目录

请求结构

客户调用金山云**操作审计**服务的openAPI接口是通过向指定服务地址发送请求，并按照openAPI文档说明在请求中添加相应的公共参数和接口参数来完成的。

操作审计openAPI的请求结构组成如下：

1. 服务地址

   操作审计的服务接入地址为：actiontrail.api.ksyun.com

2. 通信协议

   支持通过 HTTP 或 HTTPS 两种方式进行请求通信，推荐使用安全性更高的 HTTPS方式发送请求。

3. 请求方法

   操作审计的openAPI同时支持GET和POST请求，推荐使用**GET**请求方式。

4. 请求参数

   金山云openAPI请求包含两类参数：**公共请求参数**和**业务请求参数**。其中，公共请求参数是每个接口都要用到的请求参数，具体可参见公共参数小节；业务请求参数是各个接口所特有的，具体见各个接口的“请求参数”描述。

5. 字符编码

   请求及返回结果都使用UTF-8字符集进行编码。

公共参数

公共参数是每个OpenAPI接口都需要使用的请求参数。支持GET和POST两种HTTP方法。

- GET请求：放在url的query。面。例如： actiontrail.api.ksyun.com?{业务参数}&{公共参数}。
- POST请求：放在http body里面. 例如: {业务参数}&{公共参数}。

名称	类型	是否必须参数	描述
Accesskey	String	是	用户在控制台创建的Accesskey
Service	String	是	服务名称，提供open-api的服务英文简称，操作审计服务固定值actiontrail
Action	String	是	操作接口名，与调用的具体openAPI相关
Version	String	是	接口版本号，与具体的操作接口有关
Timestamp	String	是	时间，UTC格式，例如：2019-08-13T17:18:36Z
SignatureVersion	String	是	签名版本号，固定值：1.0
SignatureMethod	String	是	签名算法，固定值：HMAC-SHA256
Signature	String	是	签名，具体请查看签名机制
Region	String	否	区域，不传默认cn-beijing-6。不同服务支持的region不同，访问操作审计使用的region为cn-beijing-6
SecurityToken	String	否	安全令牌，在使用临时ak(STS 角色扮演获取的ak/sk),需要传该字段，如果使用GET方法，需要对该字段进行urlencode
DryRun	Boolean	否	检查当前调用者是否有权限执行相关操作，而不是真的调用执行相关操作
Format	String	否	指定响应格式，固定值：json

返回结果

返回结果

调用金山云的openAPI服务，调用成功，返回的HTTP状态码（Status）为200；调用失败，返回4xx 或5xx的HTTP状态码（Status）。

金山云的**操作审计**服务的调用返回的数据格式只支持json，请通过设置HTTP Header Accept=application/json来获取json数据格式。

调用成功

json格式示例

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

调用失败

调用接口失败，不会返回结果数据；HTTP请求返回一个4xx或5xx的HTTP状态码，返回的HTTP消息体中包含具体的错误代码(code)及错误信息(message)；与调用成功一样还包含请求ID(RequestId)。

- json格式示例

```
{
    "RequestId": "68093a99-2f63-4f39-8f70-3047ab8ecb5b",
    "Error": {
        "Type": "Sender",
        "Code": "InvalidParameterValue",
        "Message": "An invalid or out-of-range value was supplied for the input parameter PathPrefix."
    }
}
```

- 当请求出错时建议可以先根据该错误码在公共错误码(所有业务都可能出现的错误码为公共错误码)和当前接口对应的错误码列表里面查找对应原因和解决方案。
- 如果您找不到错误原因时，可以联系金山云客服，并提供RequestId，以便我们尽快帮您解决问题。

签名机制

签名过程

1.构造规范化请求字符串

第一步：请求参数排序

- 请求参数包括公共参数和业务参数，不包含公共参数Signature。
- 排序规则以参数名按照字典排序。

第二步：请求参数编码 使用UTF-8字符集按照RFC3986规则编码请求参数和参数取值，编码规则如下：

- 对于字符A ~ Z、a ~ z、0 ~ 9以及字符-、_、.和~不编码。
- 对于其他字符编码成%XY的格式，其中XY是字符对应ASCII码的16进制(大写)。

