# 阿里云
# 云安全中心（态势感知）

## API参考

文档版本：20190919

# 法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

# 通用约定

| 格式 | 说明 | 样例 |
| --- | --- | --- |
|  | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  **禁止:**<br>重置操作将丢失用户配置数据。 |
|  | 该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  **警告:**<br>重启操作将导致业务中断，恢复业务所需时间约10分钟。 |
|  | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。 |  **说明:**<br>您也可以通过按Ctrl + A选中全部文件。 |
| > | 多级菜单递进。 | 设置 > 网络 > 设置网络类型 |
| 粗体 | 表示按键、菜单、页面名称等UI元素。 | 单击 确定。 |
| `courier` 字体 | 命令。 | 执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。 |
| *## * | 表示参数、变量。 | `bae log list --instanceid` *Instance_ID* |
| []或者[a\|b] | 表示可选项，至多选择一个。 | `ipconfig` *[-all\|-t]* |
| {}或者{a\|b} | 表示必选项，至多选择一个。 | `swich` *{stand \| slave}* |

# 目录

法律声明.....I
通用约定.....I
1 调用方式..... 1
2 公共参数..... 4
3 日志分析API.....6
3.1 概览.....6
3.2 日志分析Logstore和Project.....7
3.3 服务入口.....7
3.4 访问秘钥.....10
3.5 公共请求头.....10
3.6 公共响应头.....13
3.7 请求签名.....14
3.8 通用错误码.....19
3.9 日志项目接口.....21
3.9.1 CreateProject.....21
3.9.2 DeleteProject.....23
3.9.3 UpdateProject.....25
3.9.4 GetProject.....26
3.9.5 ListProject.....28
3.9.6 GetProjectLogs.....31
3.10 Logtail机器组相关接口.....34
3.10.1 CreateMachineGroup.....34
3.10.2 DeleteMachineGroup.....36
3.10.3 UpdateMachineGroup.....38
3.10.4 ListMachineGroup.....41
3.10.5 GetMachineGroup.....43
3.10.6 ApplyConfigToMachineGroup.....45
3.10.7 RemoveConfigFromMachineGroup.....47
3.10.8 ListMachines.....49
3.10.9 GetAppliedConfigs.....51
3.11 Logtail配置相关接口.....53
3.11.1 CreateConfig.....53
3.11.2 ListConfig.....56
3.11.3 GetAppliedMachineGroups.....58
3.11.4 GetConfig.....59
3.11.5 DeleteConfig.....62
3.11.6 UpdateConfig.....63
3.12 RAM/STS.....66
3.12.1 概览.....66
3.12.2 资源列表.....68

3.12.3 动作列表.....70
3.12.4 鉴权规则.....72
4 获取告警数据.....76
4.1 DescribeAlarmEventList.....76
4.2 DescribeAlarmEventDetail.....80
4.3 DescribeSuspEvents.....85
4.4 DescribeSuspEventDetail.....90
5 云产品配置检查.....96
5.1 DescribeRiskItemType.....96
5.2 StartBaselineSecurityCheck.....98
5.3 DescribeRiskCheckSummary.....99
5.4 ModifyRiskCheckStatus.....104
5.5 DescribeRiskCheckResult.....106
5.6 DescribeSecurityCheckScheduleConfig.....110
5.7 ModifyRiskSingleResultStatus.....112
6 漏洞管理.....114
6.1 DescribeVulList.....114
6.2 DescribeEmgVulGroup.....126
6.3 DescribeVulWhitelist.....129
6.4 DescribeConcernNecessity.....131
6.5 DescribeGroupedVul.....133
6.6 ModifyCreateVulWhitelist.....136
6.7 DescribeAutoDelConfig.....137
6.8 ModifyOperateVul.....138
7 基线.....141
7.1 DescribeCheckWarningSummary.....141
7.2 DescribeStratety.....152
7.3 DescribeStrategyExecDetail.....156
7.4 DescribeCheckWarnings.....158
7.5 DescribeCheckWarningDetail.....166
7.6 DescribeWarningMachines.....168
8 资产管理.....173
8.1 DescribeFieldStatistics.....173
8.2 DescribeGroupedTags.....175
8.3 DescribeAllGroups.....177
8.4 DeleteGroup.....178
8.5 CreateOrUpdateAssetGroup.....180
8.6 ModifyTagWithUuid.....181
8.7 DescribeInstanceStatistics.....182
8.8 DescribeCloudProductFieldStatistics.....185
8.9 DescribeDomainCount.....187
8.10 DescribeDomainList.....188
8.11 DescribeDomainDetail.....191

9 资产指纹.....195
9.1 DescribePropertyCount..... 195
9.2 DescribePropertyPortDetail..... 196
9.3 DescribePropertyProcDetail.....201
9.4 DescribePropertyPortItem..... 204
9.5 DescribePropertyProcItem.....207
9.6 DescribePropertySoftwareDetail..... 210
9.7 DescribePropertySoftwareItem..... 213
9.8 DescribePropertyUserDetail.....215
9.9 DescribePropertyUserItem..... 219

# 1 调用方式

您可以通过发送HTTP GET请求调用云安全中心API，并按照接口说明在请求中加入相应的请求参数。调用接口后系统会返回处理结果。请求和返回结果都使用UTF-8字符集进行编码。

## 请求结构

云安全中心的API是PRC风格，您可以通过发送HTTP GET请求调用云安全中心API。

其请求结构如下：

```
https://Endpoint/?Action=xx&Parameters
```

其中：

- Endpoint：云安全中心API的服务接入地址为tds.aliyuncs.com。
- Action：要执行的操作，如使用DescribeAlarmEventList查询安全事件列表。
- Version：要使用的API版本，云安全中心的API版本是2018-12-03。
- Parameters：请求参数，每个参数之间用“&”分隔。

  请求参数由公共请求参数和API自定义参数组成。公共参数中包含API版本号、身份验证等信息，详细内容参见公共参数。

下面是一个调用DescribeAlarmEventList接口查询安全事件列表的示例：

说明：

为了便于用户查看，本文档中的示例都做了格式化处理。

```
http(s)://tds.aliyuncs.com/?Action=DescribeAlarmEventList
&Format=xml
&Version=2018-12-03
&Signature=xxxx%xxxx%3D
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&AccessKeyId=key-test
&TimeStamp=2012-06-01T12:00:00Z
...
```

## API授权

为了确保您的账号安全，建议您使用子账号的身份凭证调用API。如果您使用RAM账号调用云安全中心API，您需要为该RAM账号创建、附加相应的授权策略。

## API签名

云安全中心服务会对每个API请求进行身份验证，无论使用HTTP还是HTTPS协议提交请求，都需要在请求中包含签名（Signature）信息。

签名计算过程参见#unique_5。

云安全中心通过使用AccessKey ID和AccessKey Secret进行对称加密的方法来验证请求的发送者身份。AccessKey是为阿里云账号和RAM用户发布的一种身份凭证（类似于用户的登录密码），其中AccessKey ID 用于标识访问者的身份，AccessKey Secret是用于加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密。

RPC API需按如下格式在请求中增加签名（Signature）：

```
https://endpoint/?SignatureVersion=1.0&SignatureMethod=HMAC-SHA1&
Signature=CT9X0VtwR86fNWSnsc6v8YGOjuE%3D&SignatureNonce=3ee8c1b8-83d3-
44af-a94f-4e0ad82fd6cf
```

以`DescribeAlarmEventList`为例，假设AccessKey ID是`testid`， AccessKey Secret是`testsecret`，则签名前的请求URL如下：

```
https://tds.aliyuncs.com/?Action=DescribeAlarmEventList
&TimeStamp=2016-02-23T12:46:24Z
&Format=XML
&AccessKeyId=testid
&SignatureMethod=HMAC-SHA1
&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
&Version=2018-01-17
&SignatureVersion=1.0
```

完成以下步骤计算签名：

1. 使用请求参数创建待签名字符串：

   ```
   GET&%2F&AccessKeyId%3Dtestid&Action%3DDescribeAlarmEventList&Format
   %3DXML&SignatureMethod%3DHMAC-SHA1&SignatureNonce%3D3ee8c1b8-83d3-
   44af-a94f-4e0ad82fd6cf&SignatureVersion%3D1.0&TimeStamp%3D2016-02-
   23T12%253A46%253A24Z&Version%3D2018-12-03
   ```

2. 计算待签名的HMAC的值。

   在AccessKey Secret后添加一个“&”作为计算HMAC值的key。本示例中的key为`testsecret&`。

   ```
   CT9X0VtwR86fNWSnsc6v8YGOjuE=
   ```

3. 将签名加到请求参数中：

   ```
   https://tds.aliyuncs.com/?Action=DescribeAlarmEventList
   &TimeStamp=2016-02-23T12:46:24Z
   &Format=XML
   &AccessKeyId=testid
   ```

```
&SignatureMethod=HMAC-SHA1
&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
&Version=2018-12-03
&SignatureVersion=1.0
&Signature=CT9X0VtwR86fNWSnsc6v8YGOjuE%3D
```

# 2 公共参数

## 公共请求参数

公共请求参数是每个接口都需要使用到的请求参数。

表 2-1: 公共请求参数表

| 名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| Region | string | 是 | 云安全中心所实例在的地域。<br>取值：<br>· cn-hangzhou：表示除马来西亚以外的所有地区，包括中国。<br>· ap-southeast-3：表示马来西亚地区。 |
| Format | string | 否 | 返回消息的格式。<br>取值：JSON （默认值） \| XML |
| Version | String | 是 | API版本号，使用YYYY-MM-DD日期格式。<br>取值：2018-12-03 |
| AccessKeyId | String | 是 | 访问服务使用的密钥ID。 |
| Signature | String | 是 | 签名结果串。 |
| SignatureMethod | String | 是 | 签名方式。<br>取值：HMAC-SHA1 |
| Timestamp | String | 是 | 请求的时间戳，为日期格式。使用UTC时间按照ISO8601标，格式为YYYY-MM-DDThh:mm:ssZ。<br>例如，北京时间2013年1月10日20点0分0秒，表示为2013-01-10T12:00:00Z。 |
| SignatureVersion | String | 是 | 签名算法版本，目前版本是1.0。 |
| SignatureNonce | String | 是 | 唯一随机数，用于防止网络重放攻击。<br>在不同请求间要使用不同的随机数值。 |

| 名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| ResourceOwnerAccount | String | 否 | 本次API请求访问到的资源拥有者账户，即登录用户名。 |

**示例**

```
https://tds.aliyuncs.com/?Action=DescribeDomainNames
&Timestamp=2014-05-19T10%3A33%3A56Z
&Format=xml
&AccessKeyId=testid
&SignatureMethod=Hmac-SHA1
&SignatureNonce=NwDAxvLU6tFE0DVb
&Version=2018-12-03
&SignatureVersion=1.0
&Signature=Signature
```

## 公共返回参数

API返回结果采用统一格式，返回2xx HTTP状态码代表调用成功；返回4xx或5xx HTTP状态码代表调用失败。

调用成功返回的数据格式有XML和JSON两种，可以在发送请求时指定返回的数据格式，默认为XML格式。

每次接口调用，无论成功与否，系统都会返回一个唯一识别码RequestId。

- XML格式

```
<?xml version="1.0" encoding="utf-8"?>
    <!--结果的根结点-->
    <接口名称+Response>
        <!--返回请求标签-->
        <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
        <!--返回结果数据-->
    </接口名称+Response>
```

- JSON格式

```
{
    "RequestId":"4C467B38-3910-447D-87BC-AC049166F216",
    /*返回结果数据*/
    }
```

# 3 日志分析API

## 3.1 概览

日志服务（Log Service，简称 LOG）是针对日志平台化服务。服务提供各种类型日志的实时收集、存储和分发。除此之外，LOG 有 ODPS Table 间同步服务，通过 LOG 可以将日志投递至 ODPS 做大数据分析。

除了通过管理控制台进行操作外，LOG 还提供了 API（Application Programming Interface）方式写入、查询日志数据，管理自己的项目及日志库等。目前开放如下 API：

| 对象 | 方法 |
| --- | --- |
| Log（日志） | 日志、日志组表示等基本概念 |
| Project （项目） | List、Create、 Delete、Get、 |
| | #unique_14（统计Project下所有日志） |
| Config （配置） | List、Create、 Delete、Get、Update |
| | GetAppliedMachineGroups（查询应用到的机器组） |
| MachineGroup （机器组） | List、Create、Delete、Get、Update |
| | Apply/Remove（应用/删除配置） |
| | GetAppliedConfigs（查询已应用配置列表） |
| LogStore （日志库） | List、Create、 Delete、Get、Update |
| | GetLogs（查询日志）、GetHistograms（查询日志分布） |
| Index（索引） | Create、Update、Delete、#unique_42 |
| Shard (分区) | List、Split、Merge |
| | PostLogStoreLogs（写入日志） |
| | GetCursor（定位日志位置） |
| | PullLogs（消费日志） |
| Shipper（日志投递规则） | GetShipperStatus（查询日志投递任务状态） |
| | RetryShipperTask（重试失败投递任务） |
| ConsumerGroup（消费组） | Create、Update、Delete、List |

| 对象 | 方法 |
| --- | --- |
| | HeartBeat（发送心跳）、GetCheckpoint, UpdateCheckpoint |

通过 API 可以操作下列服务：

- 根据#unique_15、#unique_22 信息收集日志。
- 创建 #unique_31、向日志库写入、读取日志。
- 对不同用户进行 访问控制。

说明：

- API 目前提供 Rest 风格。
- 为使用 API，需要知道 #unique_60。
- API 所有请求都需要做安全验证，请参考#unique_61解释了具体的 API 请求签名机制及流程。
- Log Service 支持 RAM、STS，RAM 子用户使用 API，和一般云账号没有区别，使用子用户的 AK 签名即可。STS 临时身份除了临时 AK 外，还需要填写一个特殊的 HTTP header，详见#unique_62，这个 HTTP header 需要参与签名，详见 #unique_61。

## 3.2 日志分析Logstore和Project

云安全中心全量日志集中存放在`sas-log`专属日志库（Logstore）中。您可以在储存日志服务的项目（Project）`sas-log-阿里云账户ID-区域名`中找到专属日志库。

## 3.3 服务入口

不同网络入口可以参考 使用 Logtail 收集各网络日志数据。

公网服务入口

日志服务入口是访问一个项目（Project）及其内部日志数据的 URL。它和 Project 所在的阿里云区域（Region）及 Project 名称相关。目前，日志服务已经在多个阿里云 Region 下开通，在各 Region 内的公网服务入口如下：

| 地域 | 服务入口 |
| --- | --- |
| 华东 1 （杭州） | cn-hangzhou.log.aliyuncs.com |
| 华东 1 （杭州-金融云） | cn-hangzhou-finance.log.aliyuncs.com |
| 华东 2 （上海） | cn-shanghai.log.aliyuncs.com |

| 地域 | 服务入口 |
| --- | --- |
| 华东 2 （上海-金融云） | cn-shanghai-finance-1.log.aliyuncs.com |
| 华北 1 （青岛） | cn-qingdao.log.aliyuncs.com |
| 华北 2 （北京） | cn-beijing.log.aliyuncs.com |
| 华北 3 （张家口） | cn-zhangjiakou.log.aliyuncs.com |
| 华北 5 （呼和浩特） | cn-huhehaote.log.aliyuncs.com |
| 华南 1 （深圳） | cn-shenzhen.log.aliyuncs.com |
| 华南 1 （深圳-金融云） | cn-shenzhen-finance.log.aliyuncs.com |
| 西南 1 （成都） | cn-chengdu.log.aliyuncs.com |
| 香港 | cn-hongkong.log.aliyuncs.com |
| 亚太东北 1 （东京） | ap-northeast-1.log.aliyuncs.com |
| 亚太东南 1 （新加坡） | ap-southeast-1.log.aliyuncs.com |
| 亚太东南 2 （悉尼） | ap-southeast-2.log.aliyuncs.com |
| 亚太东南 3 （吉隆坡） | ap-southeast-3.log.aliyuncs.com |
| 亚太东南 5 （雅加达） | ap-southeast-5.log.aliyuncs.com |
| 中东东部 1 （迪拜） | me-east-1.log.aliyuncs.com |
| 美国西部 1 （硅谷） | us-west-1.log.aliyuncs.com |
| 欧洲中部 1 （法兰克福） | eu-central-1.log.aliyuncs.com |
| 美国东部 1 （弗吉尼亚） | us-east-1.log.aliyuncs.com |
| 亚太南部 1（孟买） | ap-south-1.log.aliyuncs.com |
| 英国（伦敦） | eu-west-1.log.aliyuncs.com |

当访问某个具体的 Project 时，需要根据 Project 名称及其所在 Region 组合出最终访问地址。具体格式如下：

```
<project_name>.<region_endpoint>
```

例如，Project 名为 big-game，所在区域为“华东 1 （杭州）”，则对应访问地址如下：

```
big-game.cn-hangzhou.log.aliyuncs.com
```

 说明：

在创建日志服务项目时需要指定某个 Region。一旦在创建时指定了 Region，该设置就不可以更改，且无法跨区域迁移项目。创建 Project 之后，必须选择与其所在区域相匹配的根服务入口地址来组成该 Project 访问地址，用做 API 请求的服务入口。

经典/VPC网络服务入口

如果在阿里云 ECS 机器（包含VPC）环境使用日志服务 API，还可以使用内网服务入口（使用内网服务入口访问日志服务不消耗 ECS 公网流量，可以节约宝贵的 ECS 公网带宽），各个 Region 服务入口如下：

| 地域 | 根服务入口 |
| --- | --- |
| 华东 1 （杭州） | cn-hangzhou-intranet.log.aliyuncs.com |
| 华东 1 （杭州-金融云） | cn-hangzhou-finance-intranet.log.aliyuncs.com |
| 华东 2 （上海） | cn-shanghai-intranet.log.aliyuncs.com |
| 华东 2 （上海-金融云） | cn-shanghai-finance-1-intranet.log.aliyuncs.com |
| 华北 1 （青岛） | cn-qingdao-intranet.log.aliyuncs.com |
| 华北 2 （北京） | cn-beijing-intranet.log.aliyuncs.com |
| 华南 1 （深圳） | cn-shenzhen-intranet.log.aliyuncs.com |
| 华南 1 （深圳-金融云） | cn-shenzhen-finance-intranet.log.aliyuncs.com |
| 华北 3 （张家口） | cn-zhangjiakou-intranet.log.aliyuncs.com |
| 华北 5 （呼和浩特） | cn-huhehaote-intranet.log.aliyuncs.com |
| 西南 1 （成都） | cn-chengdu-intranet.log.aliyuncs.com |
| 香港 | cn-hongkong-intranet.log.aliyuncs.com |
| 美国西部 1 （硅谷） | us-west-1-intranet.log.aliyuncs.com |
| 亚太东北 1 （东京） | ap-northeast-1-intranet.log.aliyuncs.com |
| 亚太东南 1 （新加坡） | ap-southeast-1-intranet.log.aliyuncs.com |
| 亚太东南 2 （悉尼） | ap-southeast-2-intranet.log.aliyuncs.com |
| 亚太东南 3 （吉隆坡） | ap-southeast-3-intranet.log.aliyuncs.com |
| 亚太东南 5 （雅加达） | ap-southeast-5-intranet.log.aliyuncs.com |
| 中东东部 1 （迪拜） | me-east-1-intranet.log.aliyuncs.com |
| 欧洲中部 1 （法兰克福） | eu-central-1-intranet.log.aliyuncs.com |

| 地域 | 根服务入口 |
| --- | --- |
| 美国东部 1 （弗吉尼亚） | us-east-1-intranet.log.aliyuncs.com |
| 亚太南部 1（孟买） | ap-south-1-intranet.log.aliyuncs.com |
| 英国（伦敦） | eu-west-1-intranet.log.aliyuncs.com |

如上例，其内网访问地址如下：

```
big-game.cn-hangzhou-intranet.log.aliyuncs.com
```

说明:

目前，日志服务 API 服务在如上服务入口上仅支持 HTTP/HTTPS 协议。

全球加速服务入口

日志服务在VPC和公网基础上，新增全球加速公网的网络类型。相较于普通的公网访问，全球加速公网在延时和稳定性上具备显著优势，适用于对数据采集、消费延时、可靠性要求较高的场景。

全球加速的服务入口所有Region全部一致，服务入口为：`log-global.aliyuncs.com`

说明:

全球加速功能默认为关闭状态，需要手动开启后才可使用。开启全球加速，请参考#unique_66。

## 3.4 访问秘钥

阿里云访问秘钥是阿里云为用户使用 API（非控制台）来访问其云资源设计的“安全口令”。您可以用它来签名 API 请求内容以通过服务端的安全验证。

该访问秘钥成对（AccessKeyId 与 AccessKeySecret）生成和使用。每个阿里云用户可以创建多对访问秘钥，且可随时启用（Active）、禁用（Inactive）或者删除已经生成的访问秘钥对。

您可以通过阿里云控制台的秘钥管理页面创建、管理所有的访问秘钥对。由于访问秘钥是阿里云对 API 请求进行安全验证的关键因子，请妥善保管你的访问秘钥。如果某些秘钥对出现泄漏风险，建议及时删除该秘钥对并生成新的替代秘钥对。

## 3.5 公共请求头

Log Service API 是基于 HTTP 协议的 Rest 风格接口。它支持一组可以在所有 API 请求中使用的公共请求头（除特别说明，每个 Log Service API 请求都必须提供这些公共请求头），其详细定义如下：

| Header 名称 | 类型 | 说明 |
| --- | --- | --- |
| Accept | 字符串 | 客户端希望服务端返回的类型，目前支持 application/json、application/x-protobuf 两种，该字段为非必选参数，仅对 GET 请求有效。具体取值以各个接口定义为准。 |
| Accept-Encoding | 字符串 | 客户端希望服务端返回的压缩算法，目前支持 lz4、 deflate 或空（不压缩）。该字段为非必选参数，仅对 GET 类请求有效。具体取值以各个接口定义为准。 |
| Authorization | 字符串 | 签名内容，更多细节请参考 #unique_61。 |
| Content-Length | 数值 | RFC 2616 中定义的 HTTP 请求 Body 长度。如果请求无 Body 部分，则不需要提供该请求头。 |
| Content-MD5 | 字符串 | 请求 Body 经过 MD5 计算后的字符串，计算结果为大写。如果没有 Body 部分，则不需要提供该请求头。 |
| Content-Type | 字符串 | RFC 2616 中定义的 HTTP 请求 Body 类型。目前 Log Service API 请求只支持 application/x-protobuf。如果没有 Body 部分，则不需要提供该请求头。具体取值以各个接口定义为准。 |
| Date | 字符串 | 当前发送时刻的时间，参数目前只支持 RFC 822 格式，使用 GMT 标准时间。格式化字符串如下：%a, %d %b %Y %H:%M:%S GMT (如：Mon, 3 Jan 2010 08:33:47 GMT)。 |

| Header 名称 | 类型 | 说明 |
| --- | --- | --- |
| Host | 字符串 | HTTP 请求的完整 HOST 名字（不包括如 `http://` 这样的协议头）。例如，`big-game.cn-hangzhou.sls.aliyuncs.com`。 |
| x-log-apiversion | 字符串 | API 的版本号，当前版本为 0.6.0。 |
| x-log-bodyrawsize | 数值 | 请求的 Body 原始大小。当无 Body 时，该字段为 0；当 Body 是压缩数据，则为压缩前的原始数据大小。该域取值范围为 0~3x1024x1024。该字段为非必选字段，只在压缩时需要。 |
| x-log-compresstype | 字符串 | API 请求中 Body 部分使用的压缩方式。目前支持 lz4 压缩类型和 deflate 压缩类型（RFC 1951，使用 zlib 格式，参考 RFC 1950）。如果不压缩可以不提供该请求头。 |
| x-log-date | 字符串 | 当前发送时刻的时间，格式和 Date 头一致。该请求头为可选项。如果请求中包含该公共请求头，它的值会取代 Date 标准头的值用于服务端请求验证（该字段不参与签名）。无论是否有 x-log-date 头，HTTP 标准 Date 头都必须提供。 |
| x-log-signaturemethod | 字符串 | 签名计算方式，目前仅支持 `hmac-sha1`。 |
| x-acs-security-token | 字符串 | 使用 STS 临时身份发送数据。当使用 STS 临时身份时必填，其他情况不要填写。 |

- 请求中 Date 所表示的时间与服务器接收到该请求的时间最大可接受误差为 15 分钟，如果超过 15 分钟服务器端会拒绝该请求。如果请求中设置了 x-log-date 头部，则该时间误差计算基于 x-log-date 头的值。

- 如果请求指明了压缩算法（在 x-log-compresstype 中指定），则需要把原始数据压缩后放到 HTTP Body 部分，而对应的 Content-Length、Content-MD5 头部也是按照压缩后的 Body 部分计算。
- 由于某些平台上发送 HTTP 请求时无法指定 Date 头（由平台自身的库内部自动指定为发送当前时间），造成无法使用正确的 Date 值计算请求签名。在这种情况下，请指定 x-log-date 头并用该请求头的值参与请求签名计算。Log Service 服务端在接收到 API 请求后会首先判断是否有 x-log-date 头。如果有，则用它的值来做签名验证，否则就用 HTTP 的标准头 Date 做签名验证。

## 3.6 公共响应头

Log Service API 是基于 HTTP 协议的 Rest 风格接口。所有的 Log Service API 响应都提供一组公共响应头，其详细定义如下：

| Header 名称 | 类型 | 说明 |
| --- | --- | --- |
| Content-Length | 数值 | RFC 2616 中定义的 HTTP 响应内容长度。 |
| Content-MD5 | 字符串 | RFC 2616 中定义的 HTTP 响应内容的 MD5 值。Body 经过 MD5 计算后的字符串，为大写字符串。 |
| Content-Type | 字符串 | RFC 2616 中定义的 HTTP 响应内容类型。目前 Log Service 服务端响应类型支持 application/json, application/x-protobuf 两种类型。 |
| Date | 字符串 | 当前返回时刻的时间，参数目前只支持 RFC 822 格式，使用 GMT 标准时间。格式化字符串如下：%a, %d %b %Y %H:%M:%S GMT，如：Mon, 3 Jan 2010 08:33:47 GMT。 |
| x-log-requestid | 字符串 | 服务端产生的标示，该请求的唯一ID。该响应头与具体应用无关，主要用于跟踪和调查问题。如果用户希望调查出现问题的 API 请求，可以通过工单向日志服务团队提供该 ID。 |

## 3.7 请求签名

为保证用户日志数据的安全，Log Service API 的所有 HTTP 请求都必须经过安全验证。目前，该安全验证基于阿里云的 #unique_71，使用对称加密算法完成的。

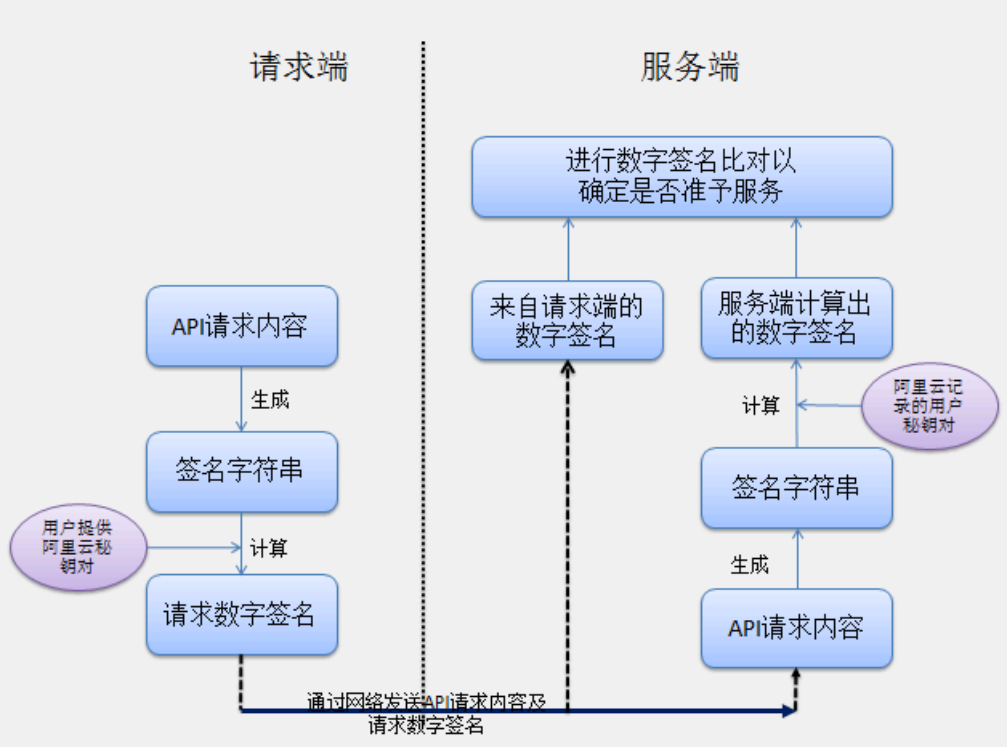
其工作流程如下：

1. 请求端根据 API 请求内容（包括 HTTP Header 和 Body）生成签名字符串。
2. 请求端使用阿里云的访问秘钥对（AccessKeyID 和 AccessKeySecret）对第一步生成的签名字符串进行签名，形成该 API 请求的数字签名。
3. 请求端把 API 请求内容和数字签名一同发送给服务端。
4. 服务端在接到请求后会重复如上的第一、二步工作，并在服务端计算出该请求期望的数字签名。

   **说明：**
   服务端会在后台取得该请求使用的用户访问秘钥对。
5. 服务端用期望的数字签名和请求端发送过来的数字签名做比对，如果完全一致则认为该请求通过安全验证。否则直接拒绝该请求。

上面的整个流程也可以使用下图直观描述：

图 3-1: 安全验证流程




上面的安全验证流程可以达到如下目的：

- 确认哪位用户在做 API 请求。因为在发送请求前需要用户指定生成数字签名的秘钥对，在服务端即可通过该秘钥对确定用户身份，进而可做访问权限管理。
- 确认用户请求在网络传输过程中有无被篡改。因为服务端会对接收到的请求内容重新计算数字签名，一旦请求内容在网络上被篡改，则无法通过数字签名比对。

### 签名 API 请求

为了通过 API 请求的安全验证，用户需要在客户端对其 API 请求进行签名（即生成正确的数字签名），并且使用 HTTP 头 Authorization 在网络上传输该请求的数字签名。Authorization 头的具体格式如下：

```
Authorization:LOG <AccessKeyId>:<Signature>
```

如上格式所示，Authorization 头的值包含用户访问秘钥对中的 AccessKeyId，且与之对应的 AccessKeySecret 将用于 Signature 值的构造。下面将详细解释如何构造该 Signature 值。

**第一步：准备合适的阿里云访问秘钥**

如上所述，给 API 请求生成签名，需使用一对访问秘钥（AccessKeyId/AccessKeySecret)。您可以使用已经存在的访问秘钥对，也可以创建新的访问秘钥对，但需要保证使用的秘钥对处在“启用”状态。

**第二步：生成请求的签名字符串**

Log Service API 的签名字符串由 HTTP 请求中的 Method，Header 和 Body 信息一同生成，具体方式如下：

```
SignString = VERB + "\n"
             + CONTENT-MD5 + "\n"
             + CONTENT-TYPE + "\n"
             + DATE + "\n"
             + CanonicalizedLOGHeaders + "\n"
             + CanonicalizedResource
```

上面公式中的 `\n` 表示换行转义字符，`+`（加号）表示字符串连接操作，其他各个部分定义如下所示。

**表 3-1: 签名字符串定义**

| 名称 | 定义 | 示例 |
| --- | --- | --- |
| VERB | HTTP 请求的方法名称 | PUT、GET、POST 等 |
| CONTENT-MD5 | HTTP 请求中 Body 部分的 MD5 值（必须为大写字符串） | 875264590688CA6171F6228AF5BBB3D2 |
| CONTENT-TYPE | HTTP 请求中 Body 部分的类型 | application/x-protobuf |

| 名称 | 定义 | 示例 |
| --- | --- | --- |
| DATE | HTTP 请求中的标准时间戳（遵循 RFC 1123 格式，使用 GMT 标准时间） | Mon, 3 Jan 2010 08:33:47 GMT |
| CanonicalizedLOGHeaders | 由 HTTP 请求中以 `x-log` 和 `x-acs` 为前缀的自定义头构造的字符串（具体构造方法见下面详述） | x-log-apiversion:0.6.0\nx-log-bodyrawsize:50\nx-log-signaturemethod:hmac-sha1 |
| CanonicalizedResource | 由 HTTP 请求资源构造的字符串（具体构造方法见下面详述） | /logstores/app_log |

对于部分无 Body 的 HTTP 请求，其 CONTENT-MD5 和 CONTENT-TYPE 两个域为空字符串，这时整个签名字符串的生成方式如下：

```
SignString = VERB + "\n"
             + "\n"
             + "\n"
             + DATE + "\n"
             + CanonicalizedLOGHeaders + "\n"
             + CanonicalizedResource
```

正如 #unique_62 中描述，Log Service API 中引入了一个自定义请求头 `x-log-date`。如果您在请求中指定了该请求头，则其值会替代 HTTP 标准请求头 Date 加入签名计算。

CanonicalizedLOGHeaders 的构造方式如下：

1. 将所有以 `x-log` 和 `x-acs` 为前缀的 HTTP 请求头的名字转换成小写字母。
2. 将上一步得到的所有 LOG 自定义请求头按照字典序进行升序排序。
3. 删除请求头和内容之间分隔符两端出现的任何空格。
4. 将所有的头和内容用 `\n` 分隔符组合成最后的 CanonicalizedLOGHeader。

CanonicalizedResource 的构造方式如下：

1. 将 CanonicalizedResource 设置为空字符串（""）。
2. 放入要访问的 LOG 资源，如 */logstores/logstorename*（无 `logstorename` 则不填）。
3. 如请求包含查询字符串（`QUERY_STRING`），则在 CanonicalizedResource 字符串尾部添加 `?` 和查询字符串。

