VPC1.0：私有网络VPC1.0 示例值：VPC1.0
StorageBlockAttr	StorageBlock	否	HDD本地存储属性

Filter

描述键值对过滤器，用于条件过滤查询。例如过滤ID、名称、状态等

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

以[DescribeInstances](#)接口的Filter为例。若我们需要查询可用区（zone）为广州一区 **并且** 实例计费模式（instance-charge-type）为包年包月 **或者** 按量计费的实例时，可如下实现：

```
Filters.0.Name=zone
&Filters.0.Values.0=ap-guangzhou-1
&Filters.1.Name=instance-charge-type
&Filters.1.Values.0=PREPAID
&Filters.1.Values.1=POSTPAID_BY_HOUR
```

被如下接口引用：DescribeAvailableCvmInstanceTypes, DescribeComputeEnvActivities, DescribeComputeEnvCreateInfos, DescribeComputeEnvs, DescribeCvmZoneInstanceConfigInfos, DescribeJobs, DescribeTask, DescribeTaskTemplates。

名称	类型	必选	描述
Name	String	是	需要过滤的字段。 示例值：zone
Values	Array of String	是	字段的过滤值。 示例值：["ap-guangzhou-2"]

InputMapping

输入映射

被如下接口引用：CreateComputeEnv, CreateTaskTemplate, DescribeComputeEnvCreateInfo, DescribeComputeEnvCreateInfos, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
SourcePath	String	是	源端路径 示例值：cos://kelvin-test-1251762227.cos.ap-nanjing.myqcloud.com/batch_test/
DestinationPath	String	是	目的端路径 示例值：/home/input/
MountOptionParameter	String	否	挂载配置项参数 示例值：-odbglevel=info

Instance

描述实例的信息

被如下接口引用：AttachInstances。

名称	类型	必选	描述
InstanceId	String	是	实例ID，可通过调用接口 DescribeInstances 获取。 示例值：ins-makv4qr2
ImageId	String	否	镜像ID，可通过调用接口 DescribeImages 获取。 示例值：img-fy8cew7d
LoginSettings	LoginSettings	否	实例登录设置。

InstanceCategoryItem

实例分类列表

被如下接口引用：DescribeInstanceCategories。

名称	类型	描述
InstanceCategory	String	实例类型名 示例值：COMPUTE_3
InstanceFamilySet	Array of String	实例族列表 示例值：["C3", "CN3"]

InstanceMarketOptionsRequest

竞价请求相关选项

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
SpotOptions	SpotMarketOptions	是	竞价相关选项 示例值：0.12

名称	类型	必选	描述
MarketType	String	否	市场选项类型，当前只支持取值：spot 示例值：spot

InstanceTypeConfig

批量计算可用的InstanceTypeConfig信息

被如下接口引用：DescribeAvailableCvmInstanceTypes。

名称	类型	描述
Mem	Integer	内存容量，单位：GB。 示例值：4
Cpu	Integer	CPU核数，单位：核。 示例值：2
InstanceType	String	实例机型。 示例值：S2.3XLARGE24
Zone	String	可用区。 示例值：ap-guangzhou-2
InstanceFamily	String	实例机型系列。 示例值：S2

InstanceTypeOptions

实例机型配置。

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
CPU	Integer	是	CPU核数。 示例值：2
Memory	Integer	是	内存值，单位GB。 示例值：4
InstanceCategories	Array of String	否	实例机型类别，可选参数：“ALL”、“GENERAL”、“GENERAL_2”、“GENERAL_3”、“COMPUTE”、“COMPUTE_2” 默认值“ALL”。 示例值：COMPUTE_2

InstanceTypeQuotaltem

描述实例机型配额信息。

被如下接口引用：DescribeCvmZoneInstanceConfigInfos。

名称	类型	描述
Zone	String	可用区。 示例值：ap-guangzhou-2
InstanceType	String	实例机型。 示例值：S5.LARGE4
InstanceChargeType	String	实例计费模式。取值范围： - PREPAID：表示预付费，即包年包月 - POSTPAID_BY_HOUR：表示后付费，即按量计费 - CDHPAID：表示专用宿主机付费，即只对专用宿主机计费，不对专用宿主机上的实例计费。 - SPOTPAID：表示竞价实例付费。 示例值：PREPAID