  不同语言实现的URLEncode方式不同，为了得到RFC3986规则的编码，分别说明如下：

  - 如果您使用的是Java中的java.net.URLEncoder，可以先用URLEncoder.encode方法编码，随后将编码后的字符中加号（+）替换为%20、星号（*）替换为%2A、%7E替换为波浪号（~）即可；
  - 如果您使用的是golang中的net/url，可以用url.Values.Encode方法编码，随后将编码后的字符中加号（+）替换为%20即可；
  - 如果您使用的是php中的rawurlencode方法，不需要替换；
  - 如果您使用的是python3中的urllib.request，可以用urllib.request.quote方法，需要指定字符串中的波浪号（~）不编码，例如urllib.request.quote(str, '~')，str为待编码字符串；

第三步：请求参数拼接成CanonicalizedQueryString 每对URLEncode后的参数名称和参数值，用=进行连接。每对之间使用&进行连接。得到规范化请求字符串CanonicalizedQueryString。

2.计算签名。

sign = hash_hmac(‘sha256’, CanonicalizedQueryString, sk)

sign值为签名算法返回的16进制格式小写字符串

签名样例：

5346bfebeb3f2162e2459e09a52b640584e5f1aa5012b15c6b85388680d4663e 计算签名时使用的sk为Accesskey对应的秘钥，使用的哈希算法是：HMAC-SHA256。

3.将签名值作为Signature参数值添加到请求参数中。

签名过程示例

该样例以iam服务的CreateUser为例，点击查看该API文档。

1、构造规范化请求字符串：CanonicalizedQueryString

```
//请求参数数组:
array (
    'Accesskey'=> 'AKLTXQVFOpOmS6aahIrD5r0B3Q', //公共参数
    'Service'=> 'iam', //公共参数
    'Action'=> 'CreateUser', //公共参数
    'Version'=> '2015-11-01', //公共参数
    'Timestamp'=> '2021-08-12T02:47:36Z', //公共参数
    'SignatureVersion'=> '1.0', //公共参数
    'SignatureMethod'=> 'HMAC-SHA256', //公共参数
    'UserName' => 'Ttest', //业务参数
    'RealName' => '周四测试', //业务参数
    'Email' => 'zsce@kkingsoft.com', //业务参数
    'Remark' => '~ce shi*%#|+', //业务参数
)
```

第一步：请求参数排序

```
//排序结果如下:
array (
    'Accesskey' => 'AKLTXQVFOpOmS6aahIrD5r0B3Q',
    'Action' => 'CreateUser',
    'Email' => 'zsce@kkingsoft.com',
    'RealName' => '周四测试',
    'Remark' => '~ce shi*%#|+',
    'Service' => 'iam',
    'SignatureMethod' => 'HMAC-SHA256',
    'SignatureVersion' => '1.0',
    'Timestamp' => '2021-08-12T02:47:36Z',
    'UserName' => 'Ttest',
    'Version' => '2015-11-01',
)
```

第二步：请求参数编码

```
//编码结果如下:
array (
    'Accesskey' => 'AKLTXQVFOpOmS6aahIrD5r0B3Q',
    'Action' => 'CreateUser',
    'Email' => 'zsce%40kkingsoft.com',
    'RealName' => '%E5%91%A8%E5%9B%9B%E6%B5%8B%E8%AF%95',
    'Remark' => '~ce%20shi%2A%25%23%7C%2B',
    'Service' => 'iam',
    'SignatureMethod' => 'HMAC-SHA256',
    'SignatureVersion' => '1.0',
    'Timestamp' => '2021-08-12T02%3A47%3A36Z',
    'UserName' => 'Ttest',
    'Version' => '2015-11-01',
)
```