其中，QUERY_STRING 是 URL 中请求参数按字典序排序后的字符串，其中参数名和值之间用 = 相隔组成字符串，并对参数名-值对按照字典序升序排序，然后以 & 符号连接构成字符串。其公式化描述如下：

```
QUERY_STRING = "KEY1=VALUE1" + "&" + "KEY2=VALUE2"
```

第三步：生成请求的数字签名

目前，Log Service API 只支持一种数字签名算法，即默认签名算法 hmac-sha1。其完整签名公式如下：

```
Signature = base64(hmac-sha1(UTF8-Encoding-Of(SignString), AccessKeyS
ecret))
```

签名的方法用 RFC 2104 中定义的 HMAC-SHA1 方法。如上公式用的 AccessKeySecret 必须和最终的 Authorization 头中使用的 AccessKeyId 相对应。否则，请求将无法通过服务端验证。

在计算出数字签名后，使用该值按本节最前面描述的 Authorization 头格式构建完整的 Log Service API 请求安全验证头，并填入 HTTP 请求中即可发送。

请求签名过程示例

为更好地理解整个请求签名的流程，我们用两个示例来演示整个过程。首先，假设您用做 Log Service API 签名的访问秘钥对如下：

```
AccessKeyId = "bq2sjzesjmo86kq*********"
AccessKeySecret = "4fdO2fTDDnZPU/L7CHNd********"
```

示例一：

您需要发送如下 GET 请求列出 ali-test-project 项目下的所有 Logstores，其 HTTP 请求如下：

```
GET /logstores HTTP 1.1
Mon, 09 Nov 2015 06:11:16 GMT
Host: ali-test-project.regionid.example.com
x-log-apiversion: 0.6.0
```

```
x-log-signaturemethod: hmac-sha1
```

如上 Log Service API 请求生成的签名字符串为：

```
GET\n\n\nMon, 09 Nov 2015 06:11:16 GMT\nx-log-apiversion:0.6.0\nx-log-
signaturemethod:hmac-sha1\n/logstores?logstoreName=&offset=0&size=1000
```

由于是 GET 请求，该请求无任何 HTTP Body，所以生成的签名字符串中 CONTENT-TYPE 与 CONTENT-MD5 域为空字符串。如果以前面指定的 AccessKeySecret 做签名运算后得到的签名为：

```
jEYOTCJs2e88o+y5F4/S5IsnBJQ=
```

最后发送经数字签名的 HTTP 请求内容如下：

```
GET /logstores HTTP 1.1
Mon, 09 Nov 2015 06:11:16 GMT
Host: ali-test-project.regionid.example.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Authorization: LOG bq2sjzesjmo86kq35behupbq:jEYOTCJs2e88o+y5F4/
S5IsnBJQ=
```

示例二：

您需要给同上例 ali-test-project 项目中名为 test-logstore 的 Logstore 写入下面的日志：

```
topic=""
time=1447048976
source="10.10.10.1"
"TestKey": "TestContent"
```

为此，按照 Log Service API 定义需要构建如下 HTTP 请求：

```
POST /logstores/test-logstore HTTP/1.1
Date: Mon, 09 Nov 2015 06:03:03 GMT
Host: test-project.regionid.example.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Content-MD5: 1DD45FA4A70A9300CC9FE7305AF2C494
Content-Length: 52
x-log-apiversion:0.6.0
x-log-bodyrawsize:50
x-log-compresstype:lz4
x-log-signaturemethod:hmac-sha1
<日志内容序列化成 ProtoBuffer 格式的字节流>
```

在这个 HTTP 请求中，写入的日志内容首先被序列化成 ProtoBuffer 格式（请参考ProtoBuffer格式了解该格式的更多细节）后作为请求 Body。所以该请求的 Content-Type 头

的值指定为 application/x-protobuf。类似，Content-MD5 头的值是请求 body 对应的 MD5 值。按照上面的签名字符串构造方式，这个请求对应的签名字符串为：

```
POST\n1DD45FA4A70A9300CC9FE7305AF2C494\napplication/x-protobuf\nMon,
09 Nov 2015 06:03:03 GMT\nx-log-apiversion:0.6.0\nx-log-bodyrawsize:50
\nx-log-compresstype:lz4\nx-log-signaturemethod:hmac-sha1\n/logstores/
test-logstore
```

同样，以前面示例中的 AccessKeySecret 做签名运算，得到的最终签名为：

```
XWLGYHGg2F2hcfxWxMLiNkGki6g=
```

最后发送经数字签名的 HTTP 请求内容如下：

```
POST /logstores/test-logstore HTTP/1.1
Date: Mon, 09 Nov 2015 06:03:03 GMT
Host: test-project.regionid.example.com
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
Content-MD5: 1DD45FA4A70A9300CC9FE7305AF2C494
Content-Length: 52
x-log-apiversion:0.6.0
x-log-bodyrawsize:50
x-log-compresstype:lz4
x-log-signaturemethod:hmac-sha1
Authorization: LOG bq2sjzesjmo86kq35behupbq:XWLGYHGg2F2hcfxWxMLi
NkGki6g=
<日志内容序列化成 ProtoBuffer 格式的字节流>
```

## 3.8 通用错误码

当 API 请求发生错误的时候，服务端会返回错误信息，包括 HTTP 的 Status Code 和响应 Body 中的具体错误细节。其中响应 Body 中的错误细节为如下格式：

```
{
"errorCode" : <ErrorCode>,
"errorMessage" : <ErrorMessage>
}
```

在所有服务端可能返回的错误信息中，一部分适用于多数 API，而另外一部分则为某些 API所独有。下表即为 API 响应中的通用错误码，它们会在多个 API 响应中出现。而每个 API 所独有的错误码会在该 API 参考中单独描述。

表 3-2: 通用错误码

| HTTP 状态码（Status Code） | 错误码（Error Code） | 错误消息（Error Message） | 描述（Description） |
| --- | --- | --- | --- |
| 411 | MissingContentLength | Content-Length does not exist in http header when it is necessary. | 没有提供必须的 Content-Length 请求头。 |
| 415 | InvalidContentType | Content-Type {type} is unsupported. | 不支持 Content-Type 指定的类型。 |
| 400 | MissingContentType | Content-Type does not exist in http header when body is not empty. | 没有为 Body 不为空的 HTTP 请求指定 Content-Type 头。 |
| 400 | MissingBodyRawSize | x-log-bodyrawsize does not exist in header when it is necessary. | 压缩场景下没有提供必须的 x-log-bodyrawsize 请求头。 |
| 400 | InvalidBodyRawSize | x-log-bodyrawsize is invalid. | x-log-bodyrawsize 的值无效。 |
| 400 | InvalidCompressType | x-log-compresstype {type} is unsupported. | x-log-compresstype 指定的压缩方式不支持。 |
| 400 | MissingHost | Host does not exist in http header. | 没有提供 HTTP 标准请求头 Host。 |
| 400 | MissingDate | Date does not exist in http header. | 没有提供 HTTP 标准请求头 Date。 |
| 400 | InvalidDateFormat | Date {date} must follow RFC822. | Date 请求头的值不符合 RFC822 标准。 |
| 400 | MissingAPIVersion | x-log-apiversion does not exist in http header. | 没有提供 HTTP 请求头 x-log-apiversion。 |
| 400 | InvalidAPIVersion | x-log-apiversion {version} is unsupported. | HTTP 请求头 x-log-apiversion 的值不支持。 |
| 400 | MissAccessKeyId | x-log-accesskeyid does not exist in header. | 没有在 Authorization 头部提供 AccessKeyId。 |
| 401 | Unauthorized | The AccessKeyId is unauthorized. | 提供的 AccessKeyId 值未授权。 |

| HTTP 状态码（Status Code） | 错误码（Error Code） | 错误消息（Error Message） | 描述（Description） |
| --- | --- | --- | --- |
| 400 | MissingSignatureMethod | x-log-signaturemethod does not exist in http header. | 没有提供 HTTP 请求头 x-log-signaturemethod。 |
| 400 | InvalidSignatureMethod | signature method {method} is unsupported. | x-log-signaturemethod 头部指定的签名方法不支持。 |
| 400 | RequestTimeTooSkewed | Request time exceeds server time more than 15 minutes. | 请求的发送时间超过当前服务处理时间前后 15 分钟的范围。 |
| 404 | ProjectNotExist | Project {name} does not exist. | 日志项目（Project）不存在。 |
| 401 | SignatureNotMatch | Signature {signature} is not matched. | 请求的数字签名不匹配。 |
| 403 | WriteQuotaExceed | Write quota is exceeded. | 超过写入日志限额。 |
| 403 | ReadQuotaExceed | Read quota is exceeded. | 超过读取日志限额。 |
| 500 | InternalServerError | Internal server error message. | 服务器内部错误。 |
| 503 | ServerBusy | The server is busy, please try again later. | 服务器正忙，请稍后再试。 |

错误消息中包括 {…} 部分为出错相关的具体信息。例如，ProjectNotExist 的错误消息中包括{name}，表示错误消息中该部分会被具体的 Project Name 来替换。

## 3.9 日志项目接口

### 3.9.1 CreateProject

创建一个 Project。

示例：

```
POST /
```

### 请求语法

```
POST / HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/json
Content-MD5: <Content-MD5>
Content-Length: <ContentLength>
Connection: Keep-Alive
{
  "projectName": <ProjectName>,
  "description": <Description>
}
```

### 请求参数

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| projectName | string | 是 | Project 名称。 |
| description | string | 是 | Project 描述。 |

请求头

CreateProject接口无特有请求头，关于 Log Service API 的公共请求头，请参考#unique_62。

响应头

CreateProject接口无特有响应头，关于 Log Service API 的公共响应头，请参考#unique_76。

响应元素

HTTP 状态码返回 200。

错误码

除了返回 Log Service API 的#unique_77，还可能返回如下特有错误码：

| HTTP状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 400 | ProjectAlreadyExist | Project {project} already exist |
| 400 | ParameterInvalid | The body is not valid json string |

| HTTP状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 500 | InternalServerError | Specified Server Error Message |

**示例**

**请求示例：**

```
POST / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 80
Connection: Keep-Alive
{
  "projectName": "my-project-test",
  "description": "Description of my-project-test"
}
```

**响应示例：**

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-log-requestid: 5B0A619F205DC3F30EDA9322
```

## 3.9.2 DeleteProject

**删除一个指定的 Project。**

**示例：**

```
DELETE /
```

**请求语法**

```
DELETE / HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
```

```
Connection: Keep-Alive
```

### 请求参数

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| projectName | string | 是 | Project 的名称，作为Header中Host的一部分。 |

#### 请求头

DeleteProject接口无特有请求头，关于 Log Service API 的公共请求头，请参考#unique_62。

#### 响应头

DeleteProject接口无特有响应头，关于 Log Service API 的公共响应头，请参考#unique_76。

#### 响应元素

HTTP 状态码返回 200。

#### 错误码

除了返回 Log Service API 的#unique_77，还可能返回如下特有错误码：

| HTTP状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | ProjectNotExist | The Project does not exist : {Project} |
| 500 | InternalServerError | Specified Server Error Message |

### 示例

#### 请求示例：

```
DELETE / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 08:25:04 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

#### 响应示例：

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
```

```
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 08:25:04 GMT
x-log-requestid: 5B0A6B60BB6EE39764D458B5
```

## 3.9.3 UpdateProject

**修改一个指定的Project 。**

**示例：**

```
PUT /
```

**请求语法**

```
PUT / HTTP/1.1
Authorization:  <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent:  <UserAgent>
x-log-apiversion: 0.6.0
Host:  <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date:  <GMT Date>
Content-Type: application/json
Content-MD5:  <Content-MD5>
Content-Length:  <ContentLength>
Connection: Keep-Alive
```

**请求参数**

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| projectName | string | 是 | Project 的名称，作为Header中Host的一部分。 |
| description | string | 否，默认为空字符串。 | Project 描述。 |

**请求头**

UpdateProject 接口无特有请求头，关于 Log Service API 的公共请求头，请参考 #unique_62。

**响应头**

UpdateProject 接口无特有响应头，关于 Log Service API 的公共响应头，请参考 #unique_62。

**响应元素**

HTTP 状态码返回 200。

### 错误码

除了返回 Log Service API 的 #unique_77，还可能返回如下特有错误码：

| HTTP状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | ProjectNotExist | The Project does not exist : <Project>. |
| 400 | ParameterInvalid | The body is not valid json string. |
| 500 | InternalServerError | Specified Server Error Message. |

### 示例

请求示例：

```
PUT / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 07:43:26 GMT
Content-Type: application/json
Content-MD5: A7967D81EFF5E3CD447FB6D8DF294E20
Content-Length: 40
Connection: Keep-Alive
{
  "description": "Description of my-project-test"
}
```

响应示例：

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 07:43:27 GMT
x-log-requestid: 5B0A619F205DC3F30EDA9322
```

## 3.9.4 GetProject

GetProject 根据Project 名称查询 Project。

示例：

```
GET /
```

### 请求语法

```
GET / HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Project Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

### 请求参数

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| projectName | string | 是 | Project 的名称，作为Header中Host的一部分。 |

请求头

GetProject接口无特有请求头，关于 Log Service API 的公共请求头，请参考#unique_62。

响应头

GetProject接口无特有响应头，关于 Log Service API 的公共响应头，请参考#unique_76。

响应元素

GetProject 请求成功，其响应 Body 中包含 Project 的详细信息，具体格式为：

| 属性名称 | 类型 | 描述 |
| --- | --- | --- |
| createTime | string | 创建时间。 |
| description | string | Project 描述 |
| lastModifyTime | string | 最后一次更新时间。 |
| owner | string | 创建人的账户Id。 |
| projectName | string | Project名称。 |
| status | string | Project 状态。 |
| region | string | Project 所属的区域。 |

错误码

除了返回 Log Service API 的#unique_77，还可能返回如下特有错误码：

| HTTP状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | ProjectNotExist | The Project does not exist : {Project} |
| 500 | InternalServerError | Specified Server Error Message |

**示例**

请求示例：

```
GET / HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: my-project-test.cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 08:25:04 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

响应示例：

```
HTTP/1.1 200
Server: nginx
Content-Length: 0
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 08:25:04 GMT
x-log-requestid: 5B0A6B60BB6EE39764D458B5
```

## 3.9.5 ListProject

查询所有 Project 列表。

示例：

```
GET /?offset={offset}&size={size}
```

**请求语法**

```
GET /?offset={offset}&size={size} HTTP/1.1
Authorization: <AuthorizationString>
x-log-bodyrawsize: 0
User-Agent: <UserAgent>
x-log-apiversion: 0.6.0
Host: <Endpoint>
x-log-signaturemethod: hmac-sha1
Date: <GMT Date>
Content-Type: application/x-protobuf
```

```
Connection: Keep-Alive
```

**请求参数**

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| offset | integer | 否 | 返回记录的起始位置，默认值为 0。 |
| size | integer | 否 | 每页返回最大条目，默认 500（最大值）。 |

**请求头**

ListProject接口无特有请求头，关于 Log Service API 的公共请求头，请参考#unique_62。

**响应头**

ListProject接口无特有响应头，关于 Log Service API 的公共响应头，请参考#unique_76。

**响应元素**

ListProject请求成功，其响应 Body 中包含 Project 列表，具体格式如下：

| 属性名称 | 类型 | 描述 |
| --- | --- | --- |
| count | integer | 返回的 Project 个数。 |
| total | integer | Project 总数。 |
| projects | array | Project 列表。 |

projects 中每个元素的格式为：

| 属性名称 | 类型 | 描述 |
| --- | --- | --- |
| createTime | string | 创建时间。 |
| description | string | Project 描述 |
| lastModifyTime | string | 最后一次更新时间。 |
| owner | string | 创建人的账户Id。 |
| projectName | string | Project名称。 |
| status | string | Project 状态。 |
| region | string | Project 所属的区域。 |

**错误码**

除了返回 Log Service API 的#unique_77，还可能返回如下特有错误码：

| HTTP状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 500 | InternalServerError | Specified Server Error Message |

**示例**

**请求示例：**

```
GET /?offset=0&size=2&projectName= HTTP/1.1
Authorization: LOG <yourAccessKeyId>:<yourSignature>
x-log-bodyrawsize: 0
User-Agent: sls-java-sdk-v-0.6.1
x-log-apiversion: 0.6.0
Host: cn-shanghai.log.aliyuncs.com
x-log-signaturemethod: hmac-sha1
Date: Sun, 27 May 2018 09:03:33 GMT
Content-Type: application/x-protobuf
Connection: Keep-Alive
```

**响应示例：**

```
HTTP/1.1 200
Server: nginx
Content-Type: application/json
Content-Length: 345
Connection: close
Access-Control-Allow-Origin: *
Date: Sun, 27 May 2018 09:03:33 GMT
x-log-requestid: 5B0A7465AAEA20CA70DE3064
{
  "count": 2,
  "total": 11,
  "projects": [
    {
      "projectName": "project1",
      "status": "Normal",
      "owner": "",
      "description": "",
      "region": "cn-shanghai",
      "createTime": "1524222931",
      "lastModifyTime": "1524539357"
    },
    {
      "projectName": "project123456",
      "status": "Normal",
      "owner": "",
      "description": "",
      "region": "cn-shanghai",
      "createTime": "1471963876",
      "lastModifyTime": "1524539357"
    }
  ]
```

```
}
```

## 3.9.6 GetProjectLogs

GetProjectLogs是Project 级别的SQL查询接口。

### 请求语法

```
GET /logs/?query=SELECT avg(latency) as avg_latency FROM  where
__date__ >'2017-09-01 00:00:00' and __date__ < '2017-09-02 00:00:00'
Authorization: <AuthorizationString>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
x-log-apiversion: 0.4.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| query | string | 是 | 查询SQL条件 |

### 请求头

GetProjectLogs接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

### 响应头

关于 Log Service API 的公共响应头，请参考 公共响应头。

响应头中有专门成员表示请求返回结果是否完整。具体响应元素格式如下：

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| x-log-progress | 字符串 | 查询结果的状态。可以有 Incomplete 和 Complete 两个选值，表示本次是否完整。 |
| x-log-count | 整型 | 当前查询结果的日志总数。 |
| x-log-processed-rows | 整型 | 本次计算处理的行数。 |
| x-log-elapsed-millisecond | 整型 | 本次计算消耗的毫秒时间。 |

### 响应元素

请求成功，其响应 Body 会包括计算结果，GetProjectLogs的响应body是一个数组，数组中每个元素是一条日志结果。数组中的每个元素结构如下：

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| __time__ | 整型 | 日志的时间戳（精度为秒，从1970-1-1 00:00:00 UTC 计算起的秒数）。 |
| __source__ | 字符串 | 日志的来源，由写入日志时指定。 |
| [content] | Key-Value对 | 日志原始内容，以 Key-value 对的形式组织。 |

细节描述

- 该接口的query是一个标准的SQL查询语句。
- 查询的Project，在请求的域名中指定。
- 查询的logstore，在查询语句的from条件中指定。logstore相当于SQL中的表。
- 在查询的SQL条件中，必须指定要查询的时间范围，时间范围由__date__（timestamp类型）来指定，或__time__（int 类型，单位是unix_time）来指定。
- 如上所述，该接口一次调用必须要在限定时间内返回结果，每次查询只能扫描指定条数的日志量。如果一次请求需要处理的数据量非常大的时候，该请求会返回不完整的结果（并在返回结果中的 x-log-progress 成员标示是否完整）。与此同时，服务端会缓存 15 分钟内的查询结果。当查询请求的结果有部分被缓存命中，则服务端会在这次请求中继续扫描未被缓存命中的日志数据。为了减少您合并多次查询结果的工作量，服务端会把缓存命中的查询结果与本次查询新命中的结果合并返回给您。因此，日志服务可以让您通过以相同参数反复调用该接口来获取最终完整结果。因为您的查询涉及的日志数据量变化非常大，日志服务 API 无法预测需要调用多少次该接口而获取完整结果。所以需要用户通过检查每次请求的返回结果中的x-log-progress成员状态值来确定是否需要继续。需要注意的是，每次重复调用该接口都会重新消耗相同数量的查询CU。

特有错误码

GetProjectLogs 接口除了返回 API 的 通用错误码，还可能返回如下特有错误码：

| HTTP状态码（Status Code） | 错误码（Error Code） | 错误消息（Error Message） | 描述（Description） |
| --- | --- | --- | --- |
| 400 | ParameterInvalid | parameter is invalid | 请求的参数错误，具体错误参考错误的message。 |

### 示例

以杭州地域内名为 big-game 的 Project 为例，查询该 Project 内名为 app_log 的 Logstore 中，主题为 groupA 的日志数据。查询区间为 2014-09-01 00:00:00 到 2014-09-01 22:00:00，查询关键字为 error，且从时间区间头开始查询，最多返回 20 条日志数据。

### 请求示例

```
GET /logs/?query=SELECT * FROM <logStoreName> where __line__ = 'abc'
and __date__ >'2017-09-01 00:00:00' and __date__ < '2017-09-02 00:00:
00'&line=20&offset=0 HTTP/1.1
Authorization: <AuthorizationString>
Date: Wed, 3 Sept. 2014 08:33:46 GMT
Host: big-game.cn-hangzhou.log.aliyuncs.com
x-log-bodyrawsize: 0
x-log-apiversion: 0.4.0
x-log-signaturemethod: hmac-sha1
```

### 响应示例

```
HTTP/1.1 200 OK
Content-MD5: 36F9F7F0339BEAF571581AF1B0AAAFB5
Content-Type: application/json
Content-Length: 269
Date: Wed, 3 Sept. 2014 08:33:47 GMT
x-log-requestid: efag01234-12341-15432f
x-log-progress : Complete
x-log-count : 10000
x-log-processed-rows: 10000
x-log-elapsed-millisecond:5
{
    "progress": "Complete",
    "count": 2,
    "logs": [
        {
            "__time__": 1409529660,
            "__source__": "10.237.0.17",
            "Key1": "error",
            "Key2": "Value2"
        },
        {
            "__time__": 1409529680,
            "__source__": "10.237.0.18",
            "Key3": "error",
            "Key4": "Value4"
        }
    ]
}
```

在这个响应示例中，x-log-progress 成员的状态为 Complete，表明整个日志查询已经完成，返回结果为完整结果。在这次请求中共查询到 2 条符合条件的日志，且日志数据在 logs 成员中。如果响应结果中的 x-log-progress 成员的状态为 Incomplete，则需要重复相同请求以获得完整结果。

# 3.10 Logtail机器组相关接口

## 3.10.1 CreateMachineGroup

**您可以根据需求创建一组机器，用以日志收集下发配置。**

**示例：**

```
POST /machinegroups
```

**请求语法**

```
POST /machinegroups HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
    "groupName" : "testgroup",
    "groupType" : "",
    "groupAttribute" : {
        "externalName" : "testgroup",
        "groupTopic": "testgrouptopic"
    },
    "machineIdentifyType" : "ip",
    "machineList" : [
        "test-ip1",
        "test-ip2"
    ]
}
```

**请求参数**

**Body 参数：**

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupName | string | 是 | 机器分组名称，Project 下唯一 |
| groupType | string | 否 | 机器分组类型，默认为空 |
| machineIdentifyType | string | 是 | 机器标识类型，分为 IP 和 userdefined 两种 |
| groupAttribute | object | 是 | 机器分组的属性，默认为空 |

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| machineList | array | 是 | 具体的机器标识，可以是 IP 或 userdefined-id |

groupAttribute 说明如下：

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupTopic | string | 否 | 机器分组的 topic，默认为空 |
| externalName | string | 否 | 机器分组所依赖的外部管理标识，默认为空 |

**请求头**

CreateMachineGroup接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

**响应头**

CreateMachineGroup接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

**响应元素**

HTTP 状态码返回 200。

**错误码**

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 400 | MachineGroupAlreadyExist | group {GroupName} already exists |
| 400 | InvalidParameter | invalid group resource json |
| 500 | InternalServerError | Internal server error |

## 示例

**请求示例：**

```
POST /machinegroups HTTP/1.1
Header :
{
```

```
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Tue, 10 Nov 2015 17:57:33 GMT",
    "Content-Length": "187",
    "x-log-signaturemethod": "hmac-sha1",
    "Content-MD5": "82033D507DEAAD72067BB58DFDCB590D",
    "User-Agent": "sls-java-sdk-v-0.6.0",
    "Content-Type": "application/json",
    "x-log-bodyrawsize": "0"
}
Body :
{
    "groupName": "test-machine-group",
    "groupType": "",
    "machineIdentifyType": "ip",
    "groupAttribute":     {
        "groupTopic": "testtopic",
        "externalName": "testgroup"
    },
    "machineList":     [
        "127.0.0.1",
        "127.0.0.2"
    ]
}
```

**响应示例：**

```
HTTP/1.1 200 OK
Header :
{
    "Date": "Tue, 10 Nov 2015 17:57:33 GMT",
    "Content-Length": "0",
    "x-log-requestid": "5642300D99248CB76D005D36",
    "Connection": "close",
    "Server": "nginx/1.6.1"
}
```

## 3.10.2 DeleteMachineGroup

**删除机器组，如果机器组上有配置，则 Logtail 上对应的配置也会被删除。**

**示例：**

```
DELETE /machinegroups/{groupName}
```

**请求语法**

```
DELETE /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
```

```
x-log-signaturemethod: hmac-sha1
```

### 请求参数

**URL 参数：**

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupName | string | 是 | 机器分组名称 |

**请求头**

DeleteMachineGroup接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

**响应头**

DeleteMachineGroup接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

**响应元素**

HTTP 状态码返回 200。

**错误码**

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | GroupNotExist | group {GroupName} does not exist |
| 500 | InternalServerError | internal server error |

### 示例

**请求示例：**

```
DELETE /machinegroups/test-machine-group-4 HTTP/1.1
Header :
{
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Tue, 10 Nov 2015 19:13:28 GMT",
    "Content-Length": "0",
    "x-log-signaturemethod": "hmac-sha1",
    "User-Agent": "sls-java-sdk-v-0.6.0",
    "Content-Type": "application/x-protobuf",
    "x-log-bodyrawsize": "0"
```

```
}
```

**响应示例：**

```
HTTP/1.1 200 OK
Header :
{
    "Date": "Tue, 10 Nov 2015 19:13:28 GMT",
    "Content-Length": "0",
    "x-log-requestid": "564241D899248C827B000CFE",
    "Connection": "close",
    "Server": "nginx/1.6.1"
}
```

## 3.10.3 UpdateMachineGroup

**更新机器组信息，如果机器组已应用配置，则新加入、减少机器会自动增加、移除配置。**

**示例：**

```
PUT /machinegroups/{groupName}
```

**请求语法**

```
PUT /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
    "groupName": "test-machine-group",
    "groupType" : "",
    "groupAttribute" : {
        "externalName" : "testgroup",
        "groupTopic": "testgrouptopic"
    },
    "machineIdentifyType" : "ip",
    "machineList" : [
        "test-ip1",
        "test-ip2"
    ]
}
```

**请求参数**

**URL 参数：**

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupName | string | 是 | 机器分组名称 |

**Body 参数：**

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupName | string | 是 | 机器分组名称，Project 下唯一 |
| groupType | string | 否 | 机器分组类型，默认为空 |
| machineIdentifyType | string | 是 | 机器标识类型，分为 ip 和 userdefined 两种 |
| groupAttribute | object | 是 | 机器分组的属性，默认为空 |
| machineList | array | 是 | 具体的机器标识，可以是 ip 或 userdefined-id |

groupAttribute 说明如下：

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupTopic | string | 否 | 机器分组的 topic，默认为空 |
| externalName | string | 否 | 机器分组所依赖的外部管理标识，默认为空 |

**请求头**

UpdateMachineGroup接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

**响应头**

UpdateMachineGroup接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

**响应元素**

HTTP 状态码返回 200。

**错误码**

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | GroupNotExist | group {GroupName} does not exist |
| 400 | InvalidParameter | invalid group resource json |
| 500 | InternalServerError | internal server error |

## 示例

**请求示例：**

```
PUT /machinegroups/test-machine-group HTTP/1.1
Header :
{
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Tue, 10 Nov 2015 18:41:43 GMT",
    "Content-Length": "194",
    "x-log-signaturemethod": "hmac-sha1",
    "Content-MD5": "2CEBAEBE53C078891527CB70A855BAF4",
    "User-Agent": "sls-java-sdk-v-0.6.0",
    "Content-Type": "application/json",
    "x-log-bodyrawsize": "0"
}
Body :
{
    "groupName": "test-machine-group",
    "groupType": "",
    "machineIdentifyType": "userdefined",
    "groupAttribute":     {
        "groupTopic": "testtopic2",
        "externalName": "testgroup2"
    },
    "machineList":     [
        "uu_id_1",
        "uu_id_2"
    ]
}
```

**响应示例：**

```
HTTP/1.1 200 OK
Header :
{
    "Date": "Tue, 10 Nov 2015 18:41:43 GMT",
    "Content-Length": "0",
    "x-log-requestid": "56423A6799248CA57B00035C",
    "Connection": "close",
    "Server": "nginx/1.6.1"
```

```
}
```

## 3.10.4 ListMachineGroup

查看机器组名称列表。

示例：

```
GET /machinegroups?offset=1&size=100
```

### 请求语法

```
GET /machinegroups?offset=1&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

URL 参数：

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| offset | int | 否 | 返回记录的起始位置，默认为 0 |
| size | int | 否 | 每页返回最大条目，默认 500（最大值） |
| groupName | string | 否 | 用于过滤的机器组名称（支持部分匹配） |

#### 请求头

ListMachineGroup接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

#### 响应头

ListMachineGroup接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

#### 响应元素

请求成功，其响应 Body 会包括指定 Project 下的所有 machinegroup 名称列表。具体格式如下：

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| count | int | 返回的 machinegroup 数目 |
| total | int | 返回 machinegroup 总数 |
| machinegroups | json array | 返回的 machinegroup 名称列表 |

```
{
    "machinegroups":    [
        "test-machine-group",
        "test-machine-group-2"
    ],
    "count": 2,
    "total": 2
}
```

**错误码**

**除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：**

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 500 | InternalServerError | internal server error |

## 示例

**请求示例：**

```
GET /machinegroups?groupName=test-machine-group&offset=0&size=3 HTTP/1
.1
Header :
{
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Tue, 10 Nov 2015 18:34:44 GMT",
    "Content-Length": "0",
    "x-log-signaturemethod": "hmac-sha1",
    "User-Agent": "sls-java-sdk-v-0.6.0",
    "Content-Type": "application/x-protobuf",
    "x-log-bodyrawsize": "0"
}
```

**响应示例：**

```
HTTP/1.1 200 OK
Header :
{
    "Date": "Tue, 10 Nov 2015 18:34:44 GMT",
    "Content-Length": "83",
    "x-log-requestid": "564238C499248C8F7B0001DE",
    "Connection": "close",
    "Content-Type": "application/json",
    "Server": "nginx/1.6.1"
```

```
}
Body :
{
    "machinegroups":     [
        "test-machine-group",
        "test-machine-group-2"
    ],
    "count": 2,
    "total": 2
}
```

## 3.10.5 GetMachineGroup

查看具体的 MachineGroup 信息。

示例：

```
GET /machinegroups/{groupName}
```

### 请求语法

```
GET /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

URL 参数：

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupName | string | 是 | 机器分组名称 |

请求头

GetMachineGroup接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

响应头

GetMachineGroup接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

响应元素

| 属性名称 | 类型 | 描述 |
| --- | --- | --- |
| groupName | string | 机器分组名称，Project 下唯一 |

| 属性名称 | 类型 | 描述 |
| --- | --- | --- |
| groupType | string | 分组类型（空或者 Armory），默认为空 |
| machineIdentifyType | string | 机器标识类型，分为 IP 和 userdefined 两种 |
| groupAttribute | json object | 机器分组的属性，默认为空 |
| machineList | json array | 具体的机器标识，可以是 IP 或 userdefined-id |
| createTime | int | 机器分组创建时间 |
| lastModifyTime | int | 机器分组最近更新时间 |

groupAttribute 说明如下：

| 属性名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupTopic | string | 否 | 机器分组的 topic，一般不设置 |
| externalName | string | 否 | 机器分组所依赖的外部管理系统（Armory）标识 |

```
{
    "groupName": "test-machine-group",
    "groupType": "",
    "groupAttribute":     {
        "externalName": "testgroup",
        "groupTopic": "testtopic"
    },
    "machineIdentifyType": "ip",
    "machineList":     [
        "127.0.0.1",
        "127.0.0.2"
    ],
    "createTime": 1447178253,
    "lastModifyTime": 1447178253
}
```

**错误码**

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | MachineGroupNotExist | group {GroupName} does not exist |
| 500 | InternalServerError | internal server error |

**示例**

**请求示例：**

```
GET /machinegroups/test-machine-group HTTP/1.1
Header :
{
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Tue, 10 Nov 2015 18:15:24 GMT",
    "Content-Length": "0",
    "x-log-signaturemethod": "hmac-sha1",
    "User-Agent": "sls-java-sdk-v-0.6.0",
    "Content-Type": "application/x-protobuf",
    "x-log-bodyrawsize": "0"
}
```

**响应示例：**

```
HTTP/1.1 200 OK
Header :
{
    "Date": "Tue, 10 Nov 2015 18:15:23 GMT",
    "Content-Length": "239",
    "x-log-requestid": "5642343B99248CB36D0060B8",
    "Connection": "close",
    "Content-Type": "application/json",
    "Server": "nginx/1.6.1"
}
Body :
{
    "groupName": "test-machine-group",
    "groupType": "",
    "groupAttribute":     {
        "externalName": "testgroup",
        "groupTopic": "testtopic"
    },
    "machineIdentifyType": "ip",
    "machineList":     [
        "127.0.0.1",
        "127.0.0.2"
    ],
    "createTime": 1447178253,
    "lastModifyTime": 1447178253
}
```

## 3.10.6 ApplyConfigToMachineGroup

**将配置应用到机器组**。

示例：

```
PUT /machinegroups/{GroupName}/configs/{ConfigName}
```

### 请求语法

```
GET /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| GroupName | string | 是 | 机器分组名称 |
| ConfigName | string | 是 | 日志配置名称 |