名称	类型	描述
NetworkCard	Integer	网卡类型，例如：25代表25G网卡 示例值：25
Externals	Externals	扩展属性。
Cpu	Integer	实例的CPU核数，单位：核。 示例值：4
Memory	Integer	实例内存容量，单位：GB。 示例值：4
InstanceFamily	String	实例机型系列。 示例值：S5
TypeName	String	机型名称。 示例值：标准型S5
LocalDiskTypeList	Array of LocalDiskType	本地磁盘规格列表。当该参数返回为空值时，表示当前情况下无法创建本地盘。
Status	String	实例是否售卖。取值范围： - SELL：表示实例可购买 - SOLD_OUT：表示实例已售罄。 示例值：SELL
Price	ItemPrice	实例的售卖价格。
SoldOutReason	String	售罄原因。 示例值：ResourcesSoldOut.SpecifiedInstanceType
InstanceBandwidth	Float	内网带宽，单位Gbps。 示例值：10.0
InstancePps	Integer	网络收发包能力，单位万PPS。 示例值：80
StorageBlockAmount	Integer	本地存储块数量。 示例值：10
CpuType	String	处理器型号。 示例值：Intel Xeon Cascade Lake 8255C
Gpu	Integer	实例的GPU数量。 示例值：1
Fpga	Integer	实例的FPGA数量。 示例值：1
Remark	String	实例备注信息。 示例值：搭载1块 11176 GB SATA HDD 本地硬盘
GpuCount	Float	实例机型映射的物理GPU卡数，单位：卡。vGPU卡型小于1，直通卡型大于等于1。vGPU是通过分片虚拟化技术，将物理GPU卡重新划分，同一块GPU卡经虚拟化分割后可分配至不同的实例使用。直通卡型会将GPU设备直接挂载给实例使用。 示例值：1.0
Frequency	String	实例的CPU主频信息 示例值：-/3.1GHz
StatusCategory	String	描述库存情况。取值范围： - EnoughStock：表示对应库存非常充足 - NormalStock：表示对应库存供应有保障 - UnderStock：表示对应库存即将售罄 - WithoutStock：表示对应库存已经售罄 示例值：EnoughStock

InternetAccessible

描述了实例的公网可访问性，声明了实例的公网使用计费模式，最大带宽等

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
InternetChargeType	String	否	网络计费类型。取值范围： - BANDWIDTH_PREPAID：预付费按带宽结算 - TRAFFIC_POSTPAID_BY_HOUR：流量按小时后付费 - BANDWIDTH_POSTPAID_BY_HOUR：带宽按小时后付费 - BANDWIDTH_PACKAGE：带宽包用户 默认取值：非带宽包用户默认与子机付费类型保持一致，比如子机付费类型为预付费，网络计费类型默认为预付费；子机付费类型为后付费，网络计费类型默认为后付费。 示例值：BANDWIDTH_PREPAID
InternetMaxBandwidthOut	Integer	否	公网出带宽上限，单位：Mbps。默认值：0Mbps。不同机型带宽上限范围不一致，具体限制详见 购买网络带宽 。 示例值：10
PublicIpAssigned	Boolean	否	是否分配公网IP。取值范围： - true：表示分配公网IP - false：表示不分配公网IP 当公网带宽大于0Mbps时，可自由选择开通与否，默认开通公网IP；当公网带宽为0，则不允许分配公网IP。该参数仅在RunInstances接口中作为入参使用。 示例值：false
BandwidthPackageId	String	否	带宽包ID。可通过 DescribeBandwidthPackages 接口返回值中的BandwidthPackageId获取。该参数仅在RunInstances接口中作为入参使用。 示例值：bwp-h1ro