### 请求头

ApplyConfigToMachineGroup接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

### 响应头

ApplyConfigToMachineGroup接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

### 响应元素

HTTP 状态码返回 200。

### 错误码

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | GroupNotExist | group {GroupName} does not exist |
| 404 | ConfigNotExist | config {ConfigName} does not exist |
| 500 | InternalServerError | internal server error |

**示例**

**请求示例：**

```
PUT /machinegroups/sample-group/configs/logtail-config-sample
Header :
{
    "Content-Length": 0,
    "x-log-signaturemethod": "hmac-sha1",
    "x-log-bodyrawsize": 0,
    "User-Agent": "log-python-sdk-v-0.6.0",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Mon, 09 Nov 2015 09:44:43 GMT",
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

**响应示例：**

```
{
    "date": "Mon, 09 Nov 2015 09:44:43 GMT",
    "connection": "close",
    "x-log-requestid": "56406B0B99248CAA230BA094",
    "content-length": "0",
    "server": "nginx/1.6.1"
}
```

## 3.10.7 RemoveConfigFromMachineGroup

**从机器组中删除配置。**

**示例：**

```
DELETE /machinegroups/{GroupName}/configs/{ConfigName}
```

**请求语法**

```
GET /machinegroups/{groupName} HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

**请求参数**

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| GroupName | string | 是 | 机器分组名称 |
| ConfigName | string | 是 | 日志配置名称 |

**请求头**

RemoveConfigFromMachineGroup接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

响应头

RemoveConfigFromMachineGroup接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

响应元素

HTTP 状态码返回 200。

错误码

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | GroupNotExist | group {GroupName} does not exist |
| 404 | ConfigNotExist | config {ConfigName} does not exist |
| 500 | InternalServerError | internal server error |

示例

请求示例：

```
DELETE /machinegroups/sample-group/configs/logtail-config-sample
{
    "Content-Length": 0,
    "x-log-signaturemethod": "hmac-sha1",
    "x-log-bodyrawsize": 0,
    "User-Agent": "log-python-sdk-v-0.6.0",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Mon, 09 Nov 2015 09:48:48 GMT",
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

响应示例：

```
{
    "date": "Mon, 09 Nov 2015 09:48:48 GMT",
    "connection": "close",
    "x-log-requestid": "56406C0099248CAA230BE135",
    "content-length": "0",
    "server": "nginx/1.6.1"
```

```
}
```

## 3.10.8 ListMachines

获得 machinegroup 下属于用户并与 Server 端连接的机器状态信息。

示例：

```
GET /machinegroups/{groupName}/machines?offset=1&size=10
```

### 请求语法

```
GET /machinegroups/{groupName}/machines HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

URL 参数：

| 名称 | 类型 | 必须 | 描述 |
| --- | --- | --- | --- |
| groupName | string | 是 | 机器分组名称 |
| offset | int | 否 | 返回记录的起始位置，默认为 0 |
| size | int | 否 | 每页返回最大条目，默认 500（最大值） |

### 请求头

ListMachines接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

### 响应头

ListMachines接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

### 响应元素

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| count | int | 返回的 machine 数目 |
| total | int | machine 总数 |
| machines | json array | 返回的 machine 名称列表 |

machines 说明如下：

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| ip | string | 机器的 IP |
| machine-uniqueid | string | 机器 DMI UUID |
| userdefined-id | string | 机器的用户自定义标识 |

```
{
    "count":10,
    "total":100,
    "machines":
    [{
        "ip" : "testip1",
        "machine-uniqueid" : "testuuid1",
        "userdefined-id" : "testuserdefinedid1",
        "lastHeartbeatTime" : 1447182247
    },
    {
        "ip" : "testip1",
        "machine-uniqueid" : "testuuid2",
        "userdefined-id" : "testuserdefinedid2",
        "lastHeartbeatTime" : 1447182247
    },
    {
        "ip" : "testip2",
        "machine-uniqueid" : "testuuid",
        "userdefined-id" : "testuserdefinedid"
        "lastHeartbeatTime" : 1447182247
    }]
}
```

**错误码**

**除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：**

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | GroupNotExist | group {GroupName} does not exist |
| 500 | InternalServerError | internal server error |

**细节描述**

**该接口只获取与 Server 端连接正常的机器列表。**

**示例**

**请求示例：**

```
GET /machinegroups/test-machine-group-5/machines?offset=0&size=3 HTTP/
1.1
Header :
{
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
```

```
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Tue, 10 Nov 2015 19:04:57 GMT",
    "Content-Length": "0",
    "x-log-signaturemethod": "hmac-sha1",
    "User-Agent": "sls-java-sdk-v-0.6.0",
    "Content-Type": "application/x-protobuf",
    "x-log-bodyrawsize": "0"
}
```

**响应示例：**

```
HTTP/1.1 200 OK
Header :
{
    "Date": "Tue, 10 Nov 2015 19:04:58 GMT",
    "Content-Length": "324",
    "x-log-requestid": "56423FD999248C827B000A57",
    "Connection": "close",
    "Content-Type": "application/json",
    "Server": "nginx/1.6.1"
}
Body :
{
    "machines":     [
        {
            "ip": "10.101.166.116",
            "machine-uniqueid": "",
            "userdefined-id": "",
            "lastHeartbeatTime": 1447182247
        },
        {
            "ip": "10.101.165.193",
            "machine-uniqueid": "",
            "userdefined-id": "",
            "lastHeartbeatTime": 1447182246
        },
        {
            "ip": "10.101.166.91",
            "machine-uniqueid": "",
            "userdefined-id": "",
            "lastHeartbeatTime": 1447182248
        }
    ],
    "count": 3,
    "total": 8
}
```

## 3.10.9 GetAppliedConfigs

**获得 MachineGroup 上已经被应用的配置名称。**

**示例：**

```
GET /machinegroups/{groupName}/configs
```

**请求语法**

```
GET /machinegroups/{groupName}/configs HTTP/1.1
Authorization: <AuthorizationString>
```

```
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

URL 参数：

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| groupName | string | 是 | 机器分组名称 |

#### 请求头

GetAppliedConfigs接口无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

#### 响应头

GetAppliedConfigs接口无特有响应头。关于 Log Service API 的公共响应头，请参考 公共响应头。

#### 响应元素

请求成功，其响应 Body 会包括指定 machinegroup 下的所有 machine 列表，具体格式如下：

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| count | 整型 | 返回的 config 数目。 |
| configs | 字符串数组 | 返回的 config 名称列表。 |

```
{
    "count":2,
    "configs":
    ["config1","config2"]
}
```

#### 错误码

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | GroupNotExist | group {GroupName} does not exist |
| 500 | InternalServerError | internal server error |

**示例**

**请求示例：**

```
GET /machinegroups/test-machine-group/configs HTTP/1.1
Header :
{
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Tue, 10 Nov 2015 19:45:48 GMT",
    "Content-Length": "0",
    "x-log-signaturemethod": "hmac-sha1",
    "User-Agent": "sls-java-sdk-v-0.6.0",
    "Content-Type": "application/x-protobuf",
    "x-log-bodyrawsize": "0"
}
```

**响应示例：**

```
HTTP/1.1 200 OK
Header :
{
    "Date": "Tue, 10 Nov 2015 19:45:48 GMT",
    "Content-Length": "53",
    "x-log-requestid": "5642496C99248C8C7B00173F",
    "Connection": "close",
    "Content-Type": "application/json",
    "Server": "nginx/1.6.1"
}
Body :
{
    "configs":     [
        "two",
        "three",
        "test_logstore"
    ],
    "count": 3
}
```

# 3.11 Logtail配置相关接口

## 3.11.1 CreateConfig

**在Project下创建日志配置。**

**示例：**

```
POST /configs
```

**请求语法**

```
POST /configs HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
```

```
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
    "configName": "testcategory1",
    "inputType": "file",
    "inputDetail": {
        "logType": "common_reg_log",
        "logPath": "/var/log/httpd/",
        "filePattern": "access*.log",
        "localStorage": true,
        "timeFormat": "%Y/%m/%d %H:%M:%S",
        "logBeginRegex": ".*",
        "regex": "(\w+)(\s+)",
        "key" :["key1", "key2"],
        "filterKey":["key1"],
        "filterRegex":["regex1"],
        "fileEncoding":"utf8",
        "topicFormat": "none"
    },
    "outputType": "LogService",
    "outputDetail":
    {
        "logstoreName": "perfcounter"
    }
}
```

**请求参数**

**说明:**

**请求参数和各种模式的Logtail配置样例请参考#unique_15。**

**请求头**

**无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。**

**响应头**

**无特有响应头。请参考 公共响应头。**

**响应元素**

**HTTP 状态码返回 200。**

**错误码**

**除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：**

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 400 | ConfigAlreadyExist | config {Configname} already exists |

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 400 | InvalidParameter | invalid config resource json |
| 500 | InternalServerError | internal server error |

**细节描述**

**创建过程中遇到配置已经存在、格式错误、必要参数遗失或者 quota 超过限制等错误，则会创建失败。**

**示例**

**请求示例：**

```
POST /configs HTTP/1.1
Header :
{
    'Content-Length': 737,
    'Host': 'ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com',
    'x-log-bodyrawsize': 737,
    'Content-MD5': 'FBA01ECF7255BE143379BC70C56BBF68',
    'x-log-signaturemethod': 'hmac-sha1',
    'Date': 'Mon, 09 Nov 2015 07:45:30 GMT',
    'x-log-apiversion': '0.6.0',
    'User-Agent': 'log-python-sdk-v-0.6.0',
    'Content-Type': 'application/json',
    'Authorization': 'LOG <yourAccessKeyId>:<yourSignature>'
}
Body:
{
    "configName": "sample-logtail-config",
    "inputType": "file",
    "inputDetail": {
        "logType": "common_reg_log",
        "logPath": "/var/log/httpd/",
        "filePattern": "access*.log",
        "localStorage": true,
        "timeFormat": "%d/%b/%Y:%H:%M:%S",
        "logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
        "regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+)
 ([^\"]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \"([^\"]*)\" \"([^
\"]*)\".*",
        "key": ["ip", "time", "method", "url", "request_time", "
request_length", "status", "length", "ref_url", "browser"],
        "filterKey": [],
        "filterRegex": [],
        "topicFormat": "none",
        "fileEncoding": "utf8"
    },
    "outputType": "LogService",
    "outputDetail":
    {
        "logstoreName": "sls-test-logstore"
    }
```

```
}
```

响应示例：

```
HTTP/1.1 200 OK
Header
{
    'date': 'Mon, 09 Nov 2015 07:45:30 GMT',
    'connection': 'close',
    'x-log-requestid': '56404F1A99248CA26C002180',
    'content-length': '0',
    'server': 'nginx/1.6.1'
}
```

## 3.11.2 ListConfig

列出 Project 下所有配置信息，可以通过参数进行翻页。

示例：

```
GET /configs?offset=1&size=100
```

### 请求语法

```
GET /configs?offset=0&size=100 HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

URL 参数如下：

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| offset(optional) | integer | 否 | 返回记录的起始位置，默认为 0 |
| size(optional) | integer | 否 | 每页返回最大条目，默认为 500（最大值） |

请求头

无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

响应头

无特有响应头。请参考 公共响应头。

响应元素

返回值：Body 包含该 project 下所有 config 列表，具体格式如下：

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| count | 整型 | 返回的 config 数目 |
| total | 整型 | 在服务端 config 总数 |
| configs | 字符串数组 | 返回的 config 名称列表 |

**错误码**

**除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：**

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | ConfigNotExist | config {Configname} does not exist |
| 500 | InternalServerError | internal server error |

**示例**

**请求示例：**

```
GET /configs?offset=0&size=10 HTTP/1.1
Header :
{
    "Content-Length": 0,
    "x-log-signaturemethod": "hmac-sha1",
    "x-log-bodyrawsize": 0,
    "User-Agent": "log-python-sdk-v-0.6.0",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Mon, 09 Nov 2015 09:19:13 GMT",
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

**响应示例：**

```
Header :
{
    "content-length": "103",
    "server": "nginx/1.6.1",
    "connection": "close",
    "date": "Mon, 09 Nov 2015 09:19:13 GMT",
    "content-type": "application/json",
    "x-log-requestid": "5640651199248CAA2300C2BA"
}
Body:
{
    "count": 3,
    "configs":
    [
        "logtail-config-sample",
        "logtail-config-sample-2",
        "logtail-config-sample-3"
    ],
    "total": 3
```

```
}
```

## 3.11.3 GetAppliedMachineGroups

列出 config 应用的机器列表。

示例：

```
GET /configs/{configName}/machinegroups
```

### 请求语法

```
GET /configs/{configName}/machinegroups HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

URL 参数：

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| ConfigName | string | 是 | 配置名称 |

请求头

无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

响应头

无特有响应头。请参考 公共响应头。

响应元素

请求成功，其响应 Body 会包括指定 machinegroup 下的所有 machine 列表，具体格式如下：

| 名称 | 类型 | 描述 |
| --- | --- | --- |
| count | 整型 | 返回的 machineGroup 数目。 |
| machinegroups | 字符串数组 | 返回的 machineGroup 名称列表。 |

```
{
    "count":2,
    "machinegroups":
    ["group1","group2"]
```

```
}
```

**错误码**

**除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：**

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | GroupNotExist | group {GroupName} does not exist |
| 500 | InternalServerError | internal server error |

**示例**

**请求示例：**

```
GET /configs/logtail-config-sample/machinegroups
Header:
{
    "Content-Length": 0,
    "x-log-signaturemethod": "hmac-sha1",
    "x-log-bodyrawsize": 0,
    "User-Agent": "log-python-sdk-v-0.6.0",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Mon, 09 Nov 2015 09:51:38 GMT",
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

**响应示例：**

```
Header :
{
    "content-length": "44",
    "server": "nginx/1.6.1",
    "connection": "close",
    "date": "Mon, 09 Nov 2015 09:51:38 GMT",
    "content-type": "application/json",
    "x-log-requestid": "56406CAA99248CAA230BE828"
}
Body:
{
    "count": 1,
    "machinegroups":
    [
        "sample-group"
    ]
}
```

## 3.11.4 GetConfig

**获得一个配置的详细信息。**

示例：

```
GET /configs/{configName}
```

### 请求语法

```
GET /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| ConfigName | String | 是 | 日志配置名称 |

请求头

无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

响应头

无特有响应头。请参考 公共响应头。

响应元素

说明：

响应内容中包含的参数信息和各种模式的L配g置样例请参考#unique_15。

错误码

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | ConfigNotExist | Config {Configname} does not exist |
| 500 | InternalServerError | Specified Server Error Message |

### 示例

请求示例：

```
GET /configs/logtail-config-sample
Header :
{
    "Content-Length": 0,
```

```
    "x-log-signaturemethod": "hmac-sha1",
    "x-log-bodyrawsize": 0,
    "User-Agent": "log-python-sdk-v-0.6.0",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Mon, 09 Nov 2015 08:29:15 GMT",
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

**响应示例：**

```
Header :
{
    "content-length": "730",
    "server": "nginx/1.6.1",
    "connection": "close",
    "date": "Mon, 09 Nov 2015 08:29:15 GMT",
    "content-type": "application/json",
    "x-log-requestid": "5640595B99248CAA23004A59"
}
Body :
{
    "configName": "logtail-config-sample",
    "outputDetail": {
        "endpoint": "http://cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
        "logstoreName": "sls-test-logstore"
    },
    "outputType": "LogService",
    "inputType": "file",
    "inputDetail": {
        "regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+)
 ([^\"]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \"([^\"]*)\" \"([^
\"]*)\".*",
        "filterKey": [],
        "logPath": "/var/log/httpd/",
        "logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
        "logType": "common_reg_log",
        "topicFormat": "none",
        "localStorage": true,
        "key": [
            "ip",
            "time",
            "method",
            "url",
            "request_time",
            "request_length",
            "status",
            "length",
            "ref_url",
            "browser"
        ],
        "filePattern": "access*.log",
        "timeFormat": "%d/%b/%Y:%H:%M:%S",
        "filterRegex": []
    },
    "createTime": 1447040456,
    "lastModifyTime": 1447050456
```

```
}
```

## 3.11.5 DeleteConfig

删除特定 config，如果 config 已被应用到机器组，则 Logtail 配置也会被删除。

示例：

```
DELETE /configs/{configName}
```

### 请求语法

```
DELETE /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
```

### 请求参数

URL参数：

| 参数名称 | 类型 | 是否必须 | 描述 |
| --- | --- | --- | --- |
| ConfigName | String | 是 | 日志配置名称 |

请求头

无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。

响应头

无特有响应头。请参考 公共响应头。

响应元素

HTTP 状态码返回 200。

错误码

除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | ConfigNotExist | config {Configname} does not exist |
| 400 | InvalidParameter | invalid config resource json |
| 500 | InternalServerError | internal server error |

**示例**

**请求示例：**

```
DELETE /configs/logtail-config-sample
Header :
{
    "Content-Length": 0,
    "x-log-signaturemethod": "hmac-sha1",
    "x-log-bodyrawsize": 0,
    "User-Agent": "log-python-sdk-v-0.6.0",
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "Date": "Mon, 09 Nov 2015 09:28:21 GMT",
    "x-log-apiversion": "0.6.0",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
```

**响应示例：**

```
Header :
{
    "date": "Mon, 09 Nov 2015 09:28:21 GMT",
    "connection": "close",
    "x-log-requestid": "5640673599248CAA230836C6",
    "content-length": "0",
    "server": "nginx/1.6.1"
}
```

## 3.11.6 UpdateConfig

**更新配置内容，如果配置被应用到机器组，对应机器也会同时更新。**

**示例：**

```
PUT /configs/{configName}
```

**请求语法**

```
PUT /configs/<configName> HTTP/1.1
Authorization: <AuthorizationString>
Content-Type:application/json
Content-Length:<Content Length>
Content-MD5<:<Content MD5>
Date: <GMT Date>
Host: <Project Endpoint>
x-log-apiversion: 0.6.0
x-log-signaturemethod: hmac-sha1
{
    "configName": "testcategory1",
    "inputType": "file",
    "inputDetail": {
        "logType": "common_reg_log",
        "logPath": "/var/log/httpd/",
        "filePattern": "access.log",
        "localStorage": true,
        "timeFormat": "%Y/%m/%d %H:%M:%S",
        "logBeginRegex": ".*",
```

```
        "regex": "(\w+)(\s+)",
        "key" :["key1", "key2"],
        "filterKey":["key1"],
        "filterRegex":["regex1"],
        "topicFormat": "none"
    },
    "outputType": "LogService",
    "outputDetail":
    {
        "logstoreName": "perfcounter"
    }
}
```

**请求参数**

**说明:**

**请求参数和各种模式的Logtail配置样例请参考#unique_15。**

**请求头**

**无特有请求头。关于 Log Service API 的公共请求头，请参考 公共请求头。**

**响应头**

**无特有响应头。请参考 公共响应头。**

**响应元素**

**返回值：成功返回 200 状态码。**

**错误码**

**除了返回 Log Service API 的 通用错误码，还可能返回如下特有错误码：**

| HTTP 状态码 | ErrorCode | ErrorMessage |
| --- | --- | --- |
| 404 | ConfigNotExist | config {Configname} does not exist |
| 400 | InvalidParameter | invalid config resource json |
| 400 | BadRequest | config resource configname does not match with request |
| 500 | InternalServerError | internal server error |

**细节描述**

**创建过程中遇到格式错误、必要参数遗失、quota 超过限制等错误，则创建失败。**

## 示例

**请求示例：**

```
PUT /configs/logtail-config-sample
Header :
{
    "Content-Length": 737,
    "Host": "ali-test-project.cn-hangzhou-devcommon-intranet.sls.
aliyuncs.com",
    "x-log-bodyrawsize": 737,
    "Content-MD5": "431263EB105D584A5555762A81E869C0",
    "x-log-signaturemethod": "hmac-sha1",
    "Date": "Mon, 09 Nov 2015 09:14:32 GMT",
    "x-log-apiversion": "0.6.0",
    "User-Agent": "log-python-sdk-v-0.6.0",
    "Content-Type": "application/json",
    "Authorization": "LOG <yourAccessKeyId>:<yourSignature>"
}
Body :
{
    "outputDetail": {
        "logstoreName": "sls-test-logstore"
    },
    "inputType": "file",
    "inputDetail": {
        "regex": "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+)
 ([^\"]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \"([^\"]*)\" \"([^
\"]*)\".*",
        "filterKey": [],
        "logPath": "/var/log/nginx/",
        "logBeginRegex": "\\d+\\.\\d+\\.\\d+\\.\\d+ - .*",
        "logType": "common_reg_log",
        "topicFormat": "none",
        "localStorage": true,
        "key": [
            "ip",
            "time",
            "method",
            "url",
            "request_time",
            "request_length",
            "status",
            "length",
            "ref_url",
            "browser"
        ],
        "filePattern": "access*.log",
        "timeFormat": "%d/%b/%Y:%H:%M:%S",
        "filterRegex": []
    },
    "outputType": "LogService",
    "configName": "logtail-config-sample"
}
```

**响应示例：**

```
{
    "date": "Mon, 09 Nov 2015 09:14:32 GMT",
    "connection": "close",
    "x-log-requestid": "564063F899248CAA2300B778",
    "content-length": "0",
```

```
    "server": "nginx/1.6.1"
}
```

# 3.12 RAM/STS

## 3.12.1 概览

借助RAM实现子账号对主账号的 Log Service 资源的访问

您创建的Project、Logstore、Config、MachineGroup，都是您自己拥有的资源。默认情况下，您对自己的资源拥有完整的操作权限，可以使用本文档中列举的所有 API 对资源进行操作。

但在主子账号的场景下，子账号刚创建时是没有资格去操作主账号的资源的。需要通过RAM授权的方式，给予子账号操作主账号资源的权限。

**说明:**

在了解如何使用 RAM 来授权和访问 Log Service 资源之前，请确保您已详细阅读了 #unique_102 和 RAM简介。

RAM和日志服务相关的授权策略：

- AliyunLogFullAccess

  给子账号授予该权限，那么子账号将对主账号拥有的日志服务的资源有完全的访问权限。授权策略描述如下：

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": "log:*",
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

- AliyunLogReadOnlyAccess

  给子账号授予该权限，那么子账号将对主账号拥有的日志服务的资源有只读的访问权限。授权策略描述如下：

```
{
   "Version": "1",
   "Statement": [
     {
       "Action": [
         "log:Get*",
         "log:List*"
       ],
       "Resource": "*",
```

```
        "Effect": "Allow"
      }
    ]
  }
```

- 向指定日志库（Logstore）上传数据

  给子账号授予该权限，那么子账号将可以通过API/SDK直接向指定日志库上传数据，授权策略描述如下：

```
  {
    "Version": "1",
    "Statement": [
      {
        "Action": [
          "log:Post*",
          "log:BatchPost*"
        ],
        "Resource": ["acs:log:*:*:project/<指定的 proejct 名称>/
logstore/<指定的 logstore 名称>"],
        "Effect": "Allow"
      }
    ]
  }
```

- 控制台查询指定日志库（Logstore）数据

  给子账号授予该权限，那么子账号在登录控制台后将对主账号拥有指定日志库资源只读的访问权限（查询日志、拖取日志、查看日志库列表）。授权策略描述如下：

```
  {
    "Version": "1",
    "Statement": [
      {
        "Action": ["log:List*"],
        "Resource": ["acs:log:*:*:project/<指定的 proejct 名称>/*"],
        "Effect": "Allow"
      },
      {
        "Action": ["log:Get*"],
        "Resource": ["acs:log:*:*:project/<指定的 proejct 名称>/
logstore/<指定的 logstore 名称>"],
        "Effect": "Allow"
      }
    ]
  }
```

如果您不需要跨账户进行 Log Service 资源的授权和访问，您可以跳过此章节。跳过这些部分并不影响您对文档中其余部分的理解和使用。

更多信息：

- RAM 中可授权的 Log Service 资源类型
- RAM 中可对 Log Service 资源进行授权的 Action
- Log Service API 发生子账号访问主账号资源时的鉴权规则

## 3.12.2 资源列表

**RAM 中可授权的 Log Service 资源类型**

**目前，可以在 RAM 中进行授权的资源类型及描述方式如下表所示：**

| 资源类型 | 授权策略中的资源描述方式 |
| --- | --- |
| Project/Logstore | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName} |
| Project/Logstore/Shipper | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/shipper/${shipperName} |
| | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/* |
| Project/Config | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailconfig} |
| | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/* |
| Project/MachineGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |
| | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/* |
| Project/ConsumerGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/${consumerGroupName} |
| | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}/consumergroup/* |
| Project/SavedSearch | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/savedsearch/${savedSearchName} |

| 资源类型 | 授权策略中的资源描述方式 |
| --- | --- |
| | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/savedsearch/* |
| Project/Dashboard | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/dashboard/${dashboardName} |
| | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/dashboard/* |
| Project/Alarm | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/alert/${alarmName} |
| | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/alert/* |
| 泛指模式 | acs:log:${regionName}:${projectOwnerAliUid}:* |
| | acs:log:*:${projectOwnerAliUid}:* |

**说明:**

Log Service 中的资源有层级关系，project 是顶级资源，logstore、config、machinegroup 是 project 的子资源，相互之间是平级资源，shipper 和 consumergroup 是 logstore 的子资源。

各个参数说明请见下表。

表 3-3:

| 参数 | 说明 |
| --- | --- |
| `${regionName}` | 某个 region 的名称。 |
| `${projectOwnerAliUid}` | 用户的阿里云账号 ID。 |
| `${projectName}` | Log Service 项目的名称。 |
| `${logstoreName}` | Logstore的名称。 |
| `${logtailconfig}` | config 的名称。 |
| `${machineGroupName}` | 机器组的名称。 |
| `${shipperName}` | 日志投递规则的名称。 |

| 参数 | 说明 |
| --- | --- |
| ${consumerGroupName} | 协同消费组的名称。 |
| ${savedSearchName} | 快速查询的名称。 |
| ${dashboardName} | 仪表盘的名称。 |
| ${alarmName} | 报警规则的名称。 |

## 3.12.3 动作列表

本文档主要介绍RAM中对日志服务资源进行授权的资源列表。

RAM 中可对 Log Service 资源进行授权的 Action

在 RAM 中，可以对一个 Log Service 资源进行以下 Action 的授权，每一个 Action 都和某一个或者两个 API 对应。

| 动作（Action） | 说明 |
| --- | --- |
| log:CreateProject | 创建Project。 |
| log:GetLogStoreLogs | 查询指定 Project 下某个 Logstore 中的日志数据。 |
| log:GetLogStoreHistogram | 查询指定的 Project 下某个 Logstore 中日志的分布情况。 |
| log:GetLogStore | 查看 Logstore 属性。 |
| log:ListLogStores | 列出指定 Project 下的所有 Logstore 的名称。 |
| log:CreateLogStore | 在 Project 下创建 Logstore。 |
| log:DeleteLogStore | 删除 Logstore，包括所有 Shard 数据，以及索引等。 |
| log:UpdateLogStore | 更新 Logstore 的属性。 |
| log:GetCursorOrData（GetCursor，PullLogs） | 根据时间获得游标（cursor）；根据游标、数量获得日志。 |
| log:ListShards | 列出 Logstore 下当前所有可用 Shard。 |
| log:PostLogStoreLogs | 向指定的 LogStore 写入日志数据。 |
| log:CreateConfig | 在Project下创建日志配置。 |
| log:UpdateConfig | 更新配置内容。 |
| log:DeleteConfig | 删除指定配置。 |
| log:GetConfig | 获得一个配置的详细信息。 |

| 动作（Action） | 说明 |
| --- | --- |
| log:ListConfig | 列出 Project 下所有配置信息，可以通过参数进行翻页。 |
| log:CreateMachineGroup | 根据需求创建一组机器，用以日志收集下发配置。 |
| log:UpdateMachineGroup | 更新机器组信息。 |
| log:DeleteMachineGroup | 删除机器组。 |
| log:GetMachineGroup | 查看具体的机器组信息。 |
| log:ListMachineGroup | 查看机器组名称列表。 |
| log:ListMachines | 获得机器组下属于用户、并与Server 端连接的机器状态信息。 |
| log:ApplyConfigToGroup | 将配置应用到机器组。 |
| log:RemoveConfigFromGroup | 从机器组中删除配置。 |
| log:GetAppliedMachineGroups | 获得机器组上已经被应用的机器列表。 |
| log:GetAppliedConfigs | 获得机器组上已经被应用的配置名称。 |
| log:GetShipperStatus | 查询日志投递任务状态。 |
| log:RetryShipperTask | 重新执行失败的日志投递任务。 |
| log:CreateConsumerGroup | 在指定的 Logstore 上创建一个消费组。 |
| log:UpdateConsumerGroup | 修改指定 Consumer Group 属性。 |
| log:DeleteConsumerGroup | 删除一个指定的 Consumer Group。 |
| log:ListConsumerGroup | 查询指定 Logstore 的所有消费组。 |
| log:ConsumerGroupUpdateCheckPoint | 更新指定Project和Logstore下的Consumer Group的某个Shard的checkpoint。 |
| log:ConsumerGroupHeartBeat | 为指定消费者发送心跳到服务端。 |
| log:GetConsumerGroupCheckPoint | 获取指定消费组的消费的某个或者所有 Shard 的checkpoint。 |
| log:CreateIndex | 为指定Logstore创建索引。 |
| log:DeleteIndex | 删除指定Logstore的索引。 |
| log:GetIndex | 查询指定Logstore的索引。 |
| log:UpdateIndex | 更新指定Logstore的索引。 |
| log:CreateSavedSearch | 创建快速查询。 |
| log:UpdateSavedSearch | 更新快速查询。 |

| 动作（Action） | 说明 |
| --- | --- |
| log:GetSavedSearch | 查看指定快速查询。 |
| log:DeleteSavedSearch | 删除快速查询。 |
| log:ListSavedSearch | 查看快速查询列表。 |
| log:CreateDashboard | 创建仪表盘。 |
| log:UpdateDashboard | 更新仪表盘。 |
| log:GetDashboard | 查看指定仪表盘。 |
| log:DeleteDashboard | 删除仪表盘。 |
| log:ListDashboard | 查看仪表盘列表。 |

## 3.12.4 鉴权规则

Log Service API 发生子账号访问主账号资源时的鉴权规则

当子账号通过 Log Service Open API 对主账号的资源进行访问时，Log Service 后台向 RAM 进行权限检查，以确保资源拥有者的确将相关资源的相关权限授予了调用者。

每个不同的 Log Service API 会根据涉及到的资源以及 API 的语义来确定需要检查哪些资源的权限。具体地，各类 API 的鉴权规则见下表。

logstore

| Action | Resource |
| --- | --- |
| log:GetLogStore | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName} |
| log:ListLogStores | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/* |
| log:CreateLogStore | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/* |
| log:DeleteLogStore | `acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}` |
| log:UpdateLogStore | `acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName}` |

### loghub

**数据写入以及消费类 API，其中获取数据游标 API GetCursor 以及获取数据 API GetLogs 共用同一个 Action（log:GetCursorOrData）。**

| Action | Resource |
| --- | --- |
| log:GetCursorOrData | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName} |
| log:ListShards | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName} |
| log:PostLogStoreLogs | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logstore/${logstoreName} |

### config

| Action | Resource |
| --- | --- |
| log:CreateConfig | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/* |
| log:UpdateConfig | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailConfigName} |
| log:DeleteConfig | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailConfigName} |
| log:GetConfig | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailConfigName} |
| log:ListConfig | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/* |

### machinegroup

| Actions | Resources |
| --- | --- |
| log:CreateMachineGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/* |
| log:UpdateMachineGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |
| log:DeleteMachineGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |
| log:GetMachineGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |
| log:ListMachineGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/* |
| log:ListMachines | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |

config 和 machinegroup 交互类 API

| Actions | Resources |
| --- | --- |
| log:ApplyConfigToGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailConfigName}<br>acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |
| log:RemoveConfigFromGroup | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailConfigName}<br>acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |
| log:GetAppliedMachineGroups | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/logtailconfig/${logtailConfigName} |

| Actions | Resources |
| --- | --- |
| log:GetAppliedConfigs | acs:log:${regionName}:${projectOwnerAliUid}:project/${projectName}/machinegroup/${machineGroupName} |

# 4 获取告警数据

## 4.1 DescribeAlarmEventList

获取态势感知安全告警模块的安全事件的列表。

告警事件分为告警与异常两个维度，一个告警事件包含多个异常事件。该API可以获取告警事件的列表。

**调试**

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeAlarmEventList | 系统规定参数。取值：DescribeAlarmEventList。 |
| CurrentPage | Integer | 是 | 1 | 告警事件列表的页码。起始值为1，默认值为1。 |
| From | String | 是 | sas | 请求来源标识，固定为sas。 |
| PageSize | String | 是 | 20 | 分页查询时设置的每页行数。默认值为20。 |
| AlarmEventName | String | 否 | DDOS木马 | 告警事件名称。 |
| AlarmEventType | String | 否 | 恶意进程（云查杀） | 告警事件类型。 |
| Dealed | String | 否 | Y | 告警事件状态。<br>· N：待处理告警<br>· Y：已处理告警 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Lang | String | 否 | zh | 告警的语言类型。<br>· zh：中文<br>· cn：英文 |
| Levels | String | 否 | serious | 告警事件的危险等级，多个严重等级用逗号分隔（严重等级递减）。<br>· serious：紧急<br>· suspicious：可疑<br>· remind：提醒 |
| Remark | String | 否 | database_server | 告警名称/资产信息。 |
| SourceIp | String | 否 | 1.1.1.1 | 请求源IP。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 分页信息。 |
| Count | Integer | 1 | 告警事件条数。 |
| CurrentPage | Integer | 1 | 告警事件列表的页码。 |
| PageSize | Integer | 20 | 分页大小。 |
| TotalCount | Integer | 1 | 总条数。 |
| RequestId | String | 28267723-D857-4DD8-B295-013100000000 | 请求ID。 |
| SuspEvents | | | 告警事件的列表。 |
| AlarmEventName | String | Linux计划任务执行异常指令 | 告警事件名称。 |
| AlarmEventType | String | 进程异常行为 | 告警事件类型。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| AlarmUniqueInfo | String | 8df914418f4211fbf756efe7a6f40cbc | 告警事件的唯一标识。 |
| CanBeDealOnLine | Boolean | true | 是否能在线处理（隔离）。 |
| CanCancelFault | Boolean | false | 能否取消标记为误报。 |
| DataSource | String | aegis_suspicious_event | 数据来源。 |
| Description | String | 黑客入侵服务器后，为了让恶意后门程序能持久化运行，黑客常常将恶意SHELL脚本写入crontab、systemd等计划任务。 | 告警事件的描述。 |
| EndTime | Long | 1543740301000 | 告警事件结束时间。 |
| InstanceName | String | 测试服务器 | 关联实例的名称。 |
| InternetIp | String | 10.1.1.1 | 关联实例的公网IP。 |
| IntranetIp | String | 10.1.1.1 | 关联实例的私网IP。 |
| Level | String | serious | 告警事件的危险等级。<br>· serious：紧急<br>· suspicious：可疑<br>· remind：提醒 |
| SaleVersion | String | 4 | 需要的售卖版本。<br>· 0：基础版本<br>· 1：企业版本 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Solution | String | 请及时排查告警中提示的恶意URL，以及所下载的目录下的恶意文件。并及时清理已运行的恶意进程。如果该指令是您自己主动执行，您可以在控制台点击标记为误报，并通过工单方式反馈给我们的安全工程师。 | 告警事件的处理方法。 |
| StartTime | Long | 1543740301000 | 告警事件的开始时间。 |
| SuspiciousEventCount | Integer | 1 | 关联的异常事件的条数。 |
| Uuid | String | 47900178-885d-4fa4-9d77-XXXXXXXXXXXX | 关联实例的唯一标识。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeAlarmEventList
&CurrentPage=1
&From=sas
&PageSize=20
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribeAlarmEventListResponse>
    <RequestId>B5446AFA-58B6-41DC-80E6-E0382AC5A1F4</RequestId>
    <PageInfo>
        <Count>1</Count>
        <TotalCount>1</TotalCount>
        <PageSize>10</PageSize>
        <CurrentPage>1</CurrentPage>
    </PageInfo>
    <SuspEvents>
        <Uuid>47900178-885d-4fa4-9d77-XXXXXXXXXXXX</Uuid>
        <Description>黑客入侵服务器后，为了让恶意后门程序能持久化运行，黑客常常将恶意SHELL脚本写入crontab、systemd等计划任务。</Description>
        <CanCancelFault>false</CanCancelFault>
        <InternetIp>10.0.0.10</InternetIp>
```

```
      <SuspiciousEventCount>1</SuspiciousEventCount>
      <AlarmUniqueInfo>8df914418f4211fbf756efe7a6f40cbc</AlarmUniqu
eInfo>
      <AlarmEventName>Linux计划任务执行异常指令</AlarmEventName>
      <AlarmEventType>进程异常行为</AlarmEventType>
      <IntranetIp>10.0.0.10</IntranetIp>
      <Level>serious</Level>
      <EndTime>1543740301000</EndTime>
      <StartTime>1543740301000</StartTime>
      <SaleVersion>1</SaleVersion>
      <CanBeDealOnLine>false</CanBeDealOnLine>
      <InstanceName>server01</InstanceName>
   </SuspEvents>
        </DescribeAlarmEventListResponse>
```