ItemPrice

描述了单项的价格信息

被如下接口引用：DescribeCvmZoneInstanceConfigInfos。

名称	类型	描述
UnitPrice	Float	后续合计费用的原价，后付费模式使用，单位：元。 如返回了其他时间区间项，如UnitPriceSecondStep，则本项代表时间区间在(0, 96)小时；若未返回其他时间区间项，则本项代表全时段，即(0, ∞)小时 示例值：35.5
ChargeUnit	String	后续计价单元，后付费模式使用，可取值范围： - hour：表示计价单元是按每小时来计算。当前涉及该计价单元的场景有：实例按小时后付费（POSTPAID_BY_HOUR）、带宽按小时后付费（BANDWIDTH_POSTPAID_BY_HOUR）； - gb：表示计价单元是按每GB来计算。当前涉及该计价单元的场景有：流量按小时后付费（TRAFFIC_POSTPAID_BY_HOUR）。 示例值：hour
OriginalPrice	Float	预支合计费用的原价，预付费模式使用，单位：元。 示例值：35.5
DiscountPrice	Float	预支合计费用的折扣价，预付费模式使用，单位：元。 示例值：35.5
Discount	Float	折扣，如20.0代表2折。 示例值：20.0
UnitPriceDiscount	Float	后续合计费用的折扣价，后付费模式使用，单位：元。 如返回了其他时间区间项，如UnitPriceDiscountSecondStep，则本项代表时间区间在(0, 96)小时；若未返回其他时间区间项，则本项代表全时段，即(0, ∞)小时 示例值：35.5
UnitPriceSecondStep	Float	使用时间区间在(96, 360)小时的后续合计费用的原价，后付费模式使用，单位：元。

名称	类型	描述
		示例值：35.5
UnitPriceDiscountSecondStep	Float	使用时间区间在(96, 360)小时的后续合计费用的折扣价，后付费模式使用，单位：元 示例值：35.5
UnitPriceThirdStep	Float	使用时间区间在(360, ∞)小时的后续合计费用的原价，后付费模式使用，单位：元。 示例值：35.5
UnitPriceDiscountThirdStep	Float	使用时间区间在(360, ∞)小时的后续合计费用的折扣价，后付费模式使用，单位：元 示例值：35.5
OriginalPriceThreeYear	Float	预支三年合计费用的原价，预付费模式使用，单位：元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
DiscountPriceThreeYear	Float	预支三年合计费用的折扣价，预付费模式使用，单位：元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
DiscountThreeYear	Float	预支三年应用的折扣，如20.0代表2折。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
OriginalPriceFiveYear	Float	预支五年合计费用的原价，预付费模式使用，单位：元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
DiscountPriceFiveYear	Float	预支五年合计费用的折扣价，预付费模式使用，单位：元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
DiscountFiveYear	Float	预支五年应用的折扣，如20.0代表2折。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
OriginalPriceOneYear	Float	预支一年合计费用的原价，预付费模式使用，单位：元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
DiscountPriceOneYear	Float	预支一年合计费用的折扣价，预付费模式使用，单位：元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：35.5
DiscountOneYear	Float	预支一年应用的折扣，如20.0代表2折。 注意：此字段可能返回 null，表示取不到有效值。 示例值：20.0

Job

作业

被如下接口引用：SubmitJob。

名称	类型	必选	描述
Tasks	Array of Task	是	任务信息
JobName	String	否	作业名称；字符串长度限制60。 示例值：test-job
JobDescription	String	否	作业描述；字符串长度限制200。 示例值：test-job
Priority	Integer	否	作业优先级，任务（Task）和任务实例（TaskInstance）会继承作业优先级；范围0~100，数值越大，优先级越高。 示例值：1
Dependences	Array of Dependence	否	依赖信息

名称	类型	必选	描述
Notifications	Array of Notification	否	通知信息
TaskExecutionDependOn	String	否	对于存在依赖关系的任务中，后序任务执行对于前序任务的依赖条件。取值范围包括 PRE_TASK_SUCCEED，PRE_TASK_AT_LEAST_PARTLY_SUCCEED，PRE_TASK_FINISHED，默认值为PRE_TASK_SUCCEED。 示例值：PRE_TASK_SUCCEED
StatelfCreateCvmFailed	String	否	表示创建 CVM 失败按照何种策略处理。取值范围包括 FAILED，RUNNABLE。FAILED 表示创建 CVM 失败按照一次执行失败处理，RUNNABLE 表示创建 CVM 失败按照继续等待处理。默认值为FAILED。StatelfCreateCvmFailed对于提交的指定计算环境的作业无效。 示例值：FAILED
Tags	Array of Tag	否	标签列表。通过指定该参数可以支持绑定标签到作业。每个作业最多绑定10个标签。
NotificationTarget	String	否	表示通知信息的通知目标类型。 取值范围：CMQ，TDMQ_CMQ。 CMQ:表示向腾讯云CMQ发送消息。 TDMQ_CMQ：表示向腾讯云TDMQ_CMQ发送消息。 默认值为CMQ。 注：腾讯云计划于2022年6月前正式下线消息队列 CMQ，建议使用TDMQ_CMQ。参考文档： CMQ迁移到TDMQ_CMQ 示例值：TDMQ_CMQ

JobView

作业信息

被如下接口引用：DescribeJobs。

名称	类型	描述
JobId	String	作业ID；JobId详见 作业列表 示例值：job-f4llnqoz
JobName	String	作业名称 示例值：job-asdnds0143100_N13_3
JobState	String	作业状态： - SUBMITTED：已提交； - PENDING：等待中； - RUNNABLE：可运行； - STARTING：启动中； - RUNNING：运行中； - SUCCEED：成功； - FAILED：失败； - FAILED_INTERRUPTED：失败后保留实例。 示例值：RUNNING
Priority	Integer	作业优先级 示例值：1
Placement	Placement	位置信息
CreateTime	String	创建时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ 示例值：2021-05-07T04:19:49Z
EndTime	String	结束时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ 示例值：2021-05-07T04:22:06Z
TaskMetrics	TaskMetrics	任务统计指标
Tags	Array of Tag	作业绑定的标签列表。

LocalDiskType

本地磁盘规格

被如下接口引用：DescribeCvmZoneInstanceConfigInfos。

名称	类型	描述
Type	String	本地磁盘类型。 示例值：LOCAL_BASIC
PartitionType	String	本地磁盘属性。 示例值：ROOT
MinSize	Integer	本地磁盘最小值。 示例值：10
MaxSize	Integer	本地磁盘最大值。 示例值：20
Required	String	购买时本地盘是否为必选。取值范围： - REQUIRED：表示必选 - OPTIONAL：表示可选。 示例值：REQUIRED

LoginSettings

描述了实例登录相关配置与信息。

被如下接口引用：AttachInstances, CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
Password	String	否	实例登录密码。不同操作系统类型密码复杂度限制不一样，具体如下： Linux实例密码必须8到16位，至少包括两项[a-z, A-Z]、[0-9] 和 [() ~ ! @ # \$ % ^ & * - + = &#124; { } [] : ; ' , . ? \ /] 中的特殊符号。 Windows实例密码必须12到16位，至少包括三项[a-z]，[A-Z]，[0-9] 和 [() ~ ! @ # \$ % ^ & * - + = { } [] : ; ' , . ? /]中的特殊符号。 若不指定该参数，则由系统随机生成密码，并通过站内信方式通知到用户。 示例值：cbak231r7@#
KeyIds	Array of String	否	密钥ID列表。关联密钥后，就可以通过对应的私钥来访问实例；KeyId可通过接口DescribeKeyPairs获取，密钥与密码不能同时指定，同时Windows操作系统不支持指定密钥。当前仅支持购买的时候指定一个密钥。 示例值：nackdsnlana@#!@#
KeepImageLogin	String	否	保持镜像的原始设置。该参数与Password或KeyIds.N不能同时指定。只有使用自定义镜像、共享镜像或外部导入镜像创建实例时才能指定该参数为TRUE。取值范围： - TRUE：表示保持镜像的登录设置 - FALSE：表示不保持镜像的登录设置 默认取值：FALSE。 示例值：FALSE

MountDataDisk

数据盘挂载选项

被如下接口引用：CreateComputeEnv, CreateTaskTemplate, DescribeComputeEnvCreateInfo, DescribeComputeEnvCreateInfos, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
LocalPath	String	是	挂载点，Linux系统合法路径，或Windows系统盘符,比如"H:\" 示例值：/data/tmp
FileSystemType	String	否	文件系统类型，Linux系统下支持"EXT3"和"EXT4"两种，默认"EXT3"；Windows系统下仅支持"NTFS" 示例值：EXT3