JSON 格式

```
{
 "RequestId":"B5446AFA-58B6-41DC-80E6-E0382AC5A1F4",
 "SuspEvents":[
  {
   "Uuid":"47900178-885d-4fa4-9d77-XXXXXXXXXXXX",
   "Description":"黑客入侵服务器后，为了让恶意后门程序能持久化运行，黑客常常将恶
意SHELL脚本写入crontab、systemd等计划任务。",
   "CanCancelFault":false,
   "InternetIp":"10.0.0.10",
   "SuspiciousEventCount":1,
   "AlarmUniqueInfo":"8df914418f4211fbf756efe7a6f40cbc",
   "AlarmEventName":"Linux计划任务执行异常指令",
   "AlarmEventType":"进程异常行为",
   "IntranetIp":"10.0.0.10",
   "Level":"serious",
   "EndTime":1543740301000,
   "StartTime":1543740301000,
   "CanBeDealOnLine":false,
   "SaleVersion":"1",
   "InstanceName":"server01"
  }
 ],
 "PageInfo":{
  "Count":1,
  "TotalCount":1,
  "PageSize":10,
  "CurrentPage":1
 }
}
```

**错误码**

访问错误中心查看更多错误码。

## 4.2 DescribeAlarmEventDetail

获取告警事件的详细信息。

告警事件分为告警与异常两个维度，一个告警事件包含多个异常事件。该API可以获取一个告警事件的详情。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeAlarmEventDetail | 系统规定参数。取值：DescribeAlarmEventDetail。 |
| AlarmUniqueInfo | String | 是 | 8df914418f4211fbf756efe7a6f40cbc | 告警事件的唯一标识。 |
| From | String | 是 | sas | 请求来源标识，固定为sas。 |
| Lang | String | 否 | zh | 告警事件显示的语言类型。<br>· zh：中文<br>· cn：英文 |
| SourceIp | String | 否 | 1.1.1.1 | 请求源IP。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Data | | | 告警事件详情。 |
| AlarmEventAliasName | String | 进程异常行为-Linux计划任务执行异常指令 | 告警事件完整名称。 |
| AlarmEventDesc | String | 黑客入侵服务器后，为了让恶意后门程序能持久化运行，黑客常常将恶意SHELL脚本写入crontab、systemd等计划任务。 | 告警事件描述。 |
| AlarmUniqueInfo | String | 8df914418f4211fbf756efe700000000 | 告警事件的唯一标识。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CanBeDealOnLine | Boolean | false | 是否能在线处理（隔离）。 |
| CanCancelFault | Boolean | false | 能否取消标记为误报。 |
| CauseDetails | | | 告警事件发生的原因（溯源信息）。 |
| Key | String | item | 文本的展示方式：<br>· text：文本方式<br>· html：富文本方式 |
| Value | | | 溯源信息字段值。 |
| Name | String | 排查方案 | 溯源信息字段的key。 |
| Type | String | html | 溯源信息字段展示的类型。 |
| Value | String | 请根据上述信息排查您的WEB服务被利用的页面及参数是否存在漏洞，并及时修复。 | 溯源信息字段的值。 |
| DataSource | String | aegis_suspicious_event | 数据来源。 |
| EndTime | Long | 1542366542000 | 告警事件结束时间。 |
| InstanceName | String | 测试服务器 | 关联实例的名称。 |
| InternetIp | String | 10.0.0.0 | 关联实例的公网IP。 |
| IntranetIp | String | 10.0.0.0 | 关联实例的私网IP。 |
| Level | String | serious | 告警事件的危险等级。<br>· serious：紧急<br>· suspicious：可疑<br>· remind：提醒 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Solution | String | 请及时排查告警中提示的恶意URL，以及所下载的目录下的恶意文件。并及时清理已运行的恶意进程。如果该指令是您自己主动执行，您可以在控制台点击标记为误报，并通过工单方式反馈给我们的安全工程师。 | 告警事件的处理方法。 |
| StartTime | Long | 1542378601000 | 告警事件的开始时间。 |
| Type | String | 异常网络连接 | 事件类型。 |
| Uuid | String | 47900178-885d-4fa4-9d77-XXXXXXXXXXXX | 关联实例的唯一标识。 |
| RequestId | String | 5A1DDB3C-798C-4A84-BF6E-3DC700000000 | 请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeAlarmEventDetail
&AlarmUniqueInfo=8df914418f4211fbf756efe7a6f40cbc
&From=sas
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribeAlarmEventDetailResponse>
   <Data>
      <Uuid>47900178-885d-4fa4-9d77-XXXXXXXXXXXX</Uuid>
      <AlarmEventAliasName>进程异常行为-Linux计划任务执行异常指令</
AlarmEventAliasName>
      <Type>进程异常行为</Type>
      <InternetIp>10.0.0.10</InternetIp>
      <AlarmEventDesc>黑客入侵服务器后，为了让恶意后门程序能持久化运行，黑客常常
将恶意SHELL脚本写入crontab、systemd等计划任务。</AlarmEventDesc>
      <IntranetIp>10.0.0.0</IntranetIp>
```

```
       <CauseDetails>
          <Value>
             <Type>text</Type>
             <Value>黑客登录ECS后通过编辑文件方式 写WEBSHELL</Value>
             <Name>入侵原因</Name>
          </Value>
          <Value>
             <Type>text</Type>
             <Value>2018-11-16 19:09:02</Value>
             <Name>攻击时间</Name>
          </Value>
          <Value>
             <Type>text</Type>
             <Value>N/A</Value>
             <Name>攻击源IP</Name>
          </Value>
          <Value>
             <Type>text</Type>
             <Value>N/A</Value>
             <Name>漏洞攻击载荷</Name>
          </Value>
          <Value>
             <Type>text</Type>
             <Value>请根据上述信息排查您的WEB服务被利用的页面及参数是否存在漏
洞，并及时修复。</Value>
             <Name>排查方案</Name>
          </Value>
          <Key>item</Key>
       </CauseDetails>
       <Level>serious</Level>
       <EndTime>1543741201000</EndTime>
       <StartTime>1543312803000</StartTime>
       <CanBeDealOnLine>false</CanBeDealOnLine>
       <InstanceName>server01</InstanceName>
    </Data>
    <RequestId>5A1DDB3C-798C-4A84-BF6E-3DC7F7D7EB4A</RequestId>
     </DescribeAlarmEventDetailResponse>
```

JSON 格式

```
{
 "Data":{
  "Uuid":"47900178-885d-4fa4-9d77-XXXXXXXXXXXX",
  "AlarmEventDesc":"黑客入侵服务器后，为了让恶意后门程序能持久化运行，黑客常常将
恶意SHELL脚本写入crontab、systemd等计划任务。",
  "AlarmEventAliasName":"进程异常行为-Linux计划任务执行异常指令",
  "Type":"进程异常行为",
  "IntranetIp":"10.0.0.0",
  "CauseDetails":[
   {
    "Value":[
     {
      "Name":"入侵原因",
      "Value":"黑客登录ECS后通过编辑文件方式 写WEBSHELL",
      "Type":"text"
     },
     {
      "Name":"攻击时间",
      "Value":"2018-11-16 19:09:02",
      "Type":"text"
     },
     {
      "Name":"攻击源IP",
```

```
        "Value":"N/A",
        "Type":"text"
       },
       {
        "Name":"漏洞攻击载荷",
        "Value":"N/A",
        "Type":"text"
       },
       {
        "Name":"排查方案",
        "Value":"请根据上述信息排查您的WEB服务被利用的页面及参数是否存在漏洞，并及
时修复。",
        "Type":"text"
       }
      ],
      "Key":"item"
     }
    ],
    "InternetIp":"10.0.0.10",
    "EndTime":1543741201000,
    "Level":"serious",
    "StartTime":1543312803000,
    "CanBeDealOnLine":false,
    "InstanceName":"server01"
   },
   "RequestId":"5A1DDB3C-798C-4A84-BF6E-3DC7F7D7EB4A"
}
```


### 错误码

访问错误中心查看更多错误码。

## 4.3 DescribeSuspEvents

查询异常事件列表。

通过该API接口可查询异常事件列表。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeSuspEvents | 系统规定参数。取值：DescribeSuspEvents。 |
| From | String | 是 | sas | 请求来源标识，固定为sas。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| AlarmUniqueInfo | String | 否 | 8df914418f4211fbf756efe7a6f40cbc | 告警事件的唯一标识。 |
| CurrentPage | String | 否 | 1 | 当前页码。 |
| Dealed | String | 否 | N | 异常事件状态。<br>· N：待处理告警<br>· Y：已处理告警 |
| Lang | String | 否 | zh | 异常事件的语言类型。<br>· zh：中文<br>· cn：英文 |
| Levels | String | 否 | serious | 告警事件的危险等级，多个危险等级用逗号分隔。以下危险等级严重程度依次递减。<br>· serious：紧急<br>· suspicious：可疑<br>· remind：提醒 |
| Name | String | 否 | 矿 | 异常事件名称或者是主机名称，模糊匹配。 |
| PageSize | String | 否 | 20 | 分页查询时设置的每页行数，默认值为20。 |
| ParentEventTypes | String | 否 | 网站后门 | 异常事件分类名称。 |
| Remark | String | 否 | 测试机器 | 主机IP或者名称。 |
| SourceIp | String | 否 | 1.1.1.1 | 接口访问者源IP。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 1 | 当前返回结果数量。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CurrentPage | Integer | 1 | 当前页。 |
| PageSize | Integer | 20 | 分页大小。 |
| RequestId | String | 43F670F3-AB40-4E91-BC7D-C57400000000 | 请求ID。 |
| SuspEvents | | | 异常事件列表。 |
| AlarmEventName | String | Linux计划任务执行异常指令 | 告警事件名称。 |
| AlarmEventType | String | 进程异常行为 | 告警事件类型。 |
| AlarmUniqueInfo | String | 8df914418f4211fbf756efe700000000 | 告警事件的唯一标识。 |
| CanBeDealOnLine | Boolean | true | 能否在线处理（隔离）。 |
| DataSource | String | N/A | 数据来源（可忽略）。 |
| Desc | String | webshell | 影响概况描述。 |
| EventStatus | Integer | 1 | 异常事件的状态：<br>· PENDING(1, "待处理"),<br>· IGNORE(2, "已忽略"),<br>· HANDLED(4, "已确认"),<br>· FAULT(8, "已标记误报"),<br>· DEALING(16, "处理中"),<br>· DONE(32, "处理完毕"),<br>· EXPIRE(64, "已经过期"); |
| EventSubType | String | XorDDoS木马 | 异常事件名称。 |
| Id | Long | 1000 | 唯一标识。 |
| InstanceName | String | nginx | 关联实例的名称。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| InternetIp | String | 10.0.0.10 | 关联实例的公网IP。 |
| IntranetIp | String | 10.0.0.10 | 关联实例的私网IP。 |
| LastTime | String | 2018-09-26 01:51:01 | 事件发生时间。 |
| Level | String | serious | 告警事件的危险等级：<br>· serious：紧急<br>· suspicious：可疑<br>· remind：提醒 |
| Name | String | 恶意进程（云查杀）-XorDDoS木马 | 异常事件的完整名称。 |
| OccurrenceTime | String | 2018-09-26 01:51:01 | 首次发生时间。 |
| OperateMsg | String | success | 操作备注消息。 |
| SaleVersion | String | 1 | 需要的售卖版本：<br>· 0：基础版本<br>· 1：企业版本 |
| Uuid | String | bf6b30d3-eea8-4924-9f0a-XXXXXXXXXXXX | 关联实例唯一标识。 |
| TotalCount | Integer | 100 | 查询总数量。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeSuspEvents
&From=saas
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribeSuspEventsResponse>
```

```
    <TotalCount>3</TotalCount>
    <Count>2</Count>
    <PageSize>20</PageSize>
    <RequestId>0C7FAD74-83FA-4671-9250-A5F2A64F437A</RequestId>
    <CurrentPage>1</CurrentPage>
    <SuspEvents>
       <EventStatus>1</EventStatus>
       <SaleVersion>1</SaleVersion>
       <IntranetIp>10.0.0.0</IntranetIp>
       <EventSubType>XorDDoS木马</EventSubType>
       <Name>恶意进程（云查杀）-XorDDoS木马</Name>
       <DataSource>aegis_suspicious_event</DataSource>
       <OccurrenceTime>2018-09-26 01:51:01</OccurrenceTime>
       <InstanceName>server01</InstanceName>
       <Desc>XORDDoS木马入侵后，会在Linux的定时任务中植入恶意代码。</Desc>
       <CanBeDealOnLine>false</CanBeDealOnLine>
       <Uuid>bf6b30d3-eea8-4924-9f0a-XXXXXXXXXXXX</Uuid>
       <InternetIp>10.0.0.0</InternetIp>
       <Level>serious</Level>
       <Id>3682</Id>
       <LastTime>2018-10-24 21:06:01</LastTime>
    </SuspEvents>
    <SuspEvents>
       <EventStatus>1</EventStatus>
       <SaleVersion>1</SaleVersion>
       <IntranetIp>172.24.40.51</IntranetIp>
       <EventSubType>XorDDoS木马</EventSubType>
       <Name>恶意进程（云查杀）-XorDDoS木马</Name>
       <DataSource>aegis_suspicious_event</DataSource>
       <OccurrenceTime>2018-09-26 02:01:01</OccurrenceTime>
       <InstanceName>server01</InstanceName>
       <Desc>XORDDoS木马入侵后，会在Linux的定时任务中植入恶意代码。</Desc>
       <CanBeDealOnLine>false</CanBeDealOnLine>
       <Uuid>bf6b30d3-eea8-4924-9f0a-98461cb8ffeb</Uuid>
       <InternetIp>10.0.0.0</InternetIp>
       <Level>serious</Level>
       <Id>3683</Id>
       <LastTime>2018-10-24 21:01:01</LastTime>
    </SuspEvents>
         </DescribeSuspEventsResponse>
```

JSON 格式

```
{
 "Count":2,
 "TotalCount":3,
 "PageSize":20,
 "RequestId":"0C7FAD74-83FA-4671-9250-A5F2A64F437A",
 "SuspEvents":[
  {
   "Uuid":"bf6b30d3-eea8-4924-9f0a-XXXXXXXXXXXX",
   "EventStatus":1,
   "LastTime":"2018-10-24 21:06:01",
   "InternetIp":"10.0.0.0",
   "Name":"恶意进程（云查杀）-XorDDoS木马",
   "DataSource":"aegis_suspicious_event",
   "OccurrenceTime":"2018-09-26 01:51:01",
   "IntranetIp":"10.0.0.0",
   "Id":3682,
   "Level":"serious",
   "SaleVersion":"1",
   "CanBeDealOnLine":false,
   "InstanceName":"server01",
```

```
    "Desc":"XORDDoS木马入侵后，会在Linux的定时任务中植入恶意代码。",
    "EventSubType":"XorDDoS木马"
   },
   {
    "Uuid":"bf6b30d3-eea8-4924-9f0a-XXXXXXXXXXXX",
    "EventStatus":1,
    "LastTime":"2018-10-24 21:01:01",
    "InternetIp":"10.0.0.0",
    "Name":"恶意进程（云查杀）-XorDDoS木马",
    "DataSource":"aegis_suspicious_event",
    "OccurrenceTime":"2018-09-26 02:01:01",
    "IntranetIp":"172.24.40.51",
    "Id":3683,
    "Level":"serious",
    "SaleVersion":"1",
    "CanBeDealOnLine":false,
    "InstanceName":"server01",
    "Desc":"XORDDoS木马入侵后，会在Linux的定时任务中植入恶意代码。",
    "EventSubType":"XorDDoS木马"
   }
  ],
  "CurrentPage":1
}
```


### 错误码

访问错误中心查看更多错误码。

## 4.4 DescribeSuspEventDetail

获取异常事件详情。

告警事件分为告警与异常两个维度，一个告警事件包含多个异常事件。该API可以获取异常事件的详情。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeSuspEventDetail | 系统规定参数。取值：DescribeSuspEventDetail。 |
| From | String | 是 | sas | 请求来源，固定为sas。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Lang | String | 否 | zh | 异常事件的语言类型。<br>· zh：中文<br>· cn：英文 |
| SourceIp | String | 否 | 1.1.1.1 | 接口访问者源IP。 |
| SuspiciousEventId | Integer | 否 | 1 | 要查询的异常告警ID。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CanBeDealOnLine | Boolean | true | 是否支持在线处理（隔离）。 |
| DataSource | String | aegis_suspicious_file_v2 | 数据来源。 |
| Details | | | 异常事件的详情。 |
| InfoType | String | download_url | 图标展示的类型。 |
| Name | String | 更新时间 | 文案的标题。 |
| Type | String | html | 文案展示的方式。<br>· text：文本方式<br>· html：富文本的方式 |
| Value | String | 2018-12-12 12:00:00 | 文案的内容。 |
| EventDesc | String | 该文件极有可能是黑客成功入侵网站后种植的，建议您先确认文件合法性并处理。 | 异常事件描述。 |
| EventName | String | WEBSHELL | 异常事件的名称。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| EventStatus | String | 1 | 异常事件状态, 取值有：<br>· PENDING（1，待处理）<br>· IGNORE（2，已忽略）<br>· HANDLED（4，已确认）<br>· FAULT（8，已标记误报）<br>· DEALING（16，处理中）<br>· DONE（32，处理完毕）<br>· EXPIRE（64，已经过期） |
| EventTypeDesc | String | 网站后门-发现后门(Webshell)文件 | 异常事件类型说明。 |
| Id | Integer | 1991 | 异常事件ID。 |
| InstanceName | String | ca_cpm_test1 | 关联实例的名称。 |
| InternetIp | String | 10.0.0.0 | 关联实例的公网IP。 |
| IntranetIp | String | 10.0.0.10 | 关联实例的私网IP。 |
| LastTime | String | 2018-10-30 11:43:46 | 上次发生该事件的时间。 |
| Level | String | serious | 告警事件的危险等级（严重性依次递减）：<br>· serious：紧急<br>· suspicious：可疑<br>· remind：提醒 |
| OperateMsg | String | success | 操作的扩展信息。 |
| RequestId | String | 1 | 请求ID。 |
| SaleVersion | String | 17F3C8C2-0504-48D5-8B8F-9CF000000000 | 态势感知服务的版本信息。<br>· 0：基础版本<br>· 1：企业版本 |
| SasId | String | 1 | 态势感知系统ID。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Type | String | text | 区分是否为DDoS事件。 |
| Uuid | String | bffb12c3-590a-4db2-b538-XXXXXXXXXXXX | 关联实例的唯一标识。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeSuspEventDetail
&From=sas
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribeSuspEventDetailResponse>
    <RequestId>43F670F3-AB40-4E91-BC7D-C57468834F67</RequestId>
    <HostId>aegis.cn-hangzhou.aliyuncs.com</HostId>
    <Code>200</Code>
    <Message>
   illegal parameter, xxxx
  </Message>
    <EventDesc>该文件极有可能是黑客成功入侵网站后种植的，建议您先确认文件合法性并
处理。</EventDesc>
    <EventTypeDesc>网站后门-发现后门(Webshell)文件</EventTypeDesc>
    <EventStatus>1</EventStatus>
    <EventName>WEBSHELL</EventName>
    <SaleVersion>1</SaleVersion>
    <IntranetIp>10.0.0.0</IntranetIp>
    <DataSource>aegis_suspicious_file_v2</DataSource>
    <InstanceName>ca_cpm_test1</InstanceName>
    <Type>normal</Type>
    <CanBeDealOnLine>true</CanBeDealOnLine>
    <OperateMsg></OperateMsg>
    <Uuid>bffb12c3-590a-4db2-b538-XXXXXXXXXXXX</Uuid>
    <Details>
       <Type>text</Type>
       <Value>/data/ftpUser/pub/f12cd3bc5b484b0326309b48afb463fb</Value
>
       <InfoType>trojan_path</InfoType>
       <Name>木马文件路径</Name>
    </Details>
    <Details>
       <Type>text</Type>
       <Value>--</Value>
       <Name>影响域名</Name>
    </Details>
    <Details>
       <Type>text</Type>
       <Value>2018-10-30 05:00:56</Value>
       <InfoType>frist_found_time</InfoType>
       <Name>首次发现时间</Name>
```

```
    </Details>
    <Details>
       <Type>text</Type>
       <Value>2018-10-30 11:43:45</Value>
       <InfoType>update_time</InfoType>
       <Name>更新时间</Name>
    </Details>
    <Details>
       <Type>text</Type>
       <Value>Webshell</Value>
       <InfoType>trojan_type</InfoType>
       <Name>木马类型</Name>
    </Details>
    <Details>
       <Type>html</Type>
       <Value>&lt;a href="http://yundun-aegis-webshell-file.oss-cn-
shanghai.aliyuncs.com/XXXXXXXXXX?Expires=1540899863&amp;OSSAccessK
eyId=XXXXXX&amp;Signature=XXXXXX;response-content-disposition=
attachment%3Bfilename%3Df12cd3bc5b484b0326309b48afb463fb"&gt;下载&lt;/a
&gt;</Value>
       <InfoType>download_url</InfoType>
       <Name>源文件下载</Name>
    </Details>
    <InternetIp>39.105.41.176</InternetIp>
    <Level>serious</Level>
    <Id>129636</Id>
    <LastTime>2018-10-30 11:43:46</LastTime>
    <SasId>39938056</SasId>
          </DescribeSuspEventDetailResponse>
```

JSON 格式

```
{
 "Uuid":"bffb12c3-590a-4db2-b538-XXXXXXXXXXXX",
 "EventName":"WEBSHELL",
 "EventStatus":1,
 "Message":"illegal parameter, xxxx\n",
 "LastTime":"2018-10-30 11:43:46",
 "Details":[
  {
   "Name":"木马文件路径",
   "Value":"/data/ftpUser/pub/f12cd3bc5b484b0326309b48afb463fb",
   "Type":"text",
   "InfoType":"trojan_path"
  },
  {
   "Name":"影响域名",
   "Value":"--",
   "Type":"text"
  },
  {
   "Name":"首次发现时间",
   "Value":"2018-10-30 05:00:56",
   "Type":"text",
   "InfoType":"frist_found_time"
  },
  {
   "Name":"更新时间",
   "Value":"2018-10-30 11:43:45",
   "Type":"text",
   "InfoType":"update_time"
  },
  {
```

```
    "Name":"木马类型",
    "Value":"Webshell",
    "Type":"text",
    "InfoType":"trojan_type"
   },
   {
    "Name":"源文件下载",
    "Value":"<a href=\"http://yundun-aegis-webshell-file.oss-cn-
shanghai.aliyuncs.com/XXXXXXXXXX?Expires=1540899863&OSSAccessKeyId
=XXXXXX&Signature=XXXXXX&response-content-disposition=attachment%
3Bfilename%3Df12cd3bc5b484b0326309b48afb463fb\">下载</a>",
    "Type":"html",
    "InfoType":"download_url"
   }
  ],
  "Type":"normal",
  "InternetIp":"39.105.41.176",
  "HostId":"aegis.cn-hangzhou.aliyuncs.com",
  "EventTypeDesc":"网站后门-发现后门(Webshell)文件",
  "Code":"200",
  "DataSource":"aegis_suspicious_file_v2",
  "SasId":"39938056",
  "RequestId":"43F670F3-AB40-4E91-BC7D-C57468834F67",
  "IntranetIp":"10.0.0.0",
  "Id":129636,
  "Level":"serious",
  "EventDesc":"该文件极有可能是黑客成功入侵网站后种植的，建议您先确认文件合法性并
处理。",
  "OperateMsg":"",
  "CanBeDealOnLine":true,
  "SaleVersion":"1",
  "InstanceName":"ca_cpm_test1"
}
```


### 错误码

访问错误中心查看更多错误码。

# 5 云产品配置检查

## 5.1 DescribeRiskItemType

查看所有云产品配置检测项的类型。

每个检测项都对应一个类型，调用DescribeRiskItemType接口查看所有云产品配置检测项的类型。

检测项类型包含：

- 身份认证及权限
- 网络访问控制
- 日志审计
- 数据安全

**调试**

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeRiskItemType | 要执行的操作。<br>取值：DescribeRiskItemType。 |
| Lang | String | 否 | cn | 调用该接口返回的内容的语言种类。<br>支持中文（CN）和英文（EN）。 |
| SourceIp | String | 否 | 1.1.1.1 | 请求源IP。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| List | | | 云产品配置检测项类型的列。 |
| Id | Long | 1 | 云产品配置检测项的ID。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Title | String | 身份认证及权限 | 检测类型名称。如“身份认证及权限”。 |
| RequestId | String | 3B3F3A90-46A5-4023-A2D8-D68B14262F96 | 调用接口的请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeRiskItemType
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeRiskItemTypeResponse>
   <RequestId>3B3F3A90-46A5-4023-A2D8-D68B14262F96</RequestId>
   <List>
      <Title>zh-身份认证及权限</Title>
      <Id>1</Id>
   </List>
   <List>
      <Title>网络访问控制</Title>
      <Id>2</Id>
   </List>
   <List>
      <Title>日志审计</Title>
      <Id>3</Id>
   </List>
   <List>
      <Title>数据安全</Title>
      <Id>4</Id>
   </List>
   <List>
      <Title>基础安全防护</Title>
      <Id>6</Id>
   </List>
    </DescribeRiskItemTypeResponse>
```

#### JSON 格式

```
{
 "RequestId":"3B3F3A90-46A5-4023-A2D8-D68B14262F96",
 "List":[
  {
   "Id":1,
   "Title":"zh-身份认证及权限"
  },
  {
   "Id":2,
   "Title":"网络访问控制"
  },
```

```
    {
     "Id":3,
     "Title":"日志审计"
    },
    {
     "Id":4,
     "Title":"数据安全"
    },
    {
     "Id":6,
     "Title":"基础安全防护"
    }
   ]
}
```

### 错误码

访问错误中心查看更多错误码。

## 5.2 StartBaselineSecurityCheck

调用本接口执行基线检测任务。

执行检测任务，进行全量检测或对某个检查项进行检测或验证。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | StartBaselineSecurityCheck | 要执行的操作。<br>取值：StartBaselineSecurityCheck。 |
| SourceIp | String | 否 | 1.1.1.1 | 访问者源IP。 |
| Lang | String | 否 | cn | 调用参数返回的内容的语言种类。支持中文（CN）和英文（EN）。 |
| ItemIds.N | RepeatList | 否 | 1 | 检查项ID。 |
| Assets.N | RepeatList | 否 | 1 | 资产ID。 |
| Type | String | 否 | check | 任务类型。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 48D2E9A9-A1B0-4295-B727-0995757C47E9 | 请求id |

示例

请求示例

```
http(s)://[Endpoint]/?Action=StartBaselineSecurityCheck
&Type=check
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<StartBaselineSecurityCheckResponse>
   <RequestId>48D2E9A9-A1B0-4295-B727-0995757C47E9</RequestId>
</StartBaselineSecurityCheckResponse>
```

`JSON` 格式

```
{
 "RequestId":"48D2E9A9-A1B0-4295-B727-0995757C47E9"
}
```

错误码

访问错误中心查看更多错误码。

## 5.3 DescribeRiskCheckSummary

查看云产品检测结果汇总。

调用DescribeRiskCheckSummary接口查看云产品检测结果汇总，包括检测到的风险项数量、风险率、影响资产数量、上次检测时间以及检测项各个类型的统计数据。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeRiskCheckSummary | 要执行的操作。<br>取值：DescribeRiskCheckSummary。 |
| Lang | String | 否 | cn | 调用接口返回的内容的语种类型，支持中文和英文。 |
| SourceIp | String | 否 | 1.1.1.1 | 请求源IP。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 291B49F9-1685-4005-9D34-606B6F78740F | 调用接口的请求ID。 |
| RiskCheckSummary | | | 云产品检测的结果统计。 |
| AffectedAssetCount | Integer | 0 | 检测结果中风险项影响的资产数量。 |
| DisabledRiskCount | Integer | 0 | 检查不通过的检查项数量。 |
| EnabledRiskCount | Integer | 3 | 检查通过的检查项数量。 |
| Groups | | | 检查项类型的统计信息。 |
| CountByStatus | | | 检查项结果统计。 |
| Count | Integer | 2 | 检测到的风险项数量。 |
| Status | String | pass | 完成检测后，检测项的状态。<br>取值：<br>· pass：检测通过，表示检测项正常。<br>· failed：检测不通过，表示检测项存在风险。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Id | Long | 1 | 检查项类别ID。 |
| RemainingTime | Integer | 0 | 预计检测时间。 |
| Sort | Integer | 1 | 检测项类型在控制台全部类型下拉列表中的排列顺序。 |
| Status | String | finish | 检测完成的状态。<br>取值：<br>· finish：检测已完成。<br>· running：检测中。<br>· waiting：检测等待中。<br>· notStart：检测未开始。 |
| Title | String | 身份认证及权限 | 检测项类别的名称。 |
| ItemCount | Integer | 4 | 检查项的数量。 |
| PreviousCount | Integer | 0 | 上次检测到的风险项数量。 |
| PreviousTime | Long | 1545012926000 | 上次检测时间。 |
| RiskCount | Integer | 1 | 检测到的风险项数量。 |
| RiskLevelCount | | | 检测项每类危险等级的数量。<br>危险等级包含：<br>· 高危<br>· 中危<br>· 低危 |
| Count | Integer | 1 | 数量 |
| Key | String | medium | 检测项的危险等级。 |
| RiskRate | Float | 0.25 | 检测出的风险项数量在检测项总数中的占比。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeRiskCheckSummary
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeRiskCheckSummaryResponse>
   <RequestId>291B49F9-1685-4005-9D34-606B6F78740F</RequestId>
   <RiskCheckSummary>
      <RiskCount>1</RiskCount>
      <RiskRate>0.25</RiskRate>
      <PreviousTime>1545012926000</PreviousTime>
      <ItemCount>4</ItemCount>
      <AffectedAssetCount>0</AffectedAssetCount>
      <RiskLevelCount>
         <Count>1</Count>
         <Key>medium</Key>
      </RiskLevelCount>
      <Groups>
         <Status>finish</Status>
         <CountByStatus>
            <Status>exception</Status>
            <Count>2</Count>
         </CountByStatus>
         <RemainingTime>0</RemainingTime>
         <Sort>1</Sort>
         <Title>zh-身份认证及权限</Title>
         <Id>1</Id>
      </Groups>
      <Groups>
         <Status>finish</Status>
         <CountByStatus>
            <Status>exception</Status>
            <Count>5</Count>
         </CountByStatus>
         <RemainingTime>0</RemainingTime>
         <Sort>2</Sort>
         <Title>网络访问控制</Title>
         <Id>2</Id>
      </Groups>
      <Groups>
         <Status>finish</Status>
         <CountByStatus>
            <Status>pass</Status>
            <Count>1</Count>
         </CountByStatus>
         <RemainingTime>0</RemainingTime>
         <Sort>3</Sort>
         <Title>日志审计</Title>
         <Id>3</Id>
      </Groups>
      <Groups>
         <Status>finish</Status>
         <CountByStatus>
            <Status>exception</Status>
```