NamedComputeEnv

计算环境

被如下接口引用：CreateComputeEnv。

名称	类型	必选	描述
EnvName	String	是	计算环境名称 示例值：test-env
DesiredComputeNodeCount	Integer	是	计算节点期望个数，最大上限2000。 示例值：1
EnvDescription	String	否	计算环境描述 示例值：test-env
EnvType	String	否	计算环境管理类型，枚举如下： MANAGED: 由客户在Batch平台主动创建； THPC_QUEUE: 由THPC平台创建，关联THPC平台的集群队列。 示例值：MANAGED
EnvData	EnvData	否	计算环境具体参数
MountDataDisks	Array of MountDataDisk	否	数据盘挂载选项
Authentications	Array of Authentication	否	授权信息
InputMappings	Array of InputMapping	否	输入映射信息
AgentRunningMode	AgentRunningMode	否	agent运行模式，适用于Windows系统
Notifications	Array of Notification	否	通知信息
ActionIfComputeNodeInactive	String	否	非活跃节点处理策略，默认“RECREATE”，即对于实例创建失败或异常退还的计算节点，定期重新创建实例资源。 示例值：RECREATE
ResourceMaxRetryCount	Integer	否	对于实例创建失败或异常退还的计算节点，定期重新创建实例资源的最大重试次数，最大值100，如果不设置的话，系统会设置一个默认值，当前为7 示例值：1
Tags	Array of Tag	否	标签列表。通过指定该参数可以支持绑定标签到计算环境。每个计算环境最多绑定10个标签。
NotificationTarget	String	否	表示通知信息的通知目标类型。 取值范围：CMQ，TDMQ_CMQ。 CMQ:表示向腾讯云CMQ发送消息。 TDMQ_CMQ: 表示向腾讯云TDMQ_CMQ发送消息。 默认值为CMQ。 注：腾讯云计划于2022年6月前正式下线消息队列 CMQ，建议使用TDMQ_CMQ。 参考文档： CMQ迁移到TDMQ_CMQ 示例值：TDMQ_CMQ

Notification

通知信息

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo, DescribeComputeEnvCreateInfos, SubmitJob。

名称	类型	必选	描述
TopicName	String	是	CMQ主题名字，要求主题名有效且关联订阅 示例值：demo
EventConfigs	Array of EventConfig	是	事件配置

OutputMapping

输出映射

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
SourcePath	String	是	源端路径 示例值：cos://kelvin-test-125176278734.cos.ap-nanjing.myqcloud.com/batch_test/
DestinationPath	String	是	目的端路径 示例值：/home/output/
OutputMappingOption	OutputMappingOption	否	输出映射选项

OutputMappingConfig

输出映射配置

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Scene	String	是	存储类型，仅支持COS 示例值：COS
WorkerNum	Integer	是	并行worker数量 示例值：5
WorkerPartSize	Integer	是	worker分块大小，单位MB 示例值：100

OutputMappingOption

输出映射选项

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Workspace	String	否	容器场景下,输出选项从实例映射到容器内的实例侧的工作空间。 BATCH_WORKSPACE: 工作空间为BATCH在实例内定义的工作空间，BATCH侧保证作业之间的隔离。（默认） GLOBAL_WORKSPACE: 工作空间为实例操作系统空间。 示例值：BATCH_WORKSPACE

Placement

描述了实例的抽象位置，包括其所在的可用区，所属的项目，宿主机（仅专用宿主机产品可用），母机IP等

被如下接口引用：CreateComputeEnv, DescribeComputeEnv, DescribeComputeEnvs, DescribeJobs, SubmitJob。

名称	类型	必选	描述
Zone	String	是	实例所属的可用区名称。该参数可以通过调用 DescribeZones 的返回值中的Zone字段来获取。 示例值：ap-guangzhou-4
ProjectId	Integer	否	实例所属项目ID。该参数可以通过调用 DescribeProject 的返回值中的ProjectId字段来获取。默认取值0，表示默认项目。 示例值：0
HostIds	Array of String	否	实例所属的专用宿主机ID列表，仅用于入参。如果您有购买专用宿主机并且指定了该参数，则您购买的实例就会随机的部署在这些专用宿主机上。该参数可以通过调用 DescribeHosts 的返回值中的HostId字段来获取。 示例值：["host-o5i742go"]
HostId	String	否	实例所属的专用宿主机ID，仅用于出参。 示例值：host-o5i742go

RedirectInfo

重定向信息

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, DescribeTask, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
StdoutRedirectPath	String	否	标准输出重定向路径; 示例值: cos://kelvin-test-1351762227.cos.ap-nanjing.myqcloud.com/batch_test/
StderrRedirectPath	String	否	标准错误重定向路径 示例值: cos://kelvin-test-1351762227.cos.ap-nanjing.myqcloud.com/batch_test/
StdoutRedirectFileName	String	否	标准输出重定向文件名, 支持三个占位符\${BATCH_JOB_ID}、\${BATCH_TASK_NAME}、 \${BATCH_TASK_INSTANCE_INDEX} 示例值: demo_stdout.log
StderrRedirectFileName	String	否	标准错误重定向文件名, 支持三个占位符\${BATCH_JOB_ID}、\${BATCH_TASK_NAME}、 \${BATCH_TASK_INSTANCE_INDEX} 示例值: demo_stderr.log