```
            <Count>2</Count>
         </CountByStatus>
         <RemainingTime>0</RemainingTime>
         <Sort>4</Sort>
         <Title>数据安全</Title>
         <Id>4</Id>
      </Groups>
      <Groups>
         <Status>finish</Status>
         <CountByStatus>
            <Status>exception</Status>
            <Count>2</Count>
         </CountByStatus>
         <RemainingTime>0</RemainingTime>
         <Sort>6</Sort>
         <Title>基础安全防护</Title>
         <Id>6</Id>
      </Groups>
      <PreviousCount>0</PreviousCount>
   </RiskCheckSummary>
    </DescribeRiskCheckSummaryResponse>
```

JSON 格式

```
{
 "RequestId":"291B49F9-1685-4005-9D34-606B6F78740F",
 "RiskCheckSummary":{
  "Groups":[
   {
    "Sort":1,
    "Status":"finish",
    "CountByStatus":[
     {
      "Status":"exception",
      "Count":2
     }
    ],
    "Id":1,
    "Title":"zh-身份认证及权限",
    "RemainingTime":0
   },
   {
    "Sort":2,
    "Status":"finish",
    "CountByStatus":[
     {
      "Status":"exception",
      "Count":5
     }
    ],
    "Id":2,
    "Title":"网络访问控制",
    "RemainingTime":0
   },
   {
    "Sort":3,
    "Status":"finish",
    "CountByStatus":[
     {
      "Status":"pass",
      "Count":1
     }
    ],
```

```
      "Id":3,
      "Title":"日志审计",
      "RemainingTime":0
     },
     {
      "Sort":4,
      "Status":"finish",
      "CountByStatus":[
       {
        "Status":"exception",
        "Count":2
       }
      ],
      "Id":4,
      "Title":"数据安全",
      "RemainingTime":0
     },
     {
      "Sort":6,
      "Status":"finish",
      "CountByStatus":[
       {
        "Status":"exception",
        "Count":2
       }
      ],
      "Id":6,
      "Title":"基础安全防护",
      "RemainingTime":0
     }
    ],
    "PreviousCount":0,
    "RiskCount":1,
    "ItemCount":4,
    "PreviousTime":1545012926000,
    "AffectedAssetCount":0,
    "RiskRate":0.25,
    "RiskLevelCount":[
     {
      "Key":"medium",
      "Count":1
     }
    ]
   }
}
```


**错误码**

访问错误中心查看更多错误码。

## 5.4 ModifyRiskCheckStatus

修改检查项的检测结果状态。

调用ModifyRiskCheckStatus可修改检查项检测结果状态，进行忽略或标记误报。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | ModifyRiskCheckStatus | 系统规定参数。取值：ModifyRiskCheckStatus。 |
| ItemId | Long | 否 | 1 | 检查项id |
| Lang | String | 否 | cn | 语种 |
| SourceIp | String | 否 | 1.1.1.1 | 请求源ip |
| Status | String | 否 | ignored | 新状态 |
| TaskId | Long | 否 | 57 | 任务id |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 48D2E9A9-A1B0-4295-B727-0995757C47E9 | 请求id |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=ModifyRiskCheckStatus
&ItemId=1
&TaskId=57
&Status=ignored
&<公共请求参数>
```

#### 正常返回示例

#### JSON 格式

```
{
 "RequestId":"48D2E9A9-A1B0-4295-B727-0995757C47E9"
```

```
}
```

### 错误码

访问错误中心查看更多错误码。

## 5.5 DescribeRiskCheckResult

查询检查项的检测结果。

调用DescribeRiskCheckResult可查询检查项的检测结果，可按类别或名称进行筛选。

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeRiskCheckResult | 系统规定参数。取值：DescribeRiskCheckResult。 |
| CurrentPage | Integer | 否 | 1 | 当前页 |
| GroupId | Long | 否 | 2 | 检测项类别id |
| Lang | String | 否 | cn | 语种 |
| Name | String | 否 | 云平台-主账号双因素认证配置检查 | 检查项名称 |
| PageSize | Integer | 否 | 10 | 每页行数 |
| RiskLevel | String | 否 | high | 风险级别 |
| SourceIp | String | 否 | 1.1.1.1 | 请求源ip |

### 返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 10 | 数量 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CurrentPage | Integer | 1 | 当前页 |
| List | | | 检查项列表 |
| └AffectedCount | Integer | 0 | 受影响资产数 |
| └CheckTime | Long | 1543991525000 | 检测时间 |
| └ItemId | Long | 1 | 检查项id |
| └RemainingTime | Integer | 0 | 预计检查时间 |
| └RiskItemResources | | | 详情数据 |
| └ContentResource | Json | { "type": "link", "value": "未开启多因素认证，存在风险\n", "url": "https://account.console.aliyun.com/#/secure\n" } | 详情json |
| └ResourceName | String | bestPractice | 详情条目 bestPractice检查描述，influence威胁影响，suggestion指导方案，helpResource帮助资源 |
| └RiskLevel | String | high | 风险等级 high medium low |
| └Sort | Integer | 1 | 顺序 |
| └Status | String | pass | 检查项检查状态 pass通过 failed失败 running运行中 waiting等待运行 ignored已忽略 falsePositive已标记误报 |
| └TaskId | Long | 5 | 任务id |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| └Title | String | 云平台-主账号双因素认证配置检查 | 检查项标题 |
| └Type | String | 身份认证及权限 | 类型 |
| PageCount | Integer | 2 | 页数 |
| PageSize | Integer | 10 | 页大小 |
| RequestId | String | AD271C07-4ACE-413D-AA9B-F14FD3B7717F | 请求id |
| TotalCount | Integer | 12 | 总数 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeRiskCheckResult
&CurrentPage=1
&PageSize=20
&<公共请求参数>
```

### 正常返回示例

`JSON` 格式

```
{
 "PageCount":2,
 "Count":10,
 "TotalCount":12,
 "PageSize":10,
 "RequestId":"AD271C07-4ACE-413D-AA9B-F14FD3B7717F",
 "List":[
  {
   "Sort":3,
   "Status":"pass",
   "ItemId":5,
   "RiskItemResources":[
    {
     "ContentResource":{
      "value":"未开启多因素认证，存在风险\n",
      "type":"link",
      "url":"https://account.console.aliyun.com/#/secure\n"
     },
     "ResourceName":"bestPractice"
    },
    {
     "ContentResource":{
```

```
       "value":"在只使用单一密码认证的情况下，黑客可能通过暴力破解等手段获取您的云平台管理密码。建议对云平台管理员账号开启密码加手机短信双重身份认证，防止密码泄露带来的安全隐患。",
       "type":"text"
      },
      "ResourceName":"influence"
     },
     {
      "ContentResource":{
       "value":"进入阿里云后台，顺序操作：管理控制台-账号管理-安全设置-虚拟MFA-设置 ",
       "type":"link",
       "url":"https://account.console.aliyun.com/#/selectVerificationMethod"
      },
      "ResourceName":"suggestion"
     },
     {
      "ContentResource":{
       "value":"1、使用多重机制保护主账户安全原理：\nhttps://help.aliyun.com/document_detail/28643.html\n2、设置MFA方法：\nhttps://help.aliyun.com/document_detail/28635.html",
       "type":"text"
      },
      "ResourceName":"helpResource"
     }
    ],
    "RiskLevel":"high",
    "Type":"zh-身份认证及权限",
    "CheckTime":1543991525000,
    "AffectedCount":0,
    "TaskId":58,
    "Title":"云平台-主账号双因素认证配置检查",
    "RemainingTime":0
   },
   {
    "Sort":13,
    "Status":"pass",
    "ItemId":25,
    "RiskItemResources":[
     {
      "ContentResource":{
       "emptyGridValue":{
        "value":"无影响",
        "type":"text"
       },
       "type":"grid"
      },
      "ResourceName":"bestPractice"
     },
     {
      "ContentResource":{
       "value":"超长时间未登陆的子用户可能存在数据泄露隐患，当子用户被非法登陆后，登陆者可以直接控制您的云产品。",
       "type":"text"
      },
      "ResourceName":"influence"
     },
     {
      "ContentResource":{
       "value":"进入管理控制台-访问控制 RAM， 删除暂不使用的子用户。",
       "type":"link",
       "url":"https://ram.console.aliyun.com/?#/user/list"
      },
```

```
      "ResourceName":"suggestion"
     },
     {
      "ContentResource":{
       "emptyGridValue":{
        "value":"暂无风险",
        "type":"text"
       },
       "values":[],
       "columns":[
        {
         "title":"所在可用区",
         "key":"RegionId"
        },
        {
         "title":"Bucket 名",
         "key":"RiskInstance"
        },
        {
         "title":"风险描述",
         "key":"RiskDescribe"
        }
       ],
       "resultStatus":[
        {
         "id":1,
         "status":"failed"
        }
       ],
       "type":"grid"
      },
      "ResourceName":"helpResource"
     }
    ],
    "RiskLevel":"medium",
    "Type":"zh-身份认证及权限",
    "CheckTime":1543991523000,
    "AffectedCount":0,
    "TaskId":57,
    "Title":"子账号安全-长时间未使用账户",
    "RemainingTime":0
   }
  ],
  "CurrentPage":1
}
```


**错误码**

查看本产品错误码

## 5.6 DescribeSecurityCheckScheduleConfig

**查看用户自定义设置的检测周期以及时间段。**

**调用DescribeSecurityCheckScheduleConfig查看用户自定义设置的检测周期以及时间段。**

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeSecurityCheckScheduleConfig | 系统规定参数。取值：DescribeSecurityCheckScheduleConfig。 |
| Lang | String | 否 | cn | 语种 |
| SourceIp | String | 否 | 1.1.1.1 | 请求源ip |

### 返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 48D2E9A9-A1B0-4295-B727-0995757C47E9 | 请求id |
| RiskCheckJobConfig | | | 周期检测任务设置 |
| └DaysOfWeek | String | 1，2，3 | 每周检测时间 |
| └EndTime | Integer | 12 | 结束时间 |
| └StartTime | Integer | 6 | 开始时间 |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeSecurityCheckScheduleConfig
&<公共请求参数>
```

#### 正常返回示例

`JSON` 格式

```
{
```

```
 "RequestId":"C17F3C01-0AFD-4898-A950-AA0129025B41",
 "RiskCheckJobConfig":{
  "DaysOfWeek":"1,2,3",
  "EndTime":12,
  "StartTime":6
 }
}
```

### 错误码

查看本产品错误码

## 5.7 ModifyRiskSingleResultStatus

修改检查项影响资产的状态。

调用ModifyRiskSingleResultStatus对检查项影响的资产进行操作：可忽略或标记为误报。

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | ModifyRiskSingleResultStatus | 系统规定参数。取值：ModifyRiskSingleResultStatus。 |
| Ids.N | RepeatList | 否 | 1 | 影响资产id |
| SourceIp | String | 否 | 1.1.1.1 | 请求源ip |
| Status | String | 否 | ignored | 新状态 |
| TaskId | Long | 否 | 57 | 任务id |

### 返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 3B3F3A90-46A5-4023-A2D8-D68B14262F96 | 请求id |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=ModifyRiskSingleResultStatus
&Ids.1=4
&Status=ignored
&TaskId=57
&<公共请求参数>
```

### 正常返回示例

#### JSON 格式

```
{
 "RequestId":"3B3F3A90-46A5-4023-A2D8-D68B14262F96"
}
```

## 错误码

查看本产品错误码

# 6 漏洞管理

## 6.1 DescribeVulList

查询漏洞列表。

调用本接口查询漏洞列表。

**调试**

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeVulList | 要执行的操作。<br>取值：DescribeVulList |
| AliasName | String | 否 | RHSA-2019:0230-Important: polkit security update | 漏洞别名。 |
| CurrentPage | Integer | 否 | 1 | 漏洞列表分页页码。<br>起始值：1<br>默认值：1 |
| Dealed | String | 否 | n | 漏洞是否处理。<br>取值：<br>· y：已处理<br>· n：未处理 |
| Lang | String | 否 | zh | 语言。<br>取值：<br>· zh：中文<br>· en：英文 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Necessity | String | 否 | asap,later,nntf | 漏洞修复必要性等级。多个等级用英文逗号分隔。<br>取值：<br>· asap：高<br>· later：中<br>· nntf：低 |
| PageSize | Integer | 否 | 20 | 漏洞列表分页大小。<br>默认值：20 |
| Remark | String | 否 | 192.168.1.1 | 查询标记，可以为资产内网IP、外网IP或资产名称。 |
| Type | String | 否 | cve | 漏洞类型，包括以下几类：<br>· cve：Linux软件漏洞<br>· sys：Windows系统漏洞<br>· cms：Web-CMS漏洞<br>· app：应用漏洞<br>· emg：应急漏洞 |
| Uuids | String | 否 | 1587bedb-fdb4-48c4-9330-********** | 服务器UUID列表，多个用英文逗号分隔。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | ECDE6715-6286-40E6-A32D-3094051FD74D | 请求ID。 |
| CurrentPage | Integer | 1 | 漏洞列表分页页码。 |
| PageSize | Integer | 20 | 分页大小。 |
| TotalCount | Integer | 2 | 查询结果总数。 |
| VulRecords | | | 漏洞列表。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| AliasName | String | RHSA-2017:0574 : gnutls security , bug fix, and enhancement update | 漏洞别名。 |
| ExtendContentJson | | | 扩展信息。 |
| AbsolutePath | String | /roo/www/web | 绝对路径。 |
| AliasName | String | RHSA-2017:0574 : gnutls security , bug fix, and enhancement update | 漏洞别名。 |
| LastTs | Long | 1554189334000 | 最后发现时间。 |
| Necessity | | | 漏洞修复必要性。 |
| Assets_factor | String | 1 | 资产因子。 |
| Cvss_factor | String | 7.8 | CVSS因子。 |
| Enviroment_factor | String | 1.0 | 环境因子。 |
| Gmt_create | String | 20190331 | 创建时间。 |
| Is_calc | String | 1 | 是否计算出分数。<br>· 0：未计算<br>· 1：已计算 |
| Status | String | normal | 状态。<br>取值：<br>· none：未生成分数<br>· pending：等待计算中<br>· miss：未计算出分数<br>· normal：正常 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Time_factor | String | 1.0 | 时间因子。 |
| Total_score | String | 7.8 | 总分。 |
| Os | String | centos | 操作系统。 |
| OsRelease | String | 7 | 操作系统release。 |
| PrimaryId | Long | 111 | 漏洞ID。 |
| RpmEntityList | | | RPM包列表。 |
| FullVersion | String | 3.10.0-693.2.2.el7 | 完整版本号。 |
| MatchDetail | String | python-perf version less than 0:3.10.0-693.21.1.el7 | 匹配细节 |
| Name | String | python-perf | RPM名称。 |
| Path | String | /usr/lib64/python2.7/site-packages | 路径。 |
| UpdateCmd | String | yum update python-perf | 修复命令。 |
| Version | String | 3.10.0 | 版本号。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Status | Integer | 1 | 漏洞状态。<br>取值：<br>· 1：未修复<br>· 2：修复失败<br>· 3：回滚失败<br>· 4：修复中<br>· 5：回滚中<br>· 6：验证中<br>· 7：修复成功<br>· 8：修复成功待重启<br>· 9：回滚成功<br>· 10：已忽略<br>· 11：回滚成功待重启<br>· 12：漏洞不存在<br>· 20：已失效 |
| Tag | String | oval | 漏洞标签。 |
| cveList | | ["CVE-2016-8610", "CVE-2017-5335" ] | CVE列表。 |
| FirstTs | Long | 1554189334000 | 首次发现时间，时间戳。 |
| GroupId | Integer | 281801 | 资产分组ID。 |
| InstanceId | String | i-bp18tnigcymjvmc2fw9e | 资产实例ID。 |
| InstanceName | String | 测试ECS | 资产实例名称。 |
| InternetIp | String | 47.99.0.0 | 资产外网IP。 |
| IntranetIp | String | 192.1.1.1 | 资产内网IP。 |
| Ip | String | 47.99.0.0 | 资产IP，优先显示外网IP。 |
| LastTs | Long | 1541207563000 | 最后发现时间，时间戳。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| ModifyTs | Long | 1541207563000 | 修改时间，时间戳。 |
| Name | String | oval:com.redhat.rhsa:def:20170574 | 漏洞名称。 |
| Necessity | String | asap | 漏洞修复必要性。<br>· asap：高<br>· later：中<br>· nntf：低 |
| NeedReboot | String | yes | 是否需要重启。<br>· yes：是<br>· no：否 |
| OsVersion | String | linux | 操作系统版本。 |
| PrimaryId | Long | 101162078 | 漏洞ID。 |
| Related | String | CVE-2017-7518,CVE-2017-12188 | 漏洞关联CVE列表，多个用英文逗号分隔。 |
| RepairTs | Long | 1541207563000 | 修复时间，时间戳。 |
| ResultCode | String | 0 | 修复返回码。 |
| ResultMessage | String | timeout | 修复返回消息。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Status | Integer | 1 | 漏洞状态。<br>· 1：未修复<br>· 2：修复失败<br>· 3：回滚失败<br>· 4：修复中<br>· 5：回滚中<br>· 6：验证中<br>· 7：修复成功<br>· 8：修复成功待重启<br>· 9：回滚成功<br>· 10：已忽略<br>· 11：回滚成功待重启<br>· 12：漏洞不存在<br>· 20：已失效 |
| Tag | String | oval | 漏洞标签。 |
| Type | String | cve | 漏洞类型，包含以下类型：<br>· cve：Linux漏洞<br>· sys：Windows漏洞<br>· cms：WebCMS漏洞<br>· emg：应急漏洞<br>· app：应用漏洞 |
| Uuid | String | 04c56617-23fc-43a5-ab9b-********** | 资产UUID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeVulList
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribeVulList>
    <TotalCount>6430</TotalCount>
    <PageSize>2</PageSize>
```

```
   <RequestId>ECDE6715-6286-40E6-A32D-3094051FD74D</RequestId>
   <CurrentPage>1</CurrentPage>
   <VulRecords>
      <Necessity>asap</Necessity>
      <Uuid>04c56617-23fc-43a5-ab9b-755da574ffe8</Uuid>
      <Ip>47.99.63.178</Ip>
      <ModifyTs>1541347310000</ModifyTs>
      <Type>cve</Type>
      <FirstTs>1541207563000</FirstTs>
      <InstanceId>i-bp18tnigcymjvmc2fw9e</InstanceId>
      <InternetIp>47.99.63.178</InternetIp>
      <ResultMessage>
   out:Loaded plugins: security
   Setting up Update Process
   Resolving Dependencies
   --&gt; Running transaction check
   ---&gt; Package gnutls.x86_64 0:2.8.5-14.el6_5 will be updated
   ---&gt; Package gnutls.x86_64 0:2.12.23-22.el6 will be an update
   --&gt; Finished Dependency Resolution

   Dependencies Resolved


 ================================================================================
    Package         Arch            Version
Repository     Size

 ================================================================================
   Updating:
    gnutls          x86_64          2.12.23-22.el6              base
       389 k

   Transaction Summary

 ================================================================================
   Upgrade       1 Package(s)

   Total download size: 389 k
   Downloading Packages:
   Running rpm_check_debug
   Running Transaction Test
   Transaction Test Succeeded
   Running Transaction

Updating   : gnutls-2.12.23-22.el6.x86_64
    1/2

Cleanup    : gnutls-2.8.5-14.el6_5.x86_64
    2/2

Verifying  : gnutls-2.12.23-22.el6.x86_64
    1/2

Verifying  : gnutls-2.8.5-14.el6_5.x86_64
    2/2

   Updated:
   gnutls.x86_64 0:2.12.23-22.el6


   Complete!

  err:</ResultMessage>
```

```
      <Related>CVE-2016-8610,CVE-2017-5335,CVE-2017-5336,CVE-2017-5337
</Related>
      <GroupId>281801</GroupId>
      <OsVersion>linux</OsVersion>
      <ExtendContentJson>
         <Necessity>
            <Status>pending</Status>
         </Necessity>
         <Os>centos</Os>
         <cveList>CVE-2016-8610</cveList>
         <cveList>CVE-2017-5335</cveList>
         <cveList>CVE-2017-5336</cveList>
         <cveList>CVE-2017-5337</cveList>
         <RpmEntityList>
            <Name>gnutls</Name>
            <Version>2.8.5</Version>
            <FullVersion>2.8.5-14.el6_5</FullVersion>
            <MatchDetail>gnutls version less than 0:2.12.23-21.el6</
MatchDetail>
            <UpdateCmd>yum update gnutls</UpdateCmd>
            <Path>/usr/lib64/libgnutls-extra.so.26</Path>
         </RpmEntityList>
         <OsRelease>6</OsRelease>
      </ExtendContentJson>
      <Name>oval:com.redhat.rhsa:def:20170574</Name>
      <Status>7</Status>
      <LastTs>1541207563000</LastTs>
      <NeedReboot>no</NeedReboot>
      <AliasName>RHSA-2017:0574: gnutls security, bug fix, and
enhancement update</AliasName>
      <Tag>oval</Tag>
      <IntranetIp>10.0.0.173</IntranetIp>
      <PrimaryId>101162078</PrimaryId>
      <ResultCode>0</ResultCode>
      <Level>serious</Level>
      <InstanceName>雷鹰测试-Aegis123456789</InstanceName>
   </VulRecords>
   <VulRecords>
      <Necessity>later</Necessity>
      <Uuid>1bfac26f-0301-435e-bcfd-2cdbd271c8a1</Uuid>
      <Ip>172.19.220.94</Ip>
      <ModifyTs>1554096622000</ModifyTs>
      <Type>cve</Type>
      <FirstTs>1550891785000</FirstTs>
      <InstanceId>i-uf6iywlcvu7n0v8er15l</InstanceId>
      <InternetIp></InternetIp>
      <Related>CVE-2017-7518,CVE-2017-12188</Related>
      <GroupId>281801</GroupId>
      <OsVersion>linux</OsVersion>
      <ExtendContentJson>
         <Necessity>
            <Cvss_factor>7.8</Cvss_factor>
            <Total_score>7.8</Total_score>
            <Status>normal</Status>
            <Enviroment_factor>1.0</Enviroment_factor>
            <Time_factor>1.0</Time_factor>
            <Assets_factor>1</Assets_factor>
            <Gmt_create>20190331</Gmt_create>
            <Is_calc>1</Is_calc>
         </Necessity>
         <Os>centos</Os>
         <cveList>CVE-2017-7518</cveList>
         <cveList>CVE-2017-12188</cveList>
         <RpmEntityList>
```

```
            <Name>kernel</Name>
            <Version>3.10.0</Version>
            <FullVersion>3.10.0-693.2.2.el7</FullVersion>
            <MatchDetail>kernel version less than 0:3.10.0-693.21.1.
el7</MatchDetail>
            <UpdateCmd>yum update kernel</UpdateCmd>
            <Path>/boot/.vmlinuz-3.10.0-693.2.2.el7.x86_64.hmac</Path>
         </RpmEntityList>
         <RpmEntityList>
            <Name>kernel-headers</Name>
            <Version>3.10.0</Version>
            <FullVersion>3.10.0-693.2.2.el7</FullVersion>
            <MatchDetail>kernel-headers version less than 0:3.10.0-693
.21.1.el7</MatchDetail>
            <UpdateCmd>yum update kernel-headers</UpdateCmd>
            <Path>/usr/include/asm</Path>
         </RpmEntityList>
         <RpmEntityList>
            <Name>kernel-tools</Name>
            <Version>3.10.0</Version>
            <FullVersion>3.10.0-693.2.2.el7</FullVersion>
            <MatchDetail>kernel-tools version less than 0:3.10.0-693.
21.1.el7</MatchDetail>
            <UpdateCmd>yum update kernel-tools</UpdateCmd>
            <Path>/etc/sysconfig/cpupower</Path>
         </RpmEntityList>
         <RpmEntityList>
            <Name>kernel-tools-libs</Name>
            <Version>3.10.0</Version>
            <FullVersion>3.10.0-693.2.2.el7</FullVersion>
            <MatchDetail>kernel-tools-libs version less than 0:3.10.0-
693.21.1.el7</MatchDetail>
            <UpdateCmd>yum update kernel-tools-libs</UpdateCmd>
            <Path>/usr/lib64/libcpupower.so.0</Path>
         </RpmEntityList>
         <RpmEntityList>
            <Name>python-perf</Name>
            <Version>3.10.0</Version>
            <FullVersion>3.10.0-693.2.2.el7</FullVersion>
            <MatchDetail>python-perf version less than 0:3.10.0-693.21
.1.el7</MatchDetail>
            <UpdateCmd>yum update python-perf</UpdateCmd>
            <Path>/usr/lib64/python2.7/site-packages</Path>
         </RpmEntityList>
         <OsRelease>7</OsRelease>
      </ExtendContentJson>
      <Name>oval:com.redhat.rhsa:def:20180395</Name>
      <Status>1</Status>
      <LastTs>1554096622000</LastTs>
      <NeedReboot>yes</NeedReboot>
      <AliasName>RHSA-2018:0395-Important: kernel security and bug fix
 update</AliasName>
      <Tag>oval</Tag>
      <IntranetIp>172.19.0.0</IntranetIp>
      <PrimaryId>160191232</PrimaryId>
      <Level>low</Level>
      <InstanceName>master-03-k8s-for-cs-cce881ec0ec77435e8e2
1bb52d1178e2a</InstanceName>
   </VulRecords>
```

```
    </DescribeVulList>
```

JSON 格式


```
{
 "TotalCount":6430,
 "PageSize":2,
 "RequestId":"ECDE6715-6286-40E6-A32D-3094051FD74D",
 "CurrentPage":1,
 "VulRecords":[
  {
   "Necessity":"asap",
   "Uuid":"04c56617-23fc-43a5-ab9b-755da574ffe8",
   "Ip":"47.99.63.178",
   "ModifyTs":1541347310000,
   "Type":"cve",
   "FirstTs":1541207563000,
   "InstanceId":"i-bp18tnigcymjvmc2fw9e",
   "InternetIp":"47.99.63.178",
   "ResultMessage":"out:Loaded plugins: security\nSetting up Update
 Process\nResolving Dependencies\n--> Running transaction check
\n---> Package gnutls.x86_64 0:2.8.5-14.el6_5 will be updated\n
---> Package gnutls.x86_64 0:2.12.23-22.el6 will be an update\n
--> Finished Dependency Resolution\n\nDependencies Resolved\n\n
================================================================================
\n Package         Arch            Version
                     Repository     Size\n
================================================================================
\nUpdating:\n gnutls          x86_64          2.12.23-22.
el6             base          389 k\n\nTransaction Summary\n
================================================================================
\nUpgrade       1 Package(s)\n\nTotal download size: 389 k\nDownloadi
ng Packages:\nRunning rpm_check_debug\nRunning Transaction Test\
nTransaction Test Succeeded\nRunning Transaction\n\r  Updating   :
gnutls-2.12.23-22.el6.x86_64                                   1/2 \n\
r  Cleanup    : gnutls-2.8.5-14.el6_5.x86_64
         2/2 \n\r  Verifying  : gnutls-2.12.23-22.el6.x86_64
                         1/2 \n\r  Verifying  : gnutls-2.8.5-14.el6_5
.x86_64                                  2/2 \n\nUpdated:\n  gnutls.
x86_64 0:2.12.23-22.el6
 \n\nComplete!\n\nerr:",
   "Related":"CVE-2016-8610,CVE-2017-5335,CVE-2017-5336,CVE-2017-5337
",
   "GroupId":281801,
   "OsVersion":"linux",
   "Name":"oval:com.redhat.rhsa:def:20170574",
   "ExtendContentJson":{
    "Os":"centos",
    "Necessity":{
     "Status":"pending"
    },
    "cveList":[
     "CVE-2016-8610",
     "CVE-2017-5335",
     "CVE-2017-5336",
     "CVE-2017-5337"
    ],
    "RpmEntityList":[
     {
      "Name":"gnutls",
      "FullVersion":"2.8.5-14.el6_5",
      "Version":"2.8.5",
      "MatchDetail":"gnutls version less than 0:2.12.23-21.el6",
```

```
      "Path":"/usr/lib64/libgnutls-extra.so.26",
      "UpdateCmd":"yum update gnutls"
     }
    ],
    "OsRelease":"6"
   },
   "Status":7,
   "LastTs":1541207563000,
   "NeedReboot":"no",
   "AliasName":"RHSA-2017:0574: gnutls security, bug fix, and
enhancement update",
   "Tag":"oval",
   "PrimaryId":101162078,
   "IntranetIp":"10.0.0.173",
   "ResultCode":"0",
   "Level":"serious",
   "InstanceName":"雷鹰测试-Aegis123456789"
  },
  {
   "Necessity":"later",
   "Uuid":"1bfac26f-0301-435e-bcfd-2cdbd271c8a1",
   "Ip":"172.19.220.94",
   "ModifyTs":1554096622000,
   "Type":"cve",
   "FirstTs":1550891785000,
   "InstanceId":"i-uf6iywlcvu7n0v8er15l",
   "InternetIp":"",
   "Related":"CVE-2017-7518,CVE-2017-12188",
   "GroupId":281801,
   "OsVersion":"linux",
   "ExtendContentJson":{
    "Os":"centos",
    "Necessity":{
     "Cvss_factor":"7.8",
     "Status":"normal",
     "Total_score":"7.8",
     "Enviroment_factor":"1.0",
     "Assets_factor":"1",
     "Time_factor":"1.0",
     "Gmt_create":"20190331",
     "Is_calc":"1"
    },
    "cveList":[
     "CVE-2017-7518",
     "CVE-2017-12188"
    ],
    "RpmEntityList":[
     {
      "Name":"kernel",
      "FullVersion":"3.10.0-693.2.2.el7",
      "Version":"3.10.0",
      "MatchDetail":"kernel version less than 0:3.10.0-693.21.1.el7",
      "Path":"/boot/.vmlinuz-3.10.0-693.2.2.el7.x86_64.hmac",
      "UpdateCmd":"yum update kernel"
     },
     {
      "Name":"kernel-headers",
      "FullVersion":"3.10.0-693.2.2.el7",
      "Version":"3.10.0",
      "MatchDetail":"kernel-headers version less than 0:3.10.0-693.21.
1.el7",
      "Path":"/usr/include/asm",
      "UpdateCmd":"yum update kernel-headers"
     },
```

```
      {
       "Name":"kernel-tools",
       "FullVersion":"3.10.0-693.2.2.el7",
       "Version":"3.10.0",
       "MatchDetail":"kernel-tools version less than 0:3.10.0-693.21.1.
el7",
       "Path":"/etc/sysconfig/cpupower",
       "UpdateCmd":"yum update kernel-tools"
      },
      {
       "Name":"kernel-tools-libs",
       "FullVersion":"3.10.0-693.2.2.el7",
       "Version":"3.10.0",
       "MatchDetail":"kernel-tools-libs version less than 0:3.10.0-693.
21.1.el7",
       "Path":"/usr/lib64/libcpupower.so.0",
       "UpdateCmd":"yum update kernel-tools-libs"
      },
      {
       "Name":"python-perf",
       "FullVersion":"3.10.0-693.2.2.el7",
       "Version":"3.10.0",
       "MatchDetail":"python-perf version less than 0:3.10.0-693.21.1.
el7",
       "Path":"/usr/lib64/python2.7/site-packages",
       "UpdateCmd":"yum update python-perf"
      }
     ],
     "OsRelease":"7"
    },
    "Name":"oval:com.redhat.rhsa:def:20180395",
    "Status":1,
    "LastTs":1554096622000,
    "NeedReboot":"yes",
    "AliasName":"RHSA-2018:0395-Important: kernel security and bug fix
update",
    "Tag":"oval",
    "PrimaryId":160191232,
    "IntranetIp":"172.19.0.0",
    "Level":"low",
    "InstanceName":"master-03-k8s-for-cs-cce881ec0ec77435e8e21bb52d1178
e2a"
   }
  ]
}
```


**错误码**

**访问错误中心查看更多错误码。**

## 6.2 DescribeEmgVulGroup

**分组查询应急漏洞。**

**调用DescribeEmgVulGroup分组查询应急漏洞。**

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeEmgVulGroup | 需要执行的操作。<br>取值：DescribeEmgVulGroup |
| Lang | String | 否 | zh | 语言。<br>取值：<br>· zh：中文<br>· en：英文 |

### 返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| EmgVulGroupList | | | 应急漏洞分组列表。 |
| └AliasName | String | Jenkins 远程高危安全漏洞(CVE-2018-1999001和CVE-2018-1999002) | 漏洞别名。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| └Description | String | Jenkins是一个开源软件项目，是基于Java开发的一种持续集成工具，用于监控持续重复的工作，旨在提供一个开放易用的软件平台，使软件的持续集成变成可能。\n\nJenkins存在任意文件读取漏洞，攻击者在远程且未经授权的情况下，可以通过构造恶意的HTTP 请求发往Jenkins Web服务端，从请求响应中直接获取攻击者指定读取的文件内容。 | 漏洞描述。 |
| └GmtPublish | Long | 1532592480000 | 漏洞发布时间，时间戳。 |
| └Name | String | scan:ACSV-2018-072601 | 漏洞名称。 |
| └PendingCount | Integer | 0 | 待处理漏洞数。 |
| └Type | String | scan | 应急类型。<br>· scan：扫描插件<br>· python：扫描脚本 |
| RequestId | String | E836EDA2-DBFB-489E-8FD3-5B141EB81A9C | 请求ID。 |
| TotalCount | Integer | 2 | 漏洞总数。 |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeEmgVulGroup
&<公共请求参数>
```

#### 正常返回示例

JSON 格式


```
{
 "TotalCount":2,
 "EmgVulGroupList":[
  {
   "Name":"scan:ACSV-2018-072601",
   "Status":30,
   "Description":"Jenkins是一个开源软件项目，是基于Java开发的一种持续集成工具，用于监控持续重复的工作，旨在提供一个开放易用的软件平台，使软件的持续集成变成可能。\n\nJenkins存在任意文件读取漏洞，攻击者在远程且未经授权的情况下，可以通过构造恶意的HTTP 请求发往Jenkins Web服务端，从请求响应中直接获取攻击者指定读取的文件内容。",
   "PendingCount":0,
   "AliasName":"Jenkins 远程高危安全漏洞(CVE-2018-1999001和CVE-2018-1999002)",
   "Type":"scan",
   "GmtPublish":1532592480000
  },
  {
   "Name":"scan:acsv-2018-082001",
   "Status":30,
   "Description":"近日，有安全研究人员披露Metinfo多个高危安全漏洞，包括任意文件读取，XXE，敏感信息泄露等危害严重的漏洞。",
   "PendingCount":0,
   "AliasName":"Metinfo 多个高危漏洞",
   "Type":"scan",
   "GmtPublish":1534767031000
  }
 ],
 "RequestId":"E836EDA2-DBFB-489E-8FD3-5B141EB81A9C"
}
```


### 错误码

查看本产品错误码

## 6.3 DescribeVulWhitelist

分页获取漏洞白名单。

调用DescribeVulWhitelist分页获取漏洞白名单

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeVulWhitelist | 需要执行的操作。<br>取值：DescribeVulWhitelist |
| CurrentPage | Integer | 否 | 1 | 分页页码。<br>起始值：1<br>默认值：1 |
| PageSize | Integer | 否 | 10 | 分页大小。<br>默认值：20 |

### 返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CurrentPage | Integer | 1 | 分页页码。 |
| PageSize | Integer | 10 | 分页大小。 |
| RequestId | String | 74F97EF7-B543-43FD-A4E9-18456731F9C5 | 请求ID。 |
| TotalCount | Integer | 1 | 数据总数。 |
| VulWhitelists | | | 漏洞白名单列表。 |
| └AliasName | String | RHSA-2017:3263 : curl security update | 漏洞别名 |
| └Name | String | oval:com.redhat.rhsa:def:20173263 | 漏洞名称 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| └Reason | String | 暂不修复 | 加白原因 |
| └Type | String | cve | 漏洞类型 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribeVulWhitelist
&<公共请求参数>
```

**正常返回示例**

`JSON` 格式

```
{
 "TotalCount":1,
 "PageSize":3,
 "VulWhitelists":[
  {
   "Name":"oval:com.redhat.rhsa:def:20173263",
   "AliasName":"RHSA-2017:3263: curl security update",
   "Type":"cve",
   "Reason":"暂不修复"
  }
 ],
 "RequestId":"74F97EF7-B543-43FD-A4E9-18456731F9C5",
 "CurrentPage":1
}
```

**错误码**

查看本产品错误码

## 6.4 DescribeConcernNecessity

**查询关注的漏洞修复必要性。**

**调用DescribeConcernNecessity接口可查询关注的漏洞修复必要性。**

**调试**

**前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。**

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeConcernNecessity | 系统规定参数。取值：DescribeConcernNecessity。 |
| Lang | String | 否 | zh | 语言。<br>· zh：中文<br>· en：英文 |

### 返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| ConcernNecessity | | [ "asap", "later", "nntf"] | 漏洞修复必要性列表。<br>· asap：高<br>· later：中<br>· nntf：低 |
| RequestId | String | 9139DAF6-5CE0-49A0-926E-07243A290F70 | 请求ID。 |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeConcernNecessity
&Lang=zh
&<公共请求参数>
```

#### 正常返回示例

#### JSON 格式

```
{
 "ConcernNecessity":[
  "asap",
  "later",
  "nntf"
 ],
 "RequestId":"9139DAF6-5CE0-49A0-926E-07243A290F70"
}
```

### 错误码

查看本产品错误码

## 6.5 DescribeGroupedVul

分组查询漏洞信息。

调用DescribeGroupedVul接口可对漏洞进行分组查询。

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeGroupedVul | 系统规定参数。取值：DescribeGroupedVul。 |
| AliasName | String | 否 | RHSA-2019:0230-Important: polkit security update | 漏洞别名。 |
| CurrentPage | Integer | 否 | 1 | 分页页码，起始值为1，默认值为1。 |
| Dealed | String | 否 | n | 漏洞是否处理。<br>· y：已处理<br>· n：未处理 |
| Lang | String | 否 | zh | 语言。<br>· zh：中文<br>· en：英文 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Necessity | String | 否 | asap,later,nntf | 漏洞修复必要性，多个用英文逗号分隔。<br>· asap：高<br>· later：中<br>· nntf：低 |
| PageSize | Integer | 否 | 20 | 分页大小，默认值为20。 |
| Type | String | 否 | cve | 漏洞类型。<br>· cve：Linux漏洞<br>· sys：Windows漏洞<br>· cms：WebCMS漏洞 |
| Uuids | String | 否 | d42f938c-d962-48a0-90f9-05e4ea****** | 资产的UUID列表，多个用英文逗号分隔。 |

**返回参数**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CurrentPage | Integer | 1 | 分页页码。 |
| PageSize | Integer | 20 | 分页大小。 |
| RequestId | String | 0DFCADBA-7065-42DA-AF17-6868B9C2A8CF | 请求ID。 |
| TotalCount | Integer | 2 | 查询结果总数。 |
| GroupedVulItems | | | 分组漏洞列表。 |
| └AliasName | String | RHSA-2019:0230-Important: polkit security update | 漏洞别名。 |
| └AsapCount | Integer | 0 | 修复必要性高数量 |
| └GmtLast | Long | 1554185744000 | 漏洞最后发现时间，时间戳。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| └ HandledCount | Integer | 0 | 已处理漏洞数量。 |
| └LaterCount | Integer | 0 | 修复必要性为“中等”的漏洞数量。 |
| └NntfCount | Integer | 59 | 修复必要性为低的漏洞数量。 |
| └Tags | String | 需要重启 | 漏洞标签。 |
| └Type | String | cve | 漏洞类型。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribeGroupedVul
&Lang=zh
&Type=cve
&AliasName=RHSA-2019:0230-Important: polkit security update
&Necessity=asap,later,nntf
&Dealed=n
&CurrentPage=1
&PageSize=20
&<公共请求参数>
```

**正常返回示例**

`JSON` **格式**

```
{
 "GroupedVulItems":[
  {
   "Tags":"",
   "LaterCount":0,
   "AliasName":"RHSA-2019:0230-Important: polkit security update",
   "Type":"cve",
   "AsapCount":0,
   "NntfCount":59,
   "GmtLast":1554185744000,
   "HandledCount":0
  },
  {
   "Tags":"",
   "LaterCount":0,
   "AliasName":"RHSA-2019:0368-Important: systemd security update",
   "Type":"cve",
   "AsapCount":0,
   "NntfCount":59,
   "GmtLast":1554185744000,
   "HandledCount":0
  }
```

```
 ],
 "TotalCount":432,
 "PageSize":2,
 "RequestId":"0DFCADBA-7065-42DA-AF17-6868B9C2A8CF",
 "CurrentPage":1
}
```

### 错误码

查看本产品错误码

## 6.6 ModifyCreateVulWhitelist

创建漏洞白名单。

调用ModifyCreateVulWhitelist接口可创建漏洞白名单，加入白名单中的漏洞不再展示在告警提示中。

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | ModifyCreateVulWhitelist | 系统规定参数。取值：ModifyCreateVulWhitelist。 |
| Reason | String | 否 | 确认该漏洞无危害 | 添加白名单原因说明。 |
| Whitelist | String | 否 | [{"name":"oval:com.redhat.rhsa:def:20173263","type":"cve","aliasName":"RHSA-2017:3263: curl security update"}] | 添加到白名单中的漏洞信息，JSON格式。 |

返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | DFE4F166-1AC9-4FAC-A4E4-F0608AD705A6 | 请求ID。 |

示例

请求示例

```
/?Action=ModifyCreateVulWhitelist
&Reason=确认该漏洞无危害
&Whitelist=[{"name":"oval:com.redhat.rhsa:def:20173263","type":"cve","
aliasName":"RHSA-2017:3263: curl security update"}]
&<公共请求参数>
```

正常返回示例

`JSON` 格式

```
{
 "RequestId":"DFE4F166-1AC9-4FAC-A4E4-F0608AD705A6"
}
```

错误码

查看本产品错误码

## 6.7 DescribeAutoDelConfig

获取漏洞自动删除配置。

调用DescribeAutoDelConfig接口可获取漏洞自动删除的配置。

调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeAutoDelConfig | 系统规定参数。取值：DescribeAutoDelConfig。 |

返回参数

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Days | Integer | 7 | 漏洞自动删除天数。 |
| RequestId | String | 43186B14-5A39-44E0-860E-8B24D1EF82CC | 请求ID。 |

示例

请求示例

```
/?Action=DescribeAutoDelConfig
&<公共请求参数>
```

正常返回示例

JSON 格式

```
{
 "RequestId":"43186B14-5A39-44E0-860E-8B24D1EF82CC",
 "Days":7
}
```

错误码

查看本产品错误码

## 6.8 ModifyOperateVul

对漏洞进行操作，包括验证漏洞、忽略漏洞、修复漏洞等。

调用ModifyOperateVul接口可对检测到的漏洞执行对应的操作，包括验证漏洞、忽略漏洞、修复漏洞等。

调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | ModifyOperateVul | 系统规定参数。取值：ModifyOperateVul。 |
| Info | String | 否 | [{"name":"oval:com.redhat.rhsa:def:20171100","uuid":"e49fb179-bbe7-4b6c-ab65-e106fb58d235","tag":"oval"}] | 操作漏洞的必要信息，JSON列表，每个JSON对象需要包括：uuid，name，tag三个字段 |
| OperateType | String | 否 | vul_ignore | 操作类型。<br>· vul_fix：修复漏洞<br>· vul_verify：验证漏洞<br>· vul_rebooted：已重启待验证<br>· vul_rollback：回滚<br>· vul_ignore：忽略漏洞<br>· vul_undo_ignore：取消忽略<br>· vul_delete：删除漏洞 |
| Reason | String | 否 | 暂缓修复 | 表示漏洞忽略的原因。 |
| Type | String | 否 | cve | 漏洞类型。<br>· cve：Linux漏洞<br>· sys：Windows漏洞<br>· cms：WebCMS漏洞<br>· emg：应急漏洞 |

**返回参数**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | DFE4F166-1AC9-4FAC-A4E4-F0608AD705A6 | 请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=ModifyOperateVul
&Info=[{"name":"oval:com.redhat.rhsa:def:20171100","uuid":"e49fb179-
bbe7-4b6c-ab65-e106fb58d235","tag":"oval"}]
&OperateType=vul_ignore
&Type=cve
&Reason=暂缓修复
&<公共请求参数>
```

### 正常返回示例

#### JSON 格式

```
{
 "RequestId":"DFE4F166-1AC9-4FAC-A4E4-F0608AD705A6"
}
```

## 错误码

查看本产品错误码

# 7 基线

## 7.1 DescribeCheckWarningSummary

查看基线检查结果统计。

调用DescribeCheckWarningSummary接口，可查看基线检查的结果统计情况，包含检查的服务器数量、基线检查项数量、最近检查通过率等。

**调试**

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeCheckWarningSummary | 系统规定参数。<br>取值：DescribeCheckWarningSummary |
| CurrentPage | Integer | 否 | 1 | 分页页码。 |
| Lang | String | 否 | zh | 语言。 |
| PageSize | Integer | 否 | 10 | 分页大小。 |
| RiskName | String | 否 | Redis安全基线检查 | 基线检查风险项名称。 |
| RiskStatus | Integer | 否 | 1 | 基线检查的状态。<br>· 1：未通过<br>· 3：已通过 |
| SourceIp | String | 否 | 127.0.0.1 | 来源IP。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Status | String | 否 | 1 | 检查项状态。<br>· 1：未通过<br>· 2：验证中<br>· 3：已通过<br>· 5：已失效<br>· 6：已忽略 |
| StrategyId | Long | 否 | 1 | 策略ID。 |
| TypeName | String | 否 | database | 基线一级类型。 |
| Uuids | String | 否 | f03259d8-1e81-4fae-bcbb-275fb5****** | 机器ID。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 10 | 当前检测项条数。 |
| CurrentPage | Integer | 1 | 当前页。 |
| PageSize | Integer | 10 | 每页条数。 |
| RequestId | String | 00BD7CE2-284A-4534-BD09-FB69836DD750 | 请求ID。 |
| TotalCount | Integer | 100 | 基线检查项的总数。 |
| WarningSummarys | | | 检查项统计明细。 |
| CheckCount | Integer | 10 | 基线检查项个数。 |
| HighWarningCount | Integer | 1 | 高危检查项个数。 |
| LastFoundTime | String | 2019-01-01 12:23:00 | 最近执行基线检查的时间。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Level | String | high | 基线检查风险项的危险等级。<br>包含以下等级：<br>-高危<br>-中危<br>-低危 |
| LowWarningCount | Integer | 3 | 低危检查项的个数。 |
| MediumWarningCount | Integer | 2 | 中危检查项的个数。 |
| RiskId | Long | 1 | 风险项ID。 |
| RiskName | String | Redis密码检查 | 基线检查风险项名称。 |
| SubTypeAlias | String | Redis检查 | 风险项二级分类。 |
| TypeAlias | String | 数据库 | 基线检查项的分类，例如：数据库、系统、弱密码检测和中间件。 |
| WarningMachineCount | Integer | 11 | 检测出基线风险项的资产的数量。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribeCheckWarningSummary
&Lang=zh
&TypeName=database
&Status=1
&RiskStatus=1
&RiskName=Redis安全基线检查
&StrategyId=1
&Uuids=f03259d8-1e81-4fae-bcbb-275fb55efb34
&CurrentPage=1
&PageSize=20
&<公共请求参数>
```

**正常返回示例**

XML 格式

```
<DescribeCheckWarningSummary>
  <TotalCount>25</TotalCount>
  <PageSize>20</PageSize>
  <RequestId>DFA6CDC5-E826-4D18-A499-BEF9DA31F1AD</RequestId>
  <CurrentPage>1</CurrentPage>
  <Count>20</Count>
  <WarningSummarys>
    <RiskId>43</RiskId>
    <LastFoundTime>2019-04-10 00:33:00</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>3</WarningMachineCount>
    <CheckCount>16</CheckCount>
    <HighWarningCount>8</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>
   CentOS Linux 7安全基线检查

                                </SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>
   CentOS Linux 7安全基线检查

                                </RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>47</RiskId>
    <LastFoundTime>2019-04-10 00:58:11</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>2</WarningMachineCount>
    <CheckCount>15</CheckCount>
    <HighWarningCount>7</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>CentOS Linux 7合规基线检查-等保三级</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>CentOS Linux 7合规基线检查-等保三级</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>48</RiskId>
    <LastFoundTime>2019-04-10 00:58:11</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>2</WarningMachineCount>
    <CheckCount>12</CheckCount>
    <HighWarningCount>6</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>CentOS Linux 7合规基线检查-等保二级</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>CentOS Linux 7合规基线检查-等保二级</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>3</RiskId>
    <LastFoundTime>2019-04-10 08:31:32</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>1</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>1</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>数据库</TypeAlias>
    <SubTypeAlias>Redis安全基线检查</SubTypeAlias>
```

```
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Redis安全基线检查</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>19</RiskId>
    <LastFoundTime>2019-04-10 08:31:32</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>1</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>1</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>弱密码检测</TypeAlias>
    <SubTypeAlias>Linux系统登录弱口令检测</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Linux系统登录弱口令检测</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>12</RiskId>
    <LastFoundTime>2019-04-10 08:31:32</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>1</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>1</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>弱密码检测</TypeAlias>
    <SubTypeAlias>FTP匿名登录配置检测</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>FTP匿名登录配置检测</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>13</RiskId>
    <LastFoundTime>2019-04-10 06:58:58</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>1</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>1</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>弱密码检测</TypeAlias>
    <SubTypeAlias>Windows系统登录弱口令检测</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Windows系统登录弱口令检测</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>23</RiskId>
    <LastFoundTime>2019-04-10 14:00:17</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>8</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>中间件</TypeAlias>
    <SubTypeAlias>Apache Tomcat 安全基线检查</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Apache Tomcat 安全基线检查</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>2</RiskId>
    <LastFoundTime>2019-04-10 08:31:32</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>弱密码检测</TypeAlias>
```

```
    <SubTypeAlias>PostgreSQL弱密码检测</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>PostgreSql登录弱口令检测</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>11</RiskId>
    <LastFoundTime>2019-04-10 08:31:32</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>弱密码检测</TypeAlias>
    <SubTypeAlias>MySQL弱密码检测</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>MySQL弱密码检测</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>20</RiskId>
    <LastFoundTime>2019-04-10 08:31:32</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>弱密码检测</TypeAlias>
    <SubTypeAlias>FTP登陆弱口令检测</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>FTP登陆弱口令检测</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>17</RiskId>
    <LastFoundTime>2019-04-10 06:58:58</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>1</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>弱密码检测</TypeAlias>
    <SubTypeAlias>Microsoft SQL Server弱密码检测</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Microsoft SQL Server登录弱口令检测</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>24</RiskId>
    <LastFoundTime>2019-04-10 06:52:05</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>2</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>数据库</TypeAlias>
    <SubTypeAlias>Memcached安全基线检查</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Memcached安全基线检查</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>42</RiskId>
    <LastFoundTime>2019-02-19 14:59:17</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>16</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>high</Level>
```

```
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>CentOS Linux 6安全基线检查</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>CentOS Linux 6安全基线检查</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>49</RiskId>
    <LastFoundTime>2019-02-19 14:59:17</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>12</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>CentOS Linux 6合规基线检查-等保二级</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>CentOS Linux 6合规基线检查-等保二级</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>50</RiskId>
    <LastFoundTime>2019-02-19 14:59:17</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>15</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>CentOS Linux 6合规基线检查-等保三级</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>CentOS Linux 6合规基线检查-等保三级</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>55</RiskId>
    <LastFoundTime>2019-02-19 12:57:23</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>14</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>Linux Ubuntu合规基线检查-等保二级</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Linux Ubuntu合规基线检查-等保二级</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>56</RiskId>
    <LastFoundTime>2019-02-19 12:57:23</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>15</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>medium</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>Linux Ubuntu合规基线检查-等保三级</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Linux Ubuntu合规基线检查-等保三级</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>54</RiskId>
    <LastFoundTime>2019-02-19 12:57:23</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>16</CheckCount>
    <HighWarningCount>0</HighWarningCount>
```

```
    <Level>high</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>Linux Ubuntu 安全基线检查</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Linux Ubuntu 安全基线检查</RiskName>
  </WarningSummarys>
  <WarningSummarys>
    <RiskId>51</RiskId>
    <LastFoundTime>2019-02-19 12:44:58</LastFoundTime>
    <MediumWarningCount>0</MediumWarningCount>
    <WarningMachineCount>0</WarningMachineCount>
    <CheckCount>12</CheckCount>
    <HighWarningCount>0</HighWarningCount>
    <Level>high</Level>
    <TypeAlias>系统</TypeAlias>
    <SubTypeAlias>Windows 2012 R2安全基线检查</SubTypeAlias>
    <LowWarningCount>0</LowWarningCount>
    <RiskName>Windows 2012 R2安全基线检查</RiskName>
  </WarningSummarys>
</DescribeCheckWarningSummary>
```

JSON 格式

```
{
 "Count":20,
 "TotalCount":25,
 "WarningSummarys":[
  {
   "SubTypeAlias":"CentOS Linux 7安全基线检查\n
       \n                                  ",
   "HighWarningCount":8,
   "WarningMachineCount":3,
   "RiskName":"CentOS Linux 7安全基线检查\n
   \n                                  ",
   "LowWarningCount":0,
   "CheckCount":16,
   "RiskId":43,
   "Level":"high",
   "MediumWarningCount":0,
   "TypeAlias":"系统",
   "LastFoundTime":"2019-04-10 00:33:00"
  },
  {
   "SubTypeAlias":"CentOS Linux 7合规基线检查-等保三级",
   "HighWarningCount":7,
   "WarningMachineCount":2,
   "RiskName":"CentOS Linux 7合规基线检查-等保三级",
   "LowWarningCount":0,
   "CheckCount":15,
   "RiskId":47,
   "Level":"medium",
   "MediumWarningCount":0,
   "TypeAlias":"系统",
   "LastFoundTime":"2019-04-10 00:58:11"
  },
  {
   "SubTypeAlias":"CentOS Linux 7合规基线检查-等保二级",
   "HighWarningCount":6,
   "WarningMachineCount":2,
   "RiskName":"CentOS Linux 7合规基线检查-等保二级",
   "LowWarningCount":0,
   "CheckCount":12,
   "RiskId":48,
```

```
    "Level":"medium",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-04-10 00:58:11"
   },
   {
    "SubTypeAlias":"Redis安全基线检查",
    "HighWarningCount":1,
    "WarningMachineCount":1,
    "RiskName":"Redis安全基线检查",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":3,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"数据库",
    "LastFoundTime":"2019-04-10 08:31:32"
   },
   {
    "SubTypeAlias":"Linux系统登录弱口令检测",
    "HighWarningCount":1,
    "WarningMachineCount":1,
    "RiskName":"Linux系统登录弱口令检测",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":19,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"弱密码检测",
    "LastFoundTime":"2019-04-10 08:31:32"
   },
   {
    "SubTypeAlias":"FTP匿名登录配置检测",
    "HighWarningCount":1,
    "WarningMachineCount":1,
    "RiskName":"FTP匿名登录配置检测",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":12,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"弱密码检测",
    "LastFoundTime":"2019-04-10 08:31:32"
   },
   {
    "SubTypeAlias":"Windows系统登录弱口令检测",
    "HighWarningCount":1,
    "WarningMachineCount":1,
    "RiskName":"Windows系统登录弱口令检测",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":13,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"弱密码检测",
    "LastFoundTime":"2019-04-10 06:58:58"
   },
   {
    "SubTypeAlias":"Apache Tomcat 安全基线检查",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"Apache Tomcat 安全基线检查",
    "LowWarningCount":0,
    "CheckCount":8,
```

```
    "RiskId":23,
    "Level":"medium",
    "MediumWarningCount":0,
    "TypeAlias":"中间件",
    "LastFoundTime":"2019-04-10 14:00:17"
   },
   {
    "SubTypeAlias":"PostgreSQL弱密码检测",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"PostgreSql登录弱口令检测",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":2,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"弱密码检测",
    "LastFoundTime":"2019-04-10 08:31:32"
   },
   {
    "SubTypeAlias":"MySQL弱密码检测",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"MySQL弱密码检测",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":11,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"弱密码检测",
    "LastFoundTime":"2019-04-10 08:31:32"
   },
   {
    "SubTypeAlias":"FTP登陆弱口令检测",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"FTP登陆弱口令检测",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":20,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"弱密码检测",
    "LastFoundTime":"2019-04-10 08:31:32"
   },
   {
    "SubTypeAlias":"Microsoft SQL Server弱密码检测",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"Microsoft SQL Server登录弱口令检测",
    "LowWarningCount":0,
    "CheckCount":1,
    "RiskId":17,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"弱密码检测",
    "LastFoundTime":"2019-04-10 06:58:58"
   },
   {
    "SubTypeAlias":"Memcached安全基线检查",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"Memcached安全基线检查",
    "LowWarningCount":0,
```

```
    "CheckCount":2,
    "RiskId":24,
    "Level":"medium",
    "MediumWarningCount":0,
    "TypeAlias":"数据库",
    "LastFoundTime":"2019-04-10 06:52:05"
   },
   {
    "SubTypeAlias":"CentOS Linux 6安全基线检查",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"CentOS Linux 6安全基线检查",
    "LowWarningCount":0,
    "CheckCount":16,
    "RiskId":42,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-02-19 14:59:17"
   },
   {
    "SubTypeAlias":"CentOS Linux 6合规基线检查-等保二级",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"CentOS Linux 6合规基线检查-等保二级",
    "LowWarningCount":0,
    "CheckCount":12,
    "RiskId":49,
    "Level":"medium",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-02-19 14:59:17"
   },
   {
    "SubTypeAlias":"CentOS Linux 6合规基线检查-等保三级",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"CentOS Linux 6合规基线检查-等保三级",
    "LowWarningCount":0,
    "CheckCount":15,
    "RiskId":50,
    "Level":"medium",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-02-19 14:59:17"
   },
   {
    "SubTypeAlias":"Linux Ubuntu合规基线检查-等保二级",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"Linux Ubuntu合规基线检查-等保二级",
    "LowWarningCount":0,
    "CheckCount":14,
    "RiskId":55,
    "Level":"medium",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-02-19 12:57:23"
   },
   {
    "SubTypeAlias":"Linux Ubuntu合规基线检查-等保三级",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"Linux Ubuntu合规基线检查-等保三级",
```

```
    "LowWarningCount":0,
    "CheckCount":15,
    "RiskId":56,
    "Level":"medium",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-02-19 12:57:23"
   },
   {
    "SubTypeAlias":"Linux Ubuntu 安全基线检查",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"Linux Ubuntu 安全基线检查",
    "LowWarningCount":0,
    "CheckCount":16,
    "RiskId":54,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-02-19 12:57:23"
   },
   {
    "SubTypeAlias":"Windows 2012 R2安全基线检查",
    "HighWarningCount":0,
    "WarningMachineCount":0,
    "RiskName":"Windows 2012 R2安全基线检查",
    "LowWarningCount":0,
    "CheckCount":12,
    "RiskId":51,
    "Level":"high",
    "MediumWarningCount":0,
    "TypeAlias":"系统",
    "LastFoundTime":"2019-02-19 12:44:58"
   }
  ],
  "PageSize":20,
  "RequestId":"DFA6CDC5-E826-4D18-A499-BEF9DA31F1AD",
  "CurrentPage":1
}
```


### 错误码

访问错误中心查看更多错误码。

## 7.2 DescribeStratety

获取基线检查策略的设置内容。

调用DescribeStratety接口可获取基线扫描策略的设置详情。

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeStratety | 系统规定参数。<br>取值：DescribeStratety |
| Lang | String | 否 | zh | 语言。 |
| SourceIp | String | 否 | 127.0.0.1 | 来源。 |
| StrategyIds | String | 否 | 1,2,3 | 指定策略ID，多个用逗号分隔。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 16B9826C-B99F-4F8A-8048-EA81D6D3DE8B | 请求ID。 |
| Strategies | | | 策略列表。 |
| ConfigTargets | | | 关联分组。 |
| Flag | String | add | 关联标识。 |
| Target | String | 3259405 | 关联分组ID。 |
| TargetType | String | groupId | 关联类型。 |
| CycleDays | Integer | 1 | 基线检查周期。<br>可选项：<br>-每隔1天<br>-每隔3天<br>-每隔7天<br>-每隔30天 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CycleStartTime | Integer | 6 | 基线检查开始时间，可在0:00、06:00、12:00、18:00四个时间点开始执行基线检查。 |
| EcsCount | Integer | 5 | 该基线检查策略中执行基线检测的服务器数量。 |
| ExecStatus | Integer | 1 | 执行状态。<br>· 1：未执行<br>· 2：执行中 |
| Id | Integer | 212635 | 策略ID。 |
| Name | String | 测试 | 策略名称。 |
| PassRate | Integer | 80 | 上一次检测通过率。 |
| ProcessRate | Integer | 80 | 基线检查进度。 |
| RiskCount | Integer | 5 | 选择的基线检查项数量。 |
| Type | Integer | 1 | 基线检查策略的类型。<br>· 1：默认策略<br>· 2：用户自定义的策略 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeStratety
?Lang=zh
&SourceIp=127.0.0.1
&StrategyIds=1,2,3
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeStratety>
  <RequestId>ACAFE09B-3346-433B-BDA8-369AB3D9B4F9</RequestId>
  <Strategies>
```

```
    <CycleStartTime>6</CycleStartTime>
    <EcsCount>0</EcsCount>
    <Type>2</Type>
    <ExecStatus>1</ExecStatus>
    <ProcessRate>0</ProcessRate>
    <CycleDays>3</CycleDays>
    <Id>15161</Id>
    <ConfigTargets>
      <Target>4361221</Target>
      <TargetType>groupId</TargetType>
      <Flag>add</Flag>
    </ConfigTargets>
    <RiskCount>1</RiskCount>
    <Name>test</Name>
  </Strategies>
  <Strategies>
    <CycleStartTime>0</CycleStartTime>
    <EcsCount>17</EcsCount>
    <PassRate>81</PassRate>
    <Type>1</Type>
    <ExecStatus>1</ExecStatus>
    <ProcessRate>0</ProcessRate>
    <CycleDays>1</CycleDays>
    <Id>20</Id>
    <ConfigTargets>
      <Target>281801</Target>
      <TargetType>groupId</TargetType>
      <Flag>del</Flag>
    </ConfigTargets>
    <ConfigTargets>
      <Target>4790960</Target>
      <TargetType>groupId</TargetType>
      <Flag>del</Flag>
    </ConfigTargets>
    <RiskCount>13</RiskCount>
    <Name>默认策略</Name>
  </Strategies>
</DescribeStratety>
```

JSON 格式

```
{
 "RequestId":"ACAFE09B-3346-433B-BDA8-369AB3D9B4F9",
 "Strategies":[
  {
   "Name":"test",
   "RiskCount":1,
   "CycleDays":3,
   "Type":2,
   "CycleStartTime":6,
   "ConfigTargets":[
    {
     "Flag":"add",
     "Target":"4361221",
     "TargetType":"groupId"
    }
   ],
   "ProcessRate":0,
   "Id":15161,
   "ExecStatus":1,
   "EcsCount":0
  },
  {
```

```
    "Name":"默认策略",
    "RiskCount":13,
    "PassRate":81,
    "CycleDays":1,
    "Type":1,
    "CycleStartTime":0,
    "ConfigTargets":[
     {
      "Flag":"del",
      "Target":"281801",
      "TargetType":"groupId"
     },
     {
      "Flag":"del",
      "Target":"4790960",
      "TargetType":"groupId"
     }
    ],
    "ProcessRate":0,
    "Id":20,
    "ExecStatus":1,
    "EcsCount":17
   }
  ]
}
```


### 错误码

访问错误中心查看更多错误码。

## 7.3 DescribeStrategyExecDetail

查询最近一次基线检查的结果详情。

调用DescribeStrategyExecDetail接口可查询某个基线检测策略执行最近一次检查的结果详情，包括最近一次执行检查的时间、

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeStrategyExecDetail | 需要执行的操作。<br>取值：DescribeStrategyExecDetail |
| StrategyId | Integer | 是 | 1 | 基线检测策略ID。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| SourceIp | String | 否 | 127.0.0.1 | 来源IP。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| EndTime | String | 2019-01-08 20:11:20 | 基线检查执行结束时间。 |
| FailCount | Integer | 94 | 基线检查未通过的风险项数量。 |
| FailedEcsList | | | 检测出基线风险项的服务器列表。 |
| IP | String | 1.1.1.1 | 执行基线检查的服务器实例IP地址。 |
| InstanceName | String | 测试-20180703 | 实例名。 |
| IntranetIp | String | 1.1.1.1 | 内网IP。 |
| Reason | String | Detect timeout | 基线检查未通过的原因。 |
| InProcessCount | Integer | 0 | 状态为执行中的基线检查任务的个数。 |
| Percent | String | 100% | 执行进度。 |
| RequestId | String | 09322632-4668-4AD9-BD0D-32757DEFBBA6 | 请求ID。 |
| Source | String | Manual | 执行来源。 |
| StartTime | String | 2019-01-08 19:41:12 | 开始时间。 |
| SuccessCount | Integer | 81 | 基线检查状态为已通过的风险项数量。 |

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribeStrategyExecDetail
&StrategyId=1
```

```
&<公共请求参数>
```

**正常返回示例**

`XML` **格式**

```
<DescribeStrategyExecDetail>
  <RequestId>6489FA6A-D241-4380-9D8C-A8804484CA40</RequestId>
  <EndTime>2019-04-10 00:58:30</EndTime>
  <Percent>100%</Percent>
  <FailCount>1</FailCount>
  <StartTime>2019-04-10 00:19:01</StartTime>
  <SuccessCount>1</SuccessCount>
  <Source>Schedule</Source>
  <FailedEcsList>
    <InstanceName>health-check002</InstanceName>
    <IP>1.1.1.1</IP>
    <IntranetIp>172.1.1.1</IntranetIp>
    <Reason>Agent offline</Reason>
  </FailedEcsList>
  <InProcessCount>0</InProcessCount>
</DescribeStrategyExecDetail>
```

`JSON` **格式**

```
{
 "Source":"Schedule",
 "InProcessCount":0,
 "SuccessCount":1,
 "RequestId":"6489FA6A-D241-4380-9D8C-A8804484CA40",
 "Percent":"100%",
 "FailedEcsList":[
  {
   "IntranetIp":"172.1.1.1",
   "IP":"1.1.1.1",
   "InstanceName":"health-check002",
   "Reason":"Agent offline"
  }
 ],
 "EndTime":"2019-04-10 00:58:30",
 "StartTime":"2019-04-10 00:19:01",
 "FailCount":1
}
```