RedirectLocalInfo

本地重定向信息

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
StdoutLocalPath	String	否	标准输出重定向本地路径 示例值: /tmp/log/
StderrLocalPath	String	否	标准错误重定向本地路径 示例值: /tmp/log/
StdoutLocalFileName	String	否	标准输出重定向本地文件名, 支持三个占位符\${BATCH_JOB_ID}、\${BATCH_TASK_NAME}、 \${BATCH_TASK_INSTANCE_INDEX} 示例值: job-5ewdqncb-stdout.log
StderrLocalFileName	String	否	标准错误重定向本地文件名, 支持三个占位符\${BATCH_JOB_ID}、\${BATCH_TASK_NAME}、 \${BATCH_TASK_INSTANCE_INDEX} 示例值: job-5ewdqncb-stderr.log

RunAutomationServiceEnabled

描述了“云自动化助手”服务相关的信息

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
Enabled	Boolean	否	是否开启云自动化助手。取值范围： - true：表示开启云自动化助手服务 - false：表示不开启云自动化助手服务 默认取值: false。 示例值: true

RunMonitorServiceEnabled

描述了“云监控”服务相关的信息

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
Enabled	Boolean	否	是否开启云监控服务。取值范围： - true：表示开启云监控服务 - false：表示不开启云监控服务

名称	类型	必选	描述
			默认取值：true。 示例值：true

RunSecurityServiceEnabled

描述了“云安全”服务相关的信息

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
Enabled	Boolean	否	是否开启 云安全 服务。取值范围： - true：表示开启云安全服务 - false：表示不开启云安全服务 默认取值：true。 示例值：false

SpotMarketOptions

竞价相关选项

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
MaxPrice	String	是	竞价出价 示例值：0.01
SpotInstanceType	String	否	竞价请求类型，当前仅支持类型：one-time 示例值：one-time

StorageBlock

HDD的本地存储信息

被如下接口引用：DescribeCvmZoneInstanceConfigInfos。

名称	类型	必选	描述
Type	String	否	HDD本地存储类型，值为：LOCAL_PRO。 示例值：LOCAL_PRO
MinSize	Integer	否	HDD本地存储的最小容量。单位：GiB。 示例值：100
MaxSize	Integer	否	HDD本地存储的最大容量。单位：GiB。 示例值：200

SystemDisk

描述了操作系统所在块设备即系统盘的信息

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
DiskType	String	否	系统盘类型。系统盘类型限制详见 存储概述 。取值范围： - LOCAL_BASIC：本地硬盘 - LOCAL_SSD：本地SSD硬盘 - CLOUD_BASIC：普通云硬盘 - CLOUD_SSD：SSD云硬盘 - CLOUD_PREMIUM：高性能云硬盘

名称	类型	必选	描述
			- CLOUD_BSSD：通用型SSD云硬盘 - CLOUD_HSSD：增强型SSD云硬盘 - CLOUD_TSSD：极速型SSD云硬盘 默认取值：当前有库存的硬盘类型。 示例值：CLOUD_HSSD
DiskId	String	否	系统盘ID。 该参数目前仅用于DescribeInstances等查询类接口的返回参数，不可用于RunInstances等写接口的入参。 示例值：disk-xyzw1234
DiskSize	Integer	否	系统盘大小，单位：GiB。默认值为 50 示例值：50
CdId	String	否	所属的独享集群ID。 注意：此字段可能返回 null，表示取不到有效值。 示例值：cdc-xxxxxx
DiskName	String	否	磁盘名称，长度不超过128 个字符。 示例值：name

Tag

标签。

被如下接口引用：CreateComputeEnv, CreateTaskTemplate, DescribeComputeEnv, DescribeComputeEnvCreateInfo, DescribeComputeEnvCreateInfos, DescribeComputeEnvs, DescribeJob, DescribeJobSubmitInfo, DescribeJobs, DescribeTaskTemplates, SubmitJob。

名称	类型	必选	描述
Key	String	是	标签键。 示例值：batch_tag
Value	String	是	标签值。 示例值：batch-demo

Task

任务

被如下接口引用：CreateTaskTemplate, DescribeJobSubmitInfo, DescribeTaskTemplates, ModifyTaskTemplate, SubmitJob。