**错误码**

**访问错误中心查看更多错误码。**

## 7.4 DescribeCheckWarnings

**查询指定风险项和指定服务器下的检查项列表。**

**调用DescribeCheckWarnings接口可查询指定风险项信息和指定的服务器下的检查项列表。**

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeCheckWarnings | 系统规定参数。取值：DescribeCheckWarnings。 |
| RiskId | Long | 是 | 1 | 风险项ID。 |
| Uuid | String | 是 | d42f938c-d962-48a0-90f9-05********** | 执行基线检查的服务器的ID。 |
| CurrentPage | Integer | 否 | 1 | 分页页码。<br>起始值为1，表示第1个分页。默认值为1，表示默认展示第1个分页。 |
| Lang | String | 否 | zh | 语言。<br>· zh：中文<br>· en：英文 |
| PageSize | Integer | 否 | 20 | 分页的数量。<br>默认值为20，代表系统默认创建20个分页。 |
| SourceIp | String | 否 | 127.0.0.1 | 来源ID。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| CheckWarnings | | | 检查项列表。 |
| CheckId | Long | 1 | 检查项ID。 |
| CheckWarningId | Long | 10 | 告警数据ID。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Item | String | 密码到期警告 | 检查项名称。 |
| Level | String | high | 检查项级别。 |
| Status | Integer | 1 | 检查项状态。<br>· 1：基线检查未通过<br>· 2：基线修复验证中<br>· 3：基线检查已通过<br>· 5：基线检查状态已失效<br>· 6：基线检查项已忽略 |
| Type | String | 身份鉴别 | 检查项类型。 |
| Uuid | String | d42f938c-d962-48a0-90f9-********** | 执行基线检查的服务器的ID。 |
| Count | Integer | 10 | 当前条数。 |
| CurrentPage | Integer | 1 | 分页页码。 |
| PageSize | Integer | 20 | 分页的数量。 |
| RequestId | String | 0DFCADBA-7065-42DA-AF17-6868B9C2A8CF | 请求ID。 |
| TotalCount | Integer | 100 | 总数。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribeCheckWarnings
&RiskId=1
&Uuid=d42f938c-d962-48a0-90f9-05e4eaf92e34
&Lang=zh
&SourceIp=127.0.0.1
&CurrentPage=1
&PageSize=20
&<公共请求参数>
```

**正常返回示例**

XML 格式

```
<DescribeCheckWarnings>
  <TotalCount>16</TotalCount>
  <CheckWarnings>
    <Status>6</Status>
    <Item>检查系统空密码账户

                                  </Item>
    <Type>身份鉴别</Type>
    <Uuid>974af549-3248-44dd-9180-************</Uuid>
    <CheckId>1</CheckId>
    <CheckWarningId>1393768</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>密码复杂度检查</Item>
    <Type>身份鉴别</Type>
    <Uuid>974af549-3248-44dd-9180-************</Uuid>
    <CheckId>52</CheckId>
    <CheckWarningId>1393774</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>确保root是唯一的UID为0的帐户</Item>
    <Type>身份鉴别</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>15</CheckId>
    <CheckWarningId>1393773</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>开启地址空间布局随机化</Item>
    <Type>入侵防范</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>14</CheckId>
    <CheckWarningId>1393772</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>设置用户权限配置文件的权限</Item>
    <Type>文件权限</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>13</CheckId>
    <CheckWarningId>1393767</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>访问控制配置文件的权限设置</Item>
    <Type>文件权限</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>12</CheckId>
    <CheckWarningId>1393771</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>确保SSH LogLevel设置为INFO</Item>
```

```
    <Type>服务配置</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>11</CheckId>
    <CheckWarningId>1393766</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>确保rsyslog服务已启用</Item>
    <Type>安全审计</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>10</CheckId>
    <CheckWarningId>1393770</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>设置SSH空闲超时退出时间</Item>
    <Type>服务配置</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>8</CheckId>
    <CheckWarningId>1393765</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>SSHD强制使用V2安全协议</Item>
    <Type>服务配置</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>7</CheckId>
    <CheckWarningId>1393764</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>确保SSH MaxAuthTries设置为3到6之间</Item>
    <Type>服务配置</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>6</CheckId>
    <CheckWarningId>1393763</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>确保密码到期警告天数为7或更多</Item>
    <Type>身份鉴别</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>5</CheckId>
    <CheckWarningId>1393769</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>设置密码修改最小间隔时间</Item>
    <Type>身份鉴别</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>4</CheckId>
    <CheckWarningId>1393761</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>设置密码失效时间</Item>
    <Type>身份鉴别</Type>
```

```
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>3</CheckId>
    <CheckWarningId>1393760</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>禁止SSH空密码用户登录


                              </Item>
    <Type>服务配置</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>2</CheckId>
    <CheckWarningId>1393762</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <CheckWarnings>
    <Status>6</Status>
    <Item>检查密码重用是否受限制</Item>
    <Type>身份鉴别</Type>
    <Uuid>974af549-3248-44dd-9180-1bd7c9f60bd5</Uuid>
    <CheckId>58</CheckId>
    <CheckWarningId>1393775</CheckWarningId>
    <Level>high</Level>
  </CheckWarnings>
  <PageSize>20</PageSize>
  <RequestId>C1E6C4FE-DE00-4B75-A01E-FCAB55A36449</RequestId>
  <CurrentPage>1</CurrentPage>
  <Count>16</Count>
</DescribeCheckWarnings>
```

JSON 格式

```
{
 "Count":16,
 "TotalCount":16,
 "PageSize":20,
 "RequestId":"C1E6C4FE-DE00-4B75-A01E-FCAB55A36449",
 "CurrentPage":1,
 "CheckWarnings":[
  {
   "Uuid":"974af549-3248-44dd-9180-************",
   "Status":6,
   "CheckWarningId":1393768,
   "Item":"检查系统空密码账户\r\n                                \r\n
                        ",
   "Type":"身份鉴别",
   "Level":"high",
   "CheckId":1
  },
  {
   "Uuid":"974af549-3248-44dd-9180-************",
   "Status":6,
   "CheckWarningId":1393774,
   "Item":"密码复杂度检查",
   "Type":"身份鉴别",
   "Level":"high",
   "CheckId":52
  },
  {
   "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
   "Status":6,
```

```
    "CheckWarningId":1393773,
    "Item":"确保root是唯一的UID为0的帐户",
    "Type":"身份鉴别",
    "Level":"high",
    "CheckId":15
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393772,
    "Item":"开启地址空间布局随机化",
    "Type":"入侵防范",
    "Level":"high",
    "CheckId":14
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393767,
    "Item":"设置用户权限配置文件的权限",
    "Type":"文件权限",
    "Level":"high",
    "CheckId":13
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393771,
    "Item":"访问控制配置文件的权限设置",
    "Type":"文件权限",
    "Level":"high",
    "CheckId":12
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393766,
    "Item":"确保SSH LogLevel设置为INFO",
    "Type":"服务配置",
    "Level":"high",
    "CheckId":11
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393770,
    "Item":"确保rsyslog服务已启用",
    "Type":"安全审计",
    "Level":"high",
    "CheckId":10
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393765,
    "Item":"设置SSH空闲超时退出时间",
    "Type":"服务配置",
    "Level":"high",
    "CheckId":8
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393764,
```

```
    "Item":"SSHD强制使用V2安全协议",
    "Type":"服务配置",
    "Level":"high",
    "CheckId":7
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393763,
    "Item":"确保SSH MaxAuthTries设置为3到6之间",
    "Type":"服务配置",
    "Level":"high",
    "CheckId":6
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393769,
    "Item":"确保密码到期警告天数为7或更多",
    "Type":"身份鉴别",
    "Level":"high",
    "CheckId":5
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393761,
    "Item":"设置密码修改最小间隔时间",
    "Type":"身份鉴别",
    "Level":"high",
    "CheckId":4
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393760,
    "Item":"设置密码失效时间",
    "Type":"身份鉴别",
    "Level":"high",
    "CheckId":3
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393762,
    "Item":"禁止SSH空密码用户登录                                    \r\n
                        \r\n                                    \r\n
                      ",
    "Type":"服务配置",
    "Level":"high",
    "CheckId":2
   },
   {
    "Uuid":"974af549-3248-44dd-9180-1bd7c9f60bd5",
    "Status":6,
    "CheckWarningId":1393775,
    "Item":"检查密码重用是否受限制",
    "Type":"身份鉴别",
    "Level":"high",
    "CheckId":58
   }
  ]
}
```

### 错误码

访问错误中心查看更多错误码。

## 7.5 DescribeCheckWarningDetail

查询指定检查项明细信息。

调用DescribeCheckWarningDetail接口可查询指定检查项的明细信息。

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeCheckWarningDetail | 系统规定参数。<br>取值：DescribeCheckWarningDetail |
| CheckWarningId | Long | 是 | 1 | 检查项告警ID。 |
| Lang | String | 否 | zh | 请求的语言类型。<br>· zh：中文<br>· en：英文 |
| SourceIp | String | 否 | 127.0.0.1 | 来源IP。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Advice | String | 立即修复 | 基线检查风险项的加固建议。 |
| CheckId | Long | 1 | 检查项ID。 |
| Description | String | 密码即将过期 | 基线检查风险项的补充描述内容。 |
| Item | String | 密码到期警告 | 检查项名称。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Level | String | high | 风险等级：<br>包含：<br>· 高：红色高亮显示，表示风险等级高。<br>· 中：橙色高亮显示，表示风险等级为中等。<br>· 低：灰色高亮显示，表示风险等级低。 |
| Prompt | String | 密码到期警告 | 基线检查风险项的检查提示。 |
| RequestId | String | 09969D2C-4FAD-429E-BFBF-9A60DEF8BF6F | 请求ID。 |
| Type | String | 身份鉴别 | 基线检查项的类型。<br>有哪些类型? |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeCheckWarningDetail
&CheckWarningId=1
&SourceIp=127.0.0.1
&Lang=zh
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribeCheckWarningDetail>
  <Item>设置用户权限配置文件的权限</Item>
  <Type>文件权限</Type>
  <Description>设置用户权限配置文件的权限</Description>
  <RequestId>9F2361B3-B7C3-4189-8DB5-C6D1F67BC225</RequestId>
  <CheckId>13</CheckId>
  <Prompt/>
  <Level>high</Level>
  <Advice>执行以下5条命令
  chown root:root /etc/passwd /etc/shadow /etc/group /etc/gshadow
  chmod 0644 /etc/group
  chmod 0644 /etc/passwd
  chmod 0400 /etc/shadow
 chmod 0400 /etc/gshadow  </Advice>
```

```
</DescribeCheckWarningDetail>
```

JSON 格式

```
{
 "Prompt":"",
 "Description":"设置用户权限配置文件的权限",
 "Item":"设置用户权限配置文件的权限",
 "Type":"文件权限",
 "RequestId":"9F2361B3-B7C3-4189-8DB5-C6D1F67BC225",
 "Advice":"执行以下5条命令\r\nchown root:root /etc/passwd /etc/shadow /
etc/group /etc/gshadow\r\nchmod 0644 /etc/group  \r\nchmod 0644 /etc/
passwd  \r\nchmod 0400 /etc/shadow  \r\nchmod 0400 /etc/gshadow  ",
 "Level":"high",
 "CheckId":13
}
```

### 错误码

访问错误中心查看更多错误码。

## 7.6 DescribeWarningMachines

查询执行了基线检查策略的服务器的信息。

调用DescribeWarningMachines接口可查询执行了基线检查的服务器的信息，包含服务器的ID、服务器名称、检测到的风险项统计数据和风险项状态等信息。

### 调试

前往【API Explorer】在线调试，API Explorer 提供在线调用 API、动态生成 SDK Example 代码和快速检索接口等能力，能显著降低使用云 API 的难度，强烈推荐使用。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeWarningMachines | 系统规定参数。取值：DescribeWarningMachines。 |
| RiskId | Long | 是 | 1 | 风险项ID。 |
| CurrentPage | Integer | 否 | 1 | 执行了基线检查的服务器的风险项页面。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Lang | String | 否 | zh | 请求内容的语言类型。<br>· zh：中文<br>· en：英文 |
| MachineName | String | 否 | 基线测试服务器 | 执行基线检查的服务器的名称。 |
| PageSize | Integer | 否 | 10 | 风险项页面中，每页包含的风险项条数。 |
| SourceIp | String | 否 | 127.0.0.1 | 来源IP。 |
| StrategyId | Long | 否 | 1 | 基线检查策略的ID。 |
| Uuids | String | 否 | xxx-aaa-bbb-ccc | 执行基线检查的服务器ID。多个ID用英文逗号分隔。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 10 | 当前条数 |
| CurrentPage | Integer | 1 | 基线检查风险项当前页的编号。 |
| PageSize | Integer | 10 | 基线检查风险项页面每页的条数。 |
| RequestId | String | 00BD7CE2-284A-4534-BD09-FB69836DD750 | 请求ID。 |
| TotalCount | Integer | 100 | 执行了基线检查的服务器所检测出的风险项总数。 |
| WarningMachines | | | 产生告警的服务器的信息。 |
| HighWarningCount | Integer | 10 | 高危检查项的个数。 |
| InternetIp | String | 47.39.120.22 | 外网IP。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| IntranetIp | String | 127.0.0.1 | 内网IP。 |
| LowWarningCount | Integer | 3 | 低危检查项的个数。 |
| MachineName | String | 基线测试服务器 | 服务器的名称。 |
| MediumWarningCount | Integer | 2 | 中危检查项的个数。 |
| PassCount | Integer | 10 | 通过检测的检查项个数。 |
| Status | Integer | 1 | 基线检查风险项修复完成后，风险项验证的状态。<br>· 1：已完成<br>· 2：验证中 |
| Uuid | String | xxx-aaa-bbb-ccc | 用户ID。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribeWarningMachines
&RiskId=1
&<公共请求参数>
```

**正常返回示例**

`XML` **格式**

```
<DescribeWarningMachines>
  <TotalCount>47</TotalCount>
  <PageSize>20</PageSize>
  <RequestId>7E6F0031-6D83-4FEC-9077-FF87F9D9887C</RequestId>
  <CurrentPage>1</CurrentPage>
  <WarningMachines>
    <Status>1</Status>
    <MediumWarningCount>0</MediumWarningCount>
    <Uuid>6b36bd63-1848-47dd-8b5a-************</Uuid>
    <InternetIp>47.97.192.205</InternetIp>
    <MachineName>测试</MachineName>
    <HighWarningCount>8</HighWarningCount>
    <PassCount>8</PassCount>
    <IntranetIp>192.1.1.1</IntranetIp>
    <LowWarningCount>0</LowWarningCount>
  </WarningMachines>
  <WarningMachines>
```

```
    <Status>1</Status>
    <MediumWarningCount>0</MediumWarningCount>
    <Uuid>f03259d8-1e81-4fae-bcbb-************</Uuid>
    <InternetIp>1.1.1.1</InternetIp>
    <MachineName>health-check002</MachineName>
    <HighWarningCount>8</HighWarningCount>
    <PassCount>8</PassCount>
    <IntranetIp>172.1.1.1</IntranetIp>
    <LowWarningCount>0</LowWarningCount>
  </WarningMachines>
  <WarningMachines>
    <Status>1</Status>
    <MediumWarningCount>0</MediumWarningCount>
    <Uuid>f87e146b-c704-4845-a187-05c37743a614</Uuid>
    <InternetIp>1.1.1.1</InternetIp>
    <MachineName>hht测试白名单数据准确性</MachineName>
    <HighWarningCount>8</HighWarningCount>
    <PassCount>8</PassCount>
    <IntranetIp>172.1.1.1</IntranetIp>
    <LowWarningCount>0</LowWarningCount>
  </WarningMachines>
  <Count>20</Count>
</DescribeWarningMachines>
```

JSON 格式

```
{
 "Count":20,
 "TotalCount":47,
 "PageSize":20,
 "RequestId":"7E6F0031-6D83-4FEC-9077-FF87F9D9887C",
 "WarningMachines":[
  {
   "Uuid":"6b36bd63-1848-47dd-8b5a-************",
   "Status":1,
   "HighWarningCount":8,
   "LowWarningCount":0,
   "IntranetIp":"192.1.1.1",
   "InternetIp":"47.97.192.205",
   "MachineName":"测试",
   "MediumWarningCount":0,
   "PassCount":8
  },
  {
   "Uuid":"f03259d8-1e81-4fae-bcbb-************",
   "Status":1,
   "HighWarningCount":8,
   "LowWarningCount":0,
   "IntranetIp":"172.1.1.1",
   "InternetIp":"1.1.1.1",
   "MachineName":"health-check002",
   "MediumWarningCount":0,
   "PassCount":8
  },
  {
   "Uuid":"f87e146b-c704-4845-a187-05c37743a614",
   "Status":1,
   "HighWarningCount":8,
   "LowWarningCount":0,
   "IntranetIp":"172.1.1.1",
   "InternetIp":"1.1.1.1",
   "MachineName":"hht测试白名单数据准确性",
   "MediumWarningCount":0,
```

```
    "PassCount":8
   }
  ],
  "CurrentPage":1
}
```


### 错误码

访问错误中心查看更多错误码。

# 8 资产管理

## 8.1 DescribeFieldStatistics

调用该接口检索资产实例列表。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeFieldStatistics | 系统规定参数。取值：DescribeFieldStatistics。 |
| MachineTypes | String | 否 | ecs | 指定待检索的资产类型。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| GroupedFields | | | 返回的资产信息列表。 |
| GroupCount | Integer | 20 | 返回的服务器组数量。 |
| InstanceCount | Integer | 100 | 返回的所有资产数量。 |
| NewInstanceCount | Integer | 10 | 返回的新增资产数量。 |
| NotRunningStatusCount | Integer | 10 | 返回的未启动的服务器数量。 |
| RegionCount | Integer | 11 | 返回的服务器地域数量。 |
| RiskInstanceCount | Integer | 90 | 返回的存在风险的资产数量。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| UnprotectedInstanceCount | Integer | 10 | 返回的未受保护的资产数量。 |
| VpcCount | Integer | 5 | 返回的专有网络VPC数量。 |
| RequestId | String | 7E0618A9-D5EF-4220-9471-C42B5E92719F | 返回数据请求是否成功。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeFieldStatistics
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeFieldStatistics>
   <code>200</code>
   <data>
      <GroupedFields>
         <InstanceCount>100</InstanceCount>
         <RiskInstanceCount>90</RiskInstanceCount>
         <UnProtectedInstanceCount>10</UnProtectedInstanceCount>
         <GroupCount>20</GroupCount>
         <RegionCount>11</RegionCount>
         <VpcCount>5</VpcCount>
      </GroupedFields>
   </data>
   <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
   <success>true</success>
    </DescribeFieldStatistics>
```

#### JSON 格式

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "data":{
  "GroupedFields":{
   "InstanceCount":100,
   "GroupCount":20,
   "VpcCount":5,
   "UnProtectedInstanceCount":10,
   "RiskInstanceCount":90,
   "RegionCount":11
  }
 },
 "code":200,
 "success":true
```

```
}
```

### 错误码

访问错误中心查看更多错误码。

## 8.2 DescribeGroupedTags

调用该接口获取标签的统计信息。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeGroupedTags | 系统规定参数。取值：DescribeGroupedTags。 |
| MachineTypes | String | 否 | ecs | 指定待查询的资产类型。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 2 | 返回结果的当前页显示数据条数。 |
| GroupedFileds | | | 返回的标签统计信息列表。 |
| Count | String | 2 | 返回结果中标签对应资产数量。 |
| Name | String | xxx | 返回结果中标签名称。 |
| TagId | Integer | 111 | 返回结果中的标签ID。 |
| HttpStatusCode | Integer | 200 | 返回请求数据结果的状态码。 |
| RequestId | String | 7E0618A9-D5EF-4220-9471-C42B5E92719F | 返回结果的请求ID。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Success | Boolean | true | 返回数据请求是否成功。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeGroupedTags
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeGroupedTags>
   <code>200</code>
   <data>
      <GroupedFileds>
         <Name>xxx</Name>
         <Count>2</Count>
         <TagId>111</TagId>
      </GroupedFileds>
   </data>
   <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
   <success>true</success>
</DescribeGroupedTags>
```

#### JSON 格式

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "data":{
  "GroupedFileds":[
   {
    "Name":"xxx",
    "Count":2,
    "TagId":111
   }
  ]
 },
 "code":200,
 "success":true
}
```

## 错误码

访问错误中心查看更多错误码。

# 8.3 DescribeAllGroups

调用该接口获取所有的分组信息。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeAllGroups | 系统规定参数。取值：DescribeAllGroups。 |
| Lang | String | 否 | zh | 指定返回结果语言环境。<br>· zh：中文<br>· en：英文 |

## 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 2 | 返回结果中显示的当前页码。 |
| Groups | | | 返回的分组信息列表。 |
| GroupFlag | Integer | 1 | 返回结果中分组类型。0为默认分组，1为其他分组。 |
| GroupId | Integer | 100 | 返回结果中的分组ID。 |
| GroupName | String | xx | 返回结果中的分组名称。 |
| RequestId | String | "requestId": "7E0618A9-D5EF-4220-9471-C42B5E92719F", | 返回结果的请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeAllGroups
&<公共请求参数>
```

**正常返回示例**

`XML` **格式**

```
<DescribeAllGroups>
   <code>200</code>
   <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
   <success>true</success>
   <data>
      <Groups>
         <GroupId>100</GroupId>
         <GroupName>xx</GroupName>
      </Groups>
   </data>
    </DescribeAllGroups>
```

`JSON` **格式**

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "data":{
  "Groups":{
   "GroupName":"xx",
   "GroupId":100
  }
 },
 "code":200,
 "success":true
}
```

**错误码**

访问错误中心查看更多错误码。

## 8.4 DeleteGroup

调用该接口删除服务器分组。

**调试**

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DeleteGroup | 系统规定参数。取值：DeleteGroup。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| GroupId | Long | 是 | 11111 | 指定待删除的服务器分组ID。 |
| SourceIp | String | 否 | 127.1.1.1 | 指定访问源IP地址。 |

## 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 11111 | 返回结果的请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DeleteGroup
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DeleteGroup>
   <code>200</code>
   <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
   <success>true</success>
    </DeleteGroup>
```

`JSON` 格式

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "code":200,
 "success":true
}
```

## 错误码

访问错误中心查看更多错误码。

# 8.5 CreateOrUpdateAssetGroup

调用该接口修改资产与分组的关系。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | CreateOrUpdateAssetGroup | 系统规定参数。取值：CreateOrUpdateAssetGroup。 |
| GroupId | Long | 否 | 111111 | 指定待修改分组ID。 |
| GroupName | String | 否 | test | 指定待修改分组名称。 |
| Uuids | String | 否 | [] | UUIDS列表信息。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 7E0618A9-D5EF-4220-9471-C42B5E92719F | 返回结果的请求ID。 |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=CreateOrUpdateAssetGroup
&<公共请求参数>
```

#### 正常返回示例

`XML` 格式

```
<CreateOrUpdateAssetGroup>
    <code>200</code>
    <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
    <success>true</success>
```

```
    </CreateOrUpdateAssetGroup>
```

JSON 格式

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "code":200,
 "success":true
}
```

### 错误码

访问错误中心查看更多错误码。

## 8.6 ModifyTagWithUuid

调用该接口修改标签与服务器或云产品的关系。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | ModifyTagWithUuid | 系统规定参数。取值：ModifyTagWithUuid。 |
| MachineTypes | String | 否 | ecs | 指定待修改的资产类型。 |
| TagId | String | 否 | 7E0618A9-D5EF-4220-9471-C42B5E92719F | 指定待修改标签ID。 |
| TagList | String | 否 | ac,ad | 指定待修改标签列表。 |
| UuidList | String | 否 | 111-xx,aa-bb | 指定待修改机器列表。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | "requestId": "7E0618A9-D5EF-4220-9471-C42B5E92719F", | 返回结果的请求ID。 |

示例

请求示例

```
http(s)://[Endpoint]/?Action=ModifyTagWithUuid
&<公共请求参数>
```

正常返回示例

`XML` 格式

```
<ModifyTagWithUuid>
    <code>200</code>
    <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
    <success>true</success>
     </ModifyTagWithUuid>
```

`JSON` 格式

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "code":200,
 "success":true
}
```

错误码

访问错误中心查看更多错误码。

# 8.7 DescribeInstanceStatistics

调用该接口获取机器的统计信息。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeInstanceStatistics | 系统规定参数。取值：DescribeInstanceStatistics。 |
| From | String | 是 | sas | 指定数据请求来源。固定为sas。 |
| Uuid | String | 是 | ["ff675afd-703e-40dc-809b-0fad8b0ded28"] | 指定待查询的机器列表。 |
| Lang | String | 否 | zh | 指定返回结果语言环境。<br>· zh：中文<br>· en：英文 |
| SourceIp | String | 否 | 127.1.1.1 | 指定访问源IP地址。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Data | | | 返回机器的检测结果项列表。 |
| Account | Integer | 1 | 返回结果中的分组类型。<br>· 0：默认分组<br>· 1：其他分组 |
| AppNum | Integer | 0 | 返回结果中的应用漏洞数量。 |
| CmsNum | Integer | 2 | 返回结果中的Web-CMS漏洞数量。 |
| CveNum | Integer | 1 | 返回结果中的通用漏洞数量。 |
| EmgNum | Integer | 0 | 返回结果中的应急漏洞数量。 |
| Health | Integer | 1 | 返回结果中的基线检查问题数量。 |
| Suspicious | Integer | 2 | 返回结果中的安全告警数量。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| SysNum | Integer | 1 | 返回结果中的系统漏洞数量。 |
| Trojan | Integer | 1 | 返回结果中的木马数量。 |
| Uuid | String | xxx | 返回结果中的机器UUID。 |
| Vul | Integer | 4 | 返回结果中所有漏洞数量总和：CveNum+EmgNum+SysNum+CmsNum+AppNum。 |
| RequestId | String | "requestId": "7E0618A9-D5EF-4220-9471-C42B5E92719F", | 返回结果的请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeInstanceStatistics
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeInstanceStatistics>
   <code>200</code>
   <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
   <success>true</success>
   <data>
      <EntEntityListity>
         <Uuid>xxx</Uuid>
         <Health>1</Health>
         <Suspicious>2</Suspicious>
         <Vul>3</Vul>
      </EntEntityListity>
   </data>
    </DescribeInstanceStatistics>
```

#### JSON 格式

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "data":{
  "EntEntityListity":[
   {
    "Uuid":"xxx",
    "Suspicious":2,
```

```
    "Health":1,
    "Vul":3
   }
  ]
 },
 "code":200,
 "success":true
}
```


### 错误码

访问错误中心查看更多错误码。

## 8.8 DescribeCloudProductFieldStatistics

调用该接口获取云产品统计信息。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeCloudProductFieldStatistics | 系统规定参数。取值：DescribeCloudProductFieldStatistics。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| GroupedFields | | | 返回云产品信息列表。 |
| CategoryCount | String | [{"MachineType":1,"Count":11}] | 返回的不同种类资产数量的列表。 |
| InstanceCount | Integer | 100 | 返回的所有资产数量。 |
| RiskInstanceCount | Integer | 90 | 返回的存在风险的资产的数量。 |
| RequestId | String | "requestId": "7E0618A9-D5EF-4220-9471-C42B5E92719F", | 返回结果的请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeCloudProductFieldStatistics
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribeCloudProductFieldStatistics>
   <code>200</code>
   <data>
      <GroupedFields>
         <InstanceCount>100</InstanceCount>
         <RiskInstanceCount>90</RiskInstanceCount>
         <categoryCount>
            <MachineType>1</MachineType>
            <Count>11</Count>
         </categoryCount>
      </GroupedFields>
   </data>
   <requestId>7E0618A9-D5EF-4220-9471-C42B5E92719F</requestId>
   <success>true</success>
</DescribeCloudProductFieldStatistics>
```

#### JSON 格式

```
{
 "requestId":"7E0618A9-D5EF-4220-9471-C42B5E92719F",
 "data":{
  "GroupedFields":{
   "InstanceCount":100,
   "categoryCount":[
    {
     "MachineType":1,
     "Count":11
    }
   ],
   "RiskInstanceCount":90
  }
 },
 "code":200,
 "success":true
}
```

## 错误码

访问错误中心查看更多错误码。

# 8.9 DescribeDomainCount

调用该接口获取域名资产数量。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeDomainCount | 系统规定参数。取值：DescribeDomainCount。 |
| SourceIp | String | 否 | 127.1.1.1 | 指定的访问源IP地址。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| RequestId | String | 6FAFB857-FE24-4226-A09F-52EA5023C987 | 返回结果的请求ID。 |
| RootDomainsCount | Integer | 27 | 返回的根网站数量。 |
| TotalDomainsCount | Integer | 1 | 返回的所有网站数量。 |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeDomainCount
&<公共请求参数>
```

#### 正常返回示例

`XML` 格式

```
<DescribeDomainCount>
    <code>200</code>
    <data>
        <TotalDomainsCount>27</TotalDomainsCount>
        <RootDomainsCount>1</RootDomainsCount>
    </data>
```

```
    <requestId>6FAFB857-FE24-4226-A09F-52EA5023C987</requestId>
    <success>true</success>
     </DescribeDomainCount>
```

JSON 格式

```
{
 "requestId":"6FAFB857-FE24-4226-A09F-52EA5023C987",
 "data":{
  "TotalDomainsCount":27,
  "RootDomainsCount":1
 },
 "code":200,
 "success":true
}
```

### 错误码

访问错误中心查看更多错误码。

## 8.10 DescribeDomainList

调用该接口域名资产列表。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeDomainList | 系统规定参数。取值：DescribeDomainList。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| DomainType | String | 否 | root | 指定待查询的域名类型。 |
| FuzzyDomain | String | 否 | sas | 指定的域名模糊匹配搜索信息。 |
| PageSize | Integer | 否 | 1 | 指定列表每页显示数据条数。 |
| SourceIp | String | 否 | 127.1.1.1 | 指定访问源IP地址。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| DomainList ResponseList | | | 返回的域名列表。 |
| Domain | String | tst.com | 返回的域名名称或网站名称。 |
| IpList | String | 0.0.0.0，0.0.0.0 | 返回的域名对应域名IP列表信息。 |
| PageInfo | | | 返回结果的页面显示信息。 |
| Count | Integer | 10 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 10 | 返回结果中每页显示数据条数。 |
| TotalCount | Integer | 27 | 返回数据的总条数。 |
| RequestId | String | 0B48AB3C-84FC-424D-A01D-B9270EF46038 | 返回结果的请求ID。 |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeDomainList
&<公共请求参数>
```

#### 正常返回示例

`XML` 格式

```
<DescribeDomainList>
   <code>200</code>
   <data>
      <PageInfo>
         <TotalCount>27</TotalCount>
         <PageSize>10</PageSize>
         <CurrentPage>1</CurrentPage>
         <Count>10</Count>
      </PageInfo>
      <DomainListResponseList>
         <Domain>sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>p.sastst.com</Domain>
```

```
      </DomainListResponseList>
      <DomainListResponseList>
         <IpList>120.27.11.134,47.111.33.130,120.79.72.16,139.129.103.
215</IpList>
         <Domain>b.sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>r.sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>t.sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>k.sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>{.sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>w.sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>l.sastst.com</Domain>
      </DomainListResponseList>
      <DomainListResponseList>
         <Domain>v.sastst.com</Domain>
      </DomainListResponseList>
   </data>
   <requestId>0B48AB3C-84FC-424D-A01D-B9270EF46038</requestId>
   <success>true</success>
    </DescribeDomainList>
```

JSON 格式

```
{
 "requestId":"0B48AB3C-84FC-424D-A01D-B9270EF46038",
 "data":{
  "DomainListResponseList":[
   {
    "Domain":"sastst.com"
   },
   {
    "Domain":"p.sastst.com"
   },
   {
    "Domain":"b.sastst.com",
    "IpList":"120.27.11.134,47.111.33.130,120.79.72.16,139.129.103.215
"
   },
   {
    "Domain":"r.sastst.com"
   },
   {
    "Domain":"t.sastst.com"
   },
   {
    "Domain":"k.sastst.com"
   },
   {
    "Domain":"{.sastst.com"
   },
   {
    "Domain":"w.sastst.com"
```

```
    },
    {
     "Domain":"l.sastst.com"
    },
    {
     "Domain":"v.sastst.com"
    }
   ],
   "PageInfo":{
    "Count":10,
    "TotalCount":27,
    "PageSize":10,
    "CurrentPage":1
   }
  },
  "code":200,
  "success":true
}
```


**错误码**

访问错误中心查看更多错误码。

## 8.11 DescribeDomainDetail

调用该接口获取域名资产详情。

**调试**

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribeDomainDetail | 系统规定参数。取值：DescribeDomainDetail。 |
| DomainName | String | 是 | 0.0.0.0 | 指定待查询的域名名称或网站名称。 |
| SourceIp | String | 否 | 127.1.1.1 | 指定访问源IP地址。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Domain | String | b.tst.com | 返回的域名名称。 |
| DomainDetailItems | | | 返回的域名相关的资产列表。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| AssetType | String | 0 | 返回结果中域名下资产的资产类型。<br>· 0：ECS<br>· 1：负载均衡<br>· 2：NAT网关<br>· 3：RDS数据库<br>· 4：MongoDb数据库 |
| InstanceId | String | i-m5e6w7dzsktt6mz4yime | 返回结果中的资产ID。 |
| InstanceName | String | iZm5e6w7dzsktt6mz4yimeZ-60288 | 返回结果中的资产名称。 |
| InternetIp | String | 0.0.0.0 | 返回结果中资产对应的公网IP地址。 |
| IntranetIp | String | 0.0.0.0 | 返回结果中资产对应的私网IP地址。 |
| MachineIp | String | 0.0..0.0 | 返回的资产IP。 |
| Uuid | String | lb-bp1g9dohoyin9cjhn6itt | 返回结果中的资产UUID。 |
| RequestId | String | 3A85CFCF-05C8-451A-9E41-C0D5E96BA407 | 返回结果的请求ID。 |
| RootDomain | String | tst.com | 返回结果中域名对应的根域名名称。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribeDomainDetail
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribeDomainDetail>
   <code>200</code>
   <data>
      <DomainDetailItems>
         <InstanceName>iZm5e6w7dzsktt6mz4yimeZ-602888</InstanceName>
         <AssetType>0</AssetType>
```

```
          <Uuid>d664e15e-488b-4f3c-a80f-25e3560d6085</Uuid>
          <InternetIp>139.129.103.215</InternetIp>
          <InstanceId>i-m5e6w7dzsktt6mz4yime</InstanceId>
          <IntranetIp>172.31.13.43</IntranetIp>
       </DomainDetailItems>
       <DomainDetailItems>
          <InstanceName>iZm5ehn8ugwvsp2k4xdzokZ</InstanceName>
          <AssetType>0</AssetType>
          <Uuid>0d804403-8d36-4642-a5f1-a22905491c94</Uuid>
          <InternetIp>120.27.11.134</InternetIp>
          <InstanceId>i-m5ehn8ugwvsp2k4xdzok</InstanceId>
          <IntranetIp>10.30.183.121</IntranetIp>
       </DomainDetailItems>
       <DomainDetailItems>
          <AssetType>2</AssetType>
          <Uuid>eip-wz98lk3q06mhqvbhcuvb5</Uuid>
          <InternetIp>120.79.72.16</InternetIp>
          <InstanceId>eip-wz98lk3q06mhqvbhcuvb5</InstanceId>
       </DomainDetailItems>
       <DomainDetailItems>
          <InstanceName>郝浩天测试</InstanceName>
          <AssetType>1</AssetType>
          <Uuid>lb-bp1g9dohoyin9cjhn6itt</Uuid>
          <InternetIp>47.111.33.130</InternetIp>
          <InstanceId>lb-bp1g9dohoyin9cjhn6itt</InstanceId>
       </DomainDetailItems>
       <RootDomain>sastst.com</RootDomain>
       <Domain>b.sastst.com</Domain>
    </data>
    <requestId>3A85CFCF-05C8-451A-9E41-C0D5E96BA407</requestId>
    <success>true</success>
     </DescribeDomainDetail>
```