名称	类型	必选	描述
Application	Application	是	应用程序信息
TaskName	String	否	任务名称，在一个作业内部唯一 示例值：test-task
TaskInstanceNum	Integer	否	任务实例运行个数，默认为1 示例值：1
ComputeEnv	AnonymousComputeEnv	否	运行环境信息，ComputeEnv 和 EnvId 必须指定一个（且只有一个）参数。
EnvId	String	否	计算环境ID，ComputeEnv 和 EnvId 必须指定一个（且只有一个）参数。 示例值：env_238swek2
RedirectInfo	RedirectInfo	否	重定向信息
RedirectLocalInfo	RedirectLocalInfo	否	重定向本地信息
InputMappings	Array of InputMapping	否	输入映射
OutputMappings	Array of OutputMapping	否	输出映射

名称	类型	必选	描述
OutputMappingConfigs	Array of OutputMappingConfig	否	输出映射配置
EnvVars	Array of EnvVar	否	自定义环境变量
Authentications	Array of Authentication	否	授权信息
FailedAction	String	否	TaskInstance失败后处理方式，取值包括 – TERMINATE：销毁计算实例（默认）、 – INTERRUPT：中断任务，保留计算实例、 – FAST_INTERRUPT：快速中断任务，保留计算实例。 示例值：TERMINATE
MaxRetryCount	Integer	否	任务失败后的最大重试次数，默认为0 示例值：0
Timeout	Integer	否	任务启动后的超时时间，单位秒，默认为86400秒 示例值：86400
MaxConcurrentNum	Integer	否	任务最大并发数限制，默认没有限制。 示例值：100
RestartComputeNode	Boolean	否	任务完成后，重启计算节点。适用于指定计算环境执行任务。 示例值：false
ResourceMaxRetryCount	Integer	否	启动任务过程中，创建计算资源如CVM失败后的最大重试次数，默认为0。最大值100。 计算资源创建重试的等待时间间隔策略设置如下： [1, 3]: 等待600 s发起重试； [4, 10]: 等待900 s发起重试； [11, 50]: 等待1800 s发起重试； [51, 100]: 等待3600 s发起重试； [a, b]表示重试次数区间，每次重试的等待时间随着重试次数的增加而递增。 例如，计算资源创建重试8次的耗时为： $3600 + 5900 = 6300$ s 示例值：1

TaskInstanceLog

任务实例日志详情。

被如下接口引用：DescribeTaskLogs。

名称	类型	描述
TaskInstanceIndex	Integer	任务实例 示例值：1
StdoutLog	String	标准输出日志（Base64编码，解码后最大日志长度2048字节） 示例值：MApWTV8zMI80X2NIbnRvcwo
StderrLog	String	标准错误日志（Base64编码，解码后最大日志长度2048字节） 示例值：MApWTV8zMI80X2NIbnRvcwo
StdoutRedirectPath	String	标准输出重定向路径 示例值：cos://batch-test-11762227.cos.ap-shanghai.myqcloud.com/batch_test/
StderrRedirectPath	String	标准错误重定向路径 示例值：cos://batch-test-11762227.cos.ap-shanghai.myqcloud.com/batch_test/
StdoutRedirectFileName	String	标准输出重定向文件名 示例值：stdout.log
StderrRedirectFileName	String	标准错误重定向文件名 示例值：stderr.log

TaskInstanceMetrics

任务实例统计指标

被如下接口引用：DescribeJob, DescribeTask。

名称	类型	描述
SubmittedCount	Integer	Submitted个数 示例值：1
PendingCount	Integer	Pending个数 示例值：1
RunnableCount	Integer	Runnable个数 示例值：1
StartingCount	Integer	Starting个数 示例值：1
RunningCount	Integer	Running个数 示例值：1
SucceedCount	Integer	Succeed个数 示例值：1
FailedInterruptedCount	Integer	FailedInterrupted个数 示例值：1
FailedCount	Integer	Failed个数 示例值：1