**JSON 格式**

```
{
 "requestId":"3A85CFCF-05C8-451A-9E41-C0D5E96BA407",
 "data":{
  "RootDomain":"sastst.com",
  "Domain":"b.sastst.com",
  "DomainDetailItems":[
   {
    "Uuid":"d664e15e-488b-4f3c-a80f-25e3560d6085",
    "AssetType":"0",
    "IntranetIp":"172.31.13.43",
    "InstanceId":"i-m5e6w7dzsktt6mz4yime",
    "InternetIp":"139.129.103.215",
    "InstanceName":"iZm5e6w7dzsktt6mz4yimeZ-602888"
   },
   {
    "Uuid":"0d804403-8d36-4642-a5f1-a22905491c94",
    "AssetType":"0",
    "IntranetIp":"10.30.183.121",
    "InstanceId":"i-m5ehn8ugwvsp2k4xdzok",
    "InternetIp":"120.27.11.134",
    "InstanceName":"iZm5ehn8ugwvsp2k4xdzokZ"
   },
   {
    "Uuid":"eip-wz98lk3q06mhqvbhcuvb5",
    "AssetType":"2",
    "InstanceId":"eip-wz98lk3q06mhqvbhcuvb5",
    "InternetIp":"120.79.72.16"
   },
```

```
    {
     "Uuid":"lb-bp1g9dohoyin9cjhn6itt",
     "AssetType":"1",
     "InstanceId":"lb-bp1g9dohoyin9cjhn6itt",
     "InternetIp":"47.111.33.130",
     "InstanceName":"郝浩天测试"
    }
   ]
  },
  "code":200,
  "success":true
}
```


**错误码**

访问错误中心查看更多错误码。

# 9 资产指纹

## 9.1 DescribePropertyCount

调用该接口获取资产指纹，即进程、端口、软件、账户4种类型的统计数量。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertyCount | 系统规定参数。取值：DescribePropertyCount。 |
| Type | String | 否 | port | 指定待查询的指纹类型。可选：<br>· user：账户<br>· software：软件<br>· process：进程<br>· port：端口<br>说明:<br>类型不填表示获取所有类型信息。 |
| UuidList | String | 否 | [] | 指定待查询的资产UUID。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Port | Integer | 163 | 返回的端口数量信息。 |
| Process | Integer | 367 | 返回的进程数量信息。 |
| RequestId | String | 7E0618A9-D5EF-4220-9471-C42B5E92719F | 返回结果的请求ID。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Software | Integer | 5073 | 返回的软件数量信息。 |
| User | Integer | 114 | 返回的账户数量信息。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribePropertyCount
&<公共请求参数>
```

**正常返回示例**

`XML` **格式**

```
<DescribePropertyCount>
    <data>
       <Process>367</Process>
       <Software>5037</Software>
       <Port>163</Port>
       <User>114</User>
    </data>
     </DescribePropertyCount>
```

`JSON` **格式**

```
{
 "data":{
  "User":114,
  "Software":5037,
  "Port":163,
  "Process":367
 }
}
```

**错误码**

访问错误中心查看更多错误码。

## 9.2 DescribePropertyPortDetail

调用该接口查询端口列表中一个端口的详细信息。

**调试**

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertyPortDetail | 系统规定参数。取值：DescribePropertyPortDetail。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| PageSize | Integer | 否 | 6 | 指定列表每页显示数据条数。 |
| Port | String | 否 | 22 | 指定待查询的端口信息。 |
| ProcName | String | 否 | sshd | 指定待查询的进程名称。 |
| Remark | String | 否 | 0.0.0.0 | 服务器名称或IP。 |
| Uuid | String | 否 | 50d213b4-3a35-427a-b8a5-04b0c7e1f4d2 | 指定待查询的资产UUID。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |
| Count | Integer | 5 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 5 | 返回结果中每页显示数据条数。 |
| TotalCount | Integer | 762 | 返回数据的总条数。 |
| Propertys | | | 返回的端口信息列表。 |
| BindIp | String | 0.0.0.0 | 返回结果中绑定IP地址信息。 |
| Create | String | 1552632559000 | 返回结果中的最新采集时间。 |
| CreateTimestamp | Long | 1552632559000 | 返回结果中的采集时间戳。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| InstanceId | String | 50d213b4-3a35-427a-b8a5-04b0c7e1f4d2" | 返回结果中的资产ID。 |
| InstanceName | String | null | 返回结果中的资产名称。 |
| InternetIp | String | 127.1.1.1 | 返回结果中资产对应公网IP地址。 |
| IntranetIp | String | 0.0.0.0 | 返回结果中资产对应私网IP地址。 |
| Ip | String | null | 返回结果中的IP地址。 |
| Port | String | 22 | 返回的端口信息。 |
| ProcName | String | sshd | 返回端口对应的进程名称。 |
| Proto | String | tcp | 返回的端口对应网络协议。 |
| Uuid | String | 50d213b4-3a35-427a-b8a5-04b0c7e1f4d2" | 返回结果中的资产UUID。 |
| RequestId | String | null | 返回结果的请求ID。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribePropertyPortDetail
&<公共请求参数>
```

**正常返回示例**

`XML` **格式**

```
<DescribePropertyPortDetail>
   <code>200</code>
   <data>
      <Propertys>
         <BindIp>0.0.0.0</BindIp>
         <Port>22</Port>
         <InstanceName></InstanceName>
         <Proto>tcp</Proto>
         <Ip></Ip>
         <Create>1552285726000</Create>
         <ProcName>sshd</ProcName>
```

```
            <Uuid>36bde5fc-a09e-4028-a278-594091c36c74</Uuid>
         </Propertys>
         <Propertys>
            <BindIp>0.0.0.0</BindIp>
            <Port>22</Port>
            <InstanceName></InstanceName>
            <Proto>tcp</Proto>
            <Ip></Ip>
            <Create>1552520423000</Create>
            <ProcName>sshd</ProcName>
            <Uuid>27e7d065-4788-4d0e-8f88-22d7f2a2ccaf</Uuid>
         </Propertys>
         <Propertys>
            <BindIp>0.0.0.0</BindIp>
            <Port>22</Port>
            <InstanceName></InstanceName>
            <Proto>tcp</Proto>
            <Ip></Ip>
            <Create>1552630937000</Create>
            <ProcName>sshd</ProcName>
            <Uuid>inet-f7e35eaa-e112-40c5-af83-58cea026afe7</Uuid>
         </Propertys>
         <Propertys>
            <BindIp>::</BindIp>
            <Port>22</Port>
            <InstanceName></InstanceName>
            <Proto>tcp</Proto>
            <Ip></Ip>
            <Create>1552630937000</Create>
            <ProcName>sshd</ProcName>
            <Uuid>inet-f7e35eaa-e112-40c5-af83-58cea026afe7</Uuid>
         </Propertys>
         <Propertys>
            <BindIp>0.0.0.0</BindIp>
            <Port>22</Port>
            <InstanceName></InstanceName>
            <Proto>tcp</Proto>
            <Ip></Ip>
            <Create>1552632559000</Create>
            <ProcName>sshd</ProcName>
            <Uuid>50d213b4-3a35-427a-b8a5-04b0c7e1f4d2</Uuid>
         </Propertys>
         <PageInfo>
            <Count>5</Count>
            <PageSize>5</PageSize>
            <TotalCount>762</TotalCount>
            <CurrentPage>1</CurrentPage>
         </PageInfo>
      </data>
      <success>true</success>
      <requestId></requestId>
      <message>successful</message>
   </DescribePropertyPortDetail>
```

`JSON` **格式**

```
{
 "message":"successful",
 "data":{
  "PageInfo":{
   "Count":5,
   "TotalCount":762,
   "PageSize":5,
```

```
    "CurrentPage":1
   },
   "Propertys":[
    {
     "Uuid":"36bde5fc-a09e-4028-a278-594091c36c74",
     "Port":"22",
     "Create":1552285726000,
     "Proto":"tcp",
     "BindIp":"0.0.0.0",
     "ProcName":"sshd"
    },
    {
     "Uuid":"27e7d065-4788-4d0e-8f88-22d7f2a2ccaf",
     "Port":"22",
     "Create":1552520423000,
     "Proto":"tcp",
     "BindIp":"0.0.0.0",
     "ProcName":"sshd"
    },
    {
     "Uuid":"inet-f7e35eaa-e112-40c5-af83-58cea026afe7",
     "Port":"22",
     "Create":1552630937000,
     "Proto":"tcp",
     "BindIp":"0.0.0.0",
     "ProcName":"sshd"
    },
    {
     "Uuid":"inet-f7e35eaa-e112-40c5-af83-58cea026afe7",
     "Port":"22",
     "Create":1552630937000,
     "Proto":"tcp",
     "BindIp":"::",
     "ProcName":"sshd"
    },
    {
     "Uuid":"50d213b4-3a35-427a-b8a5-04b0c7e1f4d2",
     "Port":"22",
     "Create":1552632559000,
     "Proto":"tcp",
     "BindIp":"0.0.0.0",
     "ProcName":"sshd"
    }
   ]
  },
  "code":"200",
  "success":true
}
```


**错误码**

**访问错误中心查看更多错误码。**

# 9.3 DescribePropertyProcDetail

调用该接口获取进程列表中一个进程的详细信息。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertyProcDetail | 系统规定参数。取值：DescribePropertyProcDetail。 |
| Cmdline | String | 否 | ./8888 | 指定待查询启动参数。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| Name | String | 否 | 8888 | 指定待查询进程名称。 |
| PageSize | Integer | 否 | 5 | 指定列表每页显示数据条数。 |
| Remark | String | 否 | 0.0.0.0 | 服务器名称或IP。 |
| User | String | 否 | root | 指定待查询运行用户信息。 |
| Uuid | String | 否 | c4678332-ef35-4ad4-8358-681ebbc0ccab | 指定待查询的资产UUID。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |
| Count | Integer | 1 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 5 | 返回结果中每页显示数据条数。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| TotalCount | Integer | 1 | 返回数据的总条数。 |
| Propertys | | | 返回进程信息的列表。 |
| Cmdline | String | ./8888 | 返回结果中进程对应启动参数。 |
| Create | String | 1565686951000 | 返回结果中的最新采集时间。 |
| CreateTimestamp | Long | 1565686951000 | 返回结果中的采集时间戳。 |
| EuidName | String | root | 返回结果中进程对应运行权限。 |
| InstanceId | String | c4678332-ef35-4ad4-8358-681ebbc0ccab | 返回结果中的资产ID。 |
| InstanceName | String | null | 返回结果中的资产名称。 |
| InternetIp | String | 0.0.0.0 | 返回结果中资产对应公网IP地址。 |
| IntranetIp | String | 0.0.0.0 | 返回结果中资产对应私网IP地址。 |
| Md5 | String | N/A | 返回结果中进程对应文件MD5信息。 |
| Name | String | 8888 | 返回结果中进程名称。 |
| Path | String | /root/Oracle/Middleware/user_projects/domains/base_domain/8888 | 返回结果中进程路径。 |
| Pid | String | 12826 | 返回结果中进程PID。 |
| Pname | String | startWebLogic.s | 返回结果中父进程名称。 |
| StartTime | String | 2019-08-07 10:09:05 | 返回结果中进程启动时间。 |
| User | String | root | 返回结果中进程的运行用户。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Uuid | String | c4678332-ef35-4ad4-8358-681ebbc0ccab | 返回结果中的资产UUID。 |
| RequestId | String | null | 返回结果的请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribePropertyProcDetail
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
<DescribePropertyProcDetail>
   <code>200</code>
   <data>
      <PageInfo>
         <Count>1</Count>
         <PageSize>5</PageSize>
         <TotalCount>1</TotalCount>
         <CurrentPage>1</CurrentPage>
      </PageInfo>
      <Propertys>
         <InstanceName></InstanceName>
         <Pname>startWebLogic.s</Pname>
         <Euidname>root</Euidname>
         <Ip></Ip>
         <Pid>12826</Pid>
         <Uuid>c4678332-ef35-4ad4-8358-681ebbc0ccab</Uuid>
         <Path>/root/Oracle/Middleware/user_projects/domains/
base_domain/8888</Path>
         <Cmdline>./8888</Cmdline>
         <Name>8888</Name>
         <Create>1565686951000</Create>
         <StartTime>2019-08-07 10:09:05</StartTime>
         <User>root</User>
         <Md5>N/A</Md5>
      </Propertys>
   </data>
   <success>true</success>
   <requestId></requestId>
   <message>successful</message>
</DescribePropertyProcDetail>
```

#### JSON 格式

```
{
 "message":"successful",
 "data":{
  "PageInfo":{
```

```
    "Count":1,
    "TotalCount":1,
    "PageSize":5,
    "CurrentPage":1
   },
   "Propertys":[
    {
     "Uuid":"c4678332-ef35-4ad4-8358-681ebbc0ccab",
     "User":"root",
     "Md5":"N/A",
     "Euidname":"root",
     "Path":"/root/Oracle/Middleware/user_projects/domains/base_domain/
8888",
     "Name":"8888",
     "Create":1565686951000,
     "Pname":"startWebLogic.s",
     "StartTime":"2019-08-07 10:09:05",
     "Pid":"12826",
     "Cmdline":"./8888"
    }
   ]
  },
  "code":"200",
  "success":true
}
```


### 错误码

访问错误中心查看更多错误码。

## 9.4 DescribePropertyPortItem

调用该接口获取端口信息列表。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertyPortItem | 系统规定参数。取值：DescribePropertyPortItem。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| ForceFlush | Boolean | 否 | true | 是否强制刷新待查询数据。 |
| PageSize | Integer | 否 | 5 | 指定列表每页显示数据条数。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Port | String | 否 | 22 | 指定待查询端口。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |
| Count | Integer | 5 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 5 | 返回结果中每页显示数据条数。 |
| TotalCount | Integer | 163 | 返回数据的总条数。 |
| PropertyItems | | | 返回的端口列表。 |
| Count | Integer | 495 | 返回结果中端口对应的服务器数量。 |
| Port | String | 22 | 返回结果中监听端口号。 |
| Proto | String | tcp | 返回结果中端口对应的网络协议。 |
| RequestId | String | 7E0618A9-D5EF-4220-9471-C42B5E92719F | 返回结果的请求ID。 |

示例

请求示例

```
http(s)://[Endpoint]/?Action=DescribePropertyPortItem
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DescribePropertyPortItem>
   <code>200</code>
   <data>
      <PageInfo>
         <Count>5</Count>
```

```
         <PageSize>5</PageSize>
         <TotalCount>163</TotalCount>
         <CurrentPage>1</CurrentPage>
      </PageInfo>
      <PropertyItems>
         <Port>22</Port>
         <Proto>tcp</Proto>
         <Count>495</Count>
      </PropertyItems>
      <PropertyItems>
         <Port>111</Port>
         <Proto>tcp</Proto>
         <Count>43</Count>
      </PropertyItems>
      <PropertyItems>
         <Port>6000</Port>
         <Proto>tcp</Proto>
         <Count>2</Count>
      </PropertyItems>
      <PropertyItems>
         <Port>53</Port>
         <Proto>tcp</Proto>
         <Count>1</Count>
      </PropertyItems>
      <PropertyItems>
         <Port>80</Port>
         <Proto>tcp</Proto>
         <Count>38</Count>
      </PropertyItems>
   </data>
   <success>true</success>
   <requestId></requestId>
   <message>successful</message>
    </DescribePropertyPortItem>
```

**JSON 格式**

```
{
 "message":"successful",
 "data":{
  "PropertyItems":[
   {
    "Port":"22",
    "Count":495,
    "Proto":"tcp"
   },
   {
    "Port":"111",
    "Count":43,
    "Proto":"tcp"
   },
   {
    "Port":"6000",
    "Count":2,
    "Proto":"tcp"
   },
   {
    "Port":"53",
    "Count":1,
    "Proto":"tcp"
   },
   {
    "Port":"80",
```

```
    "Count":38,
    "Proto":"tcp"
   }
  ],
  "PageInfo":{
   "Count":5,
   "TotalCount":163,
   "PageSize":5,
   "CurrentPage":1
  }
 },
 "code":"200",
 "success":true
}
```


### 错误码

访问错误中心查看更多错误码。

## 9.5 DescribePropertyProcItem

调用该接口获取进程信息列表。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertyProcItem | 系统规定参数。取值：DescribePropertyProcItem。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| ForceFlush | Boolean | 否 | true | 是否强制刷新待查询数据。 |
| Name | String | 否 | test | 指定待查询的进程名称。 |
| PageSize | Integer | 否 | 10 | 指定列表每页显示数据条数。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 5 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 5 | 返回结果中每页显示数据条数。 |
| TotalCount | Integer | 372 | 返回数据的总条数。 |
| PropertyItems | | | 返回的进程列表。 |
| Count | Integer | 8888 | 返回结果中进程名对应的服务器数量。 |
| Name | String | 1 | 返回结果中的进程名。 |
| RequestId | String | null | 返回结果的请求ID。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribePropertyProcItem
&<公共请求参数>
```

**正常返回示例**

`XML` **格式**

```
<DescribePropertyProcItem>
   <code>200</code>
   <data>
      <PropertyItems>
         <Name>(sd-pam)</Name>
         <Count>6</Count>
      </PropertyItems>
      <PropertyItems>
         <Name>.ss</Name>
         <Count>1</Count>
      </PropertyItems>
      <PropertyItems>
         <Name>.sshd</Name>
         <Count>3</Count>
      </PropertyItems>
      <PropertyItems>
         <Name>360Inst_BSFree.exe</Name>
         <Count>1</Count>
      </PropertyItems>
      <PropertyItems>
         <Name>8888</Name>
```

```
            <Count>1</Count>
         </PropertyItems>
         <PageInfo>
            <Count>5</Count>
            <PageSize>5</PageSize>
            <TotalCount>372</TotalCount>
            <CurrentPage>1</CurrentPage>
         </PageInfo>
      </data>
      <success>true</success>
      <requestId></requestId>
      <message>successful</message>
       </DescribePropertyProcItem>
```

JSON 格式

```
{
 "message":"successful",
 "data":{
  "PropertyItems":[
   {
    "Name":"(sd-pam)",
    "Count":6
   },
   {
    "Name":".ss",
    "Count":1
   },
   {
    "Name":".sshd",
    "Count":3
   },
   {
    "Name":"360Inst_BSFree.exe",
    "Count":1
   },
   {
    "Name":"8888",
    "Count":1
   }
  ],
  "PageInfo":{
   "Count":5,
   "TotalCount":372,
   "PageSize":5,
   "CurrentPage":1
  }
 },
 "code":"200",
 "success":true
}
```

## 错误码

访问错误中心查看更多错误码。

# 9.6 DescribePropertySoftwareDetail

调用该接口获取软件列表中一个软件的详细信息。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertySoftwareDetail | 系统规定参数。取值：DescribePropertySoftwareDetail。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| Name | String | 否 | xxxx | 指定待查询的软件名称。 |
| PageSize | Integer | 否 | 5 | 指定列表每页显示数据条数 。 |
| Path | String | 否 | /etc/test | 指定待查询的软件安装路径。 |
| Remark | String | 否 | 0.0.0.0 | 服务器名称或IP。 |
| SoftwareVersion | String | 否 | 11 | 指定待查询的软件版本信息。 |
| Uuid | String | 否 | 50d213b4-3a35-427a-b8a5-04b0c7e1f4d2" | 指定待查询的资产UUID。 |

## 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |
| Count | Integer | 2 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 2 | 返回结果中每页显示数据条数。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| TotalCount | Integer | 23 | 返回数据的总条数。 |
| Propertys | | | 返回的软件详情列表。 |
| Create | Long | 1565539587000 | 返回结果中的最新采集时间。 |
| CreateTimestamp | Long | 1565539587000 | 返回结果中的采集时间戳。 |
| InstallTime | String | 2017-09-07 10:54:49 | 返回的软件安装时间。 |
| InstanceId | String | 4ef1115b-e423-4b4b-b930-a8be682df6ec | 返回结果中的资产ID。 |
| InstanceName | String | null | 返回结果中的资产名称。 |
| InternetIp | String | 0.0.0.0 | 返回结果中资产对应公网IP地址。 |
| IntranetIp | String | 0.0.0.0 | 返回结果中资产对应私网IP地址。 |
| Ip | String | null | 返回IP地址信息。 |
| Name | String | aaa_base | 返回的软件名称。 |
| Path | String | /etc/test | 返回软件资产的安装目录信息。 |
| Uuid | String | 4ef1115b-e423-4b4b-b930-a8be682df6ec | 返回的软件版本信息。 |
| Version | String | 11 | 返回软件资产对应的版本信息。 |
| RequestId | String | null | 返回结果的请求ID。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribePropertySoftwareDetail
```

&<公共请求参数>

**正常返回示例**

**XML 格式**

```
<DescribePropertySoftwareDetail>
   <code>200</code>
   <data>
      <Propertys>
         <Path>/etc/DIR_COLORS</Path>
         <InstanceName></InstanceName>
         <Ip></Ip>
         <Name>aaa_base</Name>
         <Create>1565539587000</Create>
         <InstallTime>2017-09-07 10:54:49</InstallTime>
         <Version>11</Version>
         <Uuid>4ef1115b-e423-4b4b-b930-a8be682df6ec</Uuid>
      </Propertys>
      <Propertys>
         <Path>/etc/bash.bashrc</Path>
         <InstanceName></InstanceName>
         <Ip></Ip>
         <Name>aaa_base</Name>
         <Create>1565544175000</Create>
         <InstallTime>2017-09-07 10:56:38</InstallTime>
         <Version>13.2+git20140911.61c1681</Version>
         <Uuid>d276e6d9-72e5-477a-b4f8-6affcbdad858</Uuid>
      </Propertys>
      <pageInfo>
         <Count>2</Count>
         <PageSize>2</PageSize>
         <TotalCount>23</TotalCount>
         <CurrentPage>1</CurrentPage>
      </pageInfo>
   </data>
   <success>true</success>
   <requestId></requestId>
   <message>successful</message>
</DescribePropertySoftwareDetail>
```

**JSON 格式**

```
{
 "message":"successful",
 "data":{
  "pageInfo":{
   "Count":2,
   "TotalCount":23,
   "PageSize":2,
   "CurrentPage":1
  },
  "Propertys":[
   {
    "Uuid":"4ef1115b-e423-4b4b-b930-a8be682df6ec",
    "Name":"aaa_base",
    "Create":1565539587000,
    "InstallTime":"2017-09-07 10:54:49",
    "Version":"11",
    "Path":"/etc/DIR_COLORS"
   },
```

```
    {
     "Uuid":"d276e6d9-72e5-477a-b4f8-6affcbdad858",
     "Name":"aaa_base",
     "Create":1565544175000,
     "InstallTime":"2017-09-07 10:56:38",
     "Version":"13.2+git20140911.61c1681",
     "Path":"/etc/bash.bashrc"
    }
   ]
  },
  "code":"200",
  "success":true
}
```


**错误码**

访问错误中心查看更多错误码。

## 9.7 DescribePropertySoftwareItem

调用该接口获取软件列表.。

**调试**

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

**请求参数**

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertySoftwareItem | 系统规定参数。取值：DescribePropertySoftwareItem。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| ForceFlush | Boolean | 否 | true | 是否强制刷新待查询数据。 |
| Name | String | 否 | xxx | 指定待查询的软件名称。 |
| PageSize | Integer | 否 | 10 | 指定列表每页显示数据条数 。 |

**返回数据**

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| Count | Integer | 2 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 2 | 返回结果中每页显示数据条数。 |
| TotalCount | Integer | 5037 | 返回数据的总条数。 |
| PropertyItems | | | 返回的软件列表。 |
| Count | Integer | 23 | 返回结果中软件资产对应的服务器数量。 |
| Name | String | aaa_base | 返回的软件资产名称。 |
| RequestId | String | null | 返回结果的请求ID。 |

**示例**

**请求示例**

```
http(s)://[Endpoint]/?Action=DescribePropertySoftwareItem
&<公共请求参数>
```

**正常返回示例**

`XML` **格式**

```
<DescribePropertySoftwareItem>
   <code>200</code>
   <data>
      <PropertyItems>
         <Name>a11y-profile-manager-indicator</Name>
         <Count>1</Count>
      </PropertyItems>
      <PropertyItems>
         <Name>aaa_base</Name>
         <Count>23</Count>
      </PropertyItems>
      <PageInfo>
         <Count>2</Count>
         <PageSize>2</PageSize>
         <TotalCount>5037</TotalCount>
         <CurrentPage>1</CurrentPage>
      </PageInfo>
   </data>
   <success>true</success>
   <requestId></requestId>
   <message>successful</message>
```

```
</DescribePropertySoftwareItem>
```

JSON 格式

```
{
 "message":"successful",
 "data":{
  "PropertyItems":[
   {
    "Name":"a11y-profile-manager-indicator",
    "Count":1
   },
   {
    "Name":"aaa_base",
    "Count":23
   }
  ],
  "PageInfo":{
   "Count":2,
   "TotalCount":5037,
   "PageSize":2,
   "CurrentPage":1
  }
 },
 "code":"200",
 "success":true
}
```

### 错误码

访问错误中心查看更多错误码。

## 9.8 DescribePropertyUserDetail

调用该接口获取账号列表中一个账号的详细信息。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertyUserDetail | 系统规定参数。取值：DescribePropertyUserDetail。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| IsRoot | String | 否 | 0 | 是否ROOT权限。<br>· 0：否<br>· 1：是 |
| PageSize | Integer | 否 | 2 | 指定列表每页显示数据条数 。 |
| Remark | String | 否 | 0.0.0.0 | 服务器名称或IP。 |
| User | String | 否 | test | 指定待查询的账户名称。 |
| Uuid | String | 否 | 50d213b4-3a35-427a-b8a5-04b0c7e1f4d2 | 指定待查询的资产UUID。 |

返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |
| Count | Integer | 2 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 2 | 返回结果中每页显示数据条数。 |
| TotalCount | Integer | 384 | 返回数据的总条数。 |
| Propertys | | | 返回的账号详情列表。 |
| AccountsExpirationDate | String | never | 返回账号的到期时间。 |
| Create | String | 1565644643000 | 返回结果中的最新采集时间。 |
| CreateTimestamp | Long | 1565644643000 | 返回结果中的采集时间戳。 |
| GroupNames | | adm | 用户组。 |

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| InstanceId | String | 02c4a5dc-c2d2-483a-9015-f972b44d2cd9 | 返回结果中的资产ID。 |
| InstanceName | String | null | 返回结果中的资产名称。 |
| InternetIp | String | 0.0.0.0 | 返回结果中资产对应公网IP地址。 |
| IntranetIp | String | 0.0.0.0 | 返回结果中资产对应私网IP地址。 |
| Ip | String | null | 返回结果中的IP地址。 |
| IsRoot | String | 0 | 返回账号是否为ROOT权限。<br>· 0：否<br>· 1：是 |
| LastLoginIp | String | N/A | 上次登录来源。 |
| LastLoginTime | String | xxxx | 上次登录时间。 |
| PasswordExpirationDate | String | never | 返回账号密码的到期时间。 |
| User | String | adm | 账户名称。 |
| Uuid | String | 02c4a5dc-c2d2-483a-9015-f972b44d2cd9 | 返回结果中的资产UUID。 |
| RequestId | String | null | 返回结果的请求ID。 |

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DescribePropertyUserDetail
&<公共请求参数>
```

### 正常返回示例

`XML` 格式

```
<DescribePropertyUserDetail>
```

```
    <code>200</code>
    <data>
       <PageInfo>
          <Count>2</Count>
          <PageSize>2</PageSize>
          <TotalCount>384</TotalCount>
          <CurrentPage>1</CurrentPage>
       </PageInfo>
       <Propertys>
          <LastLoginTime>N/A</LastLoginTime>
          <GroupName>adm</GroupName>
          <IsRoot>0</IsRoot>
          <InstanceName></InstanceName>
          <AccountsExpirationDate>never</AccountsExpirationDate>
          <PasswordExpirationDate>never</PasswordExpirationDate>
          <Ip></Ip>
          <Create>1565644643000</Create>
          <User>adm</User>
          <Uuid>00ea0155-548e-4f32-bb53-3a000110b30d</Uuid>
          <LastLoginIp>N/A</LastLoginIp>
       </Propertys>
       <Propertys>
          <LastLoginTime>N/A</LastLoginTime>
          <GroupName>adm</GroupName>
          <IsRoot>0</IsRoot>
          <InstanceName></InstanceName>
          <AccountsExpirationDate>never</AccountsExpirationDate>
          <PasswordExpirationDate>never</PasswordExpirationDate>
          <Ip></Ip>
          <Create>1565652008000</Create>
          <User>adm</User>
          <Uuid>02c4a5dc-c2d2-483a-9015-f972b44d2cd9</Uuid>
          <LastLoginIp>N/A</LastLoginIp>
       </Propertys>
    </data>
    <success>true</success>
    <requestId></requestId>
    <message>successful</message>
 </DescribePropertyUserDetail>
```

**JSON 格式**

```
{
 "message":"successful",
 "data":{
  "PageInfo":{
   "Count":2,
   "TotalCount":384,
   "PageSize":2,
   "CurrentPage":1
  },
  "Propertys":[
   {
    "Uuid":"00ea0155-548e-4f32-bb53-3a000110b30d",
    "User":"adm",
    "AccountsExpirationDate":"never",
    "IsRoot":"0",
    "LastLoginTime":"N/A",
    "PasswordExpirationDate":"never",
    "GroupName":[
     "adm"
    ],
    "Create":1565644643000,
```

```
      "LastLoginIp":"N/A"
    },
    {
      "Uuid":"02c4a5dc-c2d2-483a-9015-f972b44d2cd9",
      "User":"adm",
      "AccountsExpirationDate":"never",
      "IsRoot":"0",
      "LastLoginTime":"N/A",
      "PasswordExpirationDate":"never",
      "GroupName":[
        "adm"
      ],
      "Create":1565652008000,
      "LastLoginIp":"N/A"
    }
  ]
 },
 "code":"200",
 "success":true
}
```


### 错误码

访问错误中心查看更多错误码。

## 9.9 DescribePropertyUserItem

调用本接口获取账号信息列表。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

| 名称 | 类型 | 是否必选 | 示例值 | 描述 |
| --- | --- | --- | --- | --- |
| Action | String | 是 | DescribePropertyUserItem | 系统规定参数。取值：DescribePropertyUserItem。 |
| CurrentPage | Integer | 否 | 1 | 指定返回结果的当前页码。 |
| ForceFlush | Boolean | 否 | true | 是否强制刷新待查询数据。 |
| PageSize | Integer | 否 | 2 | 指定列表每页显示数据条数 。 |
| User | String | 否 | adm | 指定待查询的账号信息。 |

### 返回数据

| 名称 | 类型 | 示例值 | 描述 |
| --- | --- | --- | --- |
| PageInfo | | | 返回结果的页面显示信息。 |
| Count | Integer | 2 | 返回结果的当前页显示数据条数。 |
| CurrentPage | Integer | 1 | 返回结果中显示的当前页码。 |
| PageSize | Integer | 2 | 返回结果中每页显示数据条数。 |
| TotalCount | Integer | 114 | 返回数据的总条数。 |
| PropertyItems | | | 返回的账号列表。 |
| Count | Integer | 384 | 返回结果中的账号名称。 |
| User | String | adm | 返回结果中账号对应的服务器数量。 |
| RequestId | String | null | 返回结果的请求ID。 |

### 示例

#### 请求示例

```
http(s)://[Endpoint]/?Action=DescribePropertyUserItem
&<公共请求参数>
```

#### 正常返回示例

`XML` 格式

```
<DescribePropertyUserItem>
   <code>200</code>
   <data>
      <PropertyItems>
         <Count>19</Count>
         <User>abrt</User>
      </PropertyItems>
      <PropertyItems>
         <Count>384</Count>
         <User>adm</User>
      </PropertyItems>
      <PageInfo>
         <Count>2</Count>
         <PageSize>2</PageSize>
         <TotalCount>114</TotalCount>
         <CurrentPage>1</CurrentPage>
```

```
      </PageInfo>
   </data>
   <success>true</success>
   <requestId></requestId>
   <message>successful</message>
    </DescribePropertyUserItem>
```

JSON 格式

```
{
 "message":"successful",
 "data":{
  "PropertyItems":[
   {
    "User":"abrt",
    "Count":19
   },
   {
    "User":"adm",
    "Count":384
   }
  ],
  "PageInfo":{
   "Count":2,
   "TotalCount":114,
   "PageSize":2,
   "CurrentPage":1
  }
 },
 "code":"200",
 "success":true
}
```

## 错误码

访问错误中心查看更多错误码。