TaskInstanceView

任务实例视图信息

被如下接口引用：DescribeTask。

名称	类型	描述
TaskInstanceIndex	Integer	任务实例索引 示例值：1
TaskInstanceState	String	任务实例状态： - PENDING：等待中； - RUNNABLE：可运行； - STARTING：启动中； - RUNNING：运行中； - SUCCEED：成功； - FAILED：失败； - FAILED_INTERRUPTED：失败后保留实例。 示例值：RUNNING
ExitCode	Integer	应用程序执行结束的exit code 示例值：1
StateReason	String	任务实例状态原因，任务实例失败时，会记录失败原因 示例值：failed
ComputeNodeInstanceId	String	任务实例运行时所在计算节点（例如CVM）的InstanceId。任务实例未运行或者完结时，本字段为空。任务实例重试时，本字段会随之变化 示例值：ins-g9mp0ppm
CreateTime	String	创建时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 示例值：2018-02-07T09:29:09Z
LaunchTime	String	启动时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 示例值：2018-02-07T09:29:09Z
RunningTime	String	开始运行时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 示例值：2018-02-07T09:29:09Z

名称	类型	描述
EndTime	String	结束时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 示例值：2018-02-07T09:29:09Z
RedirectInfo	RedirectInfo	重定向信息
StateDetailedReason	String	任务实例状态原因详情，任务实例失败时，会记录失败原因 示例值：first heartbeat timeout

TaskMetrics

任务统计指标

被如下接口引用：DescribeJob, DescribeJobs。

名称	类型	描述
SubmittedCount	Integer	Submitted个数 示例值：1
PendingCount	Integer	Pending个数 示例值：1
RunnableCount	Integer	Runnable个数 示例值：1
StartingCount	Integer	Starting个数 示例值：1
RunningCount	Integer	Running个数 示例值：1
SucceedCount	Integer	Succeed个数 示例值：1
FailedInterruptedCount	Integer	FailedInterrupted个数 示例值：1
FailedCount	Integer	Failed个数 示例值：1

TaskTemplateView

任务模板信息

被如下接口引用：DescribeTaskTemplates。

名称	类型	描述
TaskTemplateId	String	任务模板ID 示例值：task-tmpl-0brmi1lm
TaskTemplateName	String	任务模板名称 示例值：testbatch
TaskTemplateDescription	String	任务模板描述 示例值：test batch is a example ,just for a demo.
TaskTemplateInfo	Task	任务模板信息
CreateTime	String	创建时间 示例值：2018-03-19 15:40:06
Tags	Array of Tag	任务模板绑定的标签列表。 注意：此字段可能返回 null，表示取不到有效值。

TaskView

任务视图信息

被如下接口引用：DescribeJob。

名称	类型	描述
TaskName	String	任务名称 示例值：sample_name
TaskState	String	任务状态： - PENDING：等待中； - RUNNABLE：可运行； - STARTING：启动中； - RUNNING：运行中； - SUCCEED：成功； - FAILED：失败； - FAILED_INTERRUPTED：失败后保留实例。 示例值：FAILED
CreateTime	String	开始时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 示例值：2018-02-07T09:29:09Z
EndTime	String	结束时间。按照ISO8601标准表示，并且使用UTC时间。格式为：YYYY-MM-DDThh:mm:ssZ。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2018-02-07T09:29:09Z

VirtualPrivateCloud

描述了VPC相关信息，包括子网，IP信息等

被如下接口引用：CreateComputeEnv, DescribeComputeEnvCreateInfo。

名称	类型	必选	描述
VpcId	String	是	私有网络ID，形如vpc-xxx。有效的VpcId可通过登录 控制台 查询；也可以调用接口 DescribeVpcs ，从接口返回中的VpcId字段获取。若在创建子机时VpcId与SubnetId同时传入DEFAULT，则强制使用默认vpc网络。 示例值：vpc-2ij
SubnetId	String	是	私有网络子网ID，形如subnet-xxx。有效的私有网络子网ID可通过登录 控制台 查询；也可以调用接口 DescribeSubnets ，从接口返回中的SubnetId字段获取。若在创建子机时SubnetId与VpcId同时传入DEFAULT，则强制使用默认vpc网络。 示例值：subnet-2ks
AsVpcGateway	Boolean	否	是否用作公网网关。公网网关只有在实例拥有公网IP以及处于私有网络下时才能正常使用。取值范围： • true：表示用作公网网关 • false：表示不作为公网网关 默认取值：false。 示例值：false
PrivateIpAddresses	Array of String	否	私有网络子网 IP 数组，在创建实例、修改实例vpc属性操作中可使用此参数。当前仅批量创建多台实例时支持传入相同子网的多个 IP。 示例值：["10.0.0.1"]
Ipv6AddressCount	Integer	否	为弹性网卡指定随机生成的 IPv6 地址数量。 示例值：1