第三步：请求参数拼接成CanonicalizedQueryString

```
//拼接结果如下
Accesskey=AKLTXQVFOpOmS6aahIrD5r0B3Q&Action=CreateUser&Email=zsce%40kkingsoft.com&RealName=%E5%91%A8%E5%9B%9B%E6%B5%8B%E8%AF%95&Remark=~ce%20shi%2A%25%23%7C%2B&Service=iam&SignatureMethod=HMAC-SHA256&SignatureVersion=1.0&Timestamp=2021-08-12T02%3A47%3A36Z&UserName=Ttest&Version=2015-11-01
```

2、计算签名

```
sk = 'OMovU5PTLh6y9E9Ioe3K411jt99VqyQSBXgAcDYlo49R3lvUIzb6e/efZCFDmtFlzw=='
CanonicalizedQueryString = "Accesskey=AKLTXQVFOpOmS6aahIrD5r0B3Q&Action=CreateUser&Email=zsce%40kkingsoft.com&RealName=%E5%91%A8%E5%9B%9B%E6%B5%8B%E8%AF%95&Remark=~ce%20shi%2A%25%23%7C%2B&Service=iam&SignatureMethod=HMAC-SHA256&SignatureVersion=1.0&Timestamp=2021-08-12T02%3A47%3A36Z&UserName=Ttest&Version=2015-11-01"
//sign = hash_hmac('sha256', CanonicalizedQueryString, sk)
sign: fc9088ab845949dac4040be9b7ce7859068b5c21d4c400fec8ee0cefb777f659
```

3、将签名值作为Signature参数值添加到请求参数中

```
//请求示例:
```

```
curl -s -L -X POST 'iam.api.ksyun.com' \
-H 'Accept: application/json' \
-H 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'Accesskey=AKLTXQVFOpOmS6aahIrD5rOB3Q' \
--data-urlencode 'Service=iam' \
--data-urlencode 'Action=CreateUser' \
--data-urlencode 'Version=2015-11-01' \
--data-urlencode 'Timestamp=2021-08-12T02:47:36Z' \
--data-urlencode 'SignatureVersion=1.0' \
--data-urlencode 'SignatureMethod=HMAC-SHA256' \
--data-urlencode 'UserName=Ttest' \
--data-urlencode 'RealName=周四测试' \
--data-urlencode 'Email=zsce@kkingsoft.com' \
--data-urlencode 'Remark=~ce shi*%#|+' \
--data-urlencode 'Signature=fc9088ab845949dac4040be9b7ce7859068b5c21d4c400fec8ee0cefb777f659'
```

签名demo

提供php、java、go、python四种语言的签名demo，请点击此处查看。

查询操作日志列表

概述

查询本账号内操作日志的列表信息。金山云账号或被授权的子用户可以调用该接口。 注：普通页面展示用Page与PageSize参数，此方法有限制，仅支持共查询10000条数据；查询所有数据用SearchAfter与PageSize参数。

openAPI接口

请求参数

名称	类型	必须	长度限制(字符)	参数格式	描述
Action	String	是			操作接口名，与调用的具体openAPI相关,传 ListOperateLogs
Version	String	是			接口版本号，即2019-04-01
EventName	String	否			事件名称
EventRw	String	否			事件类型：read(读操作)，write(写操作)；不传则查询所有类型操作
UserName	String	否			用户名
AccessKey	String	否			AccessKeyID
EventBeginDate	String	否			事件开始日期，格式为YYYY-MM-DD
EventEndDate	String	否			事件结束日期，格式为YYYY-MM-DD
ResourceType	String	否			资源类型
ResourceName	String	否			资源名称
Page	Integer	否			页码，默认1
PageSize	Integer	否			每页查询个数，默认10
SearchAfter	String	否			上次请求返回的最后一条数据的游标

返回元素（调用成功）

名称	类型	描述
Total	Integer	操作日志总数量
Events	List	操作日志**列表**
SearchAfter	String	记录查询结果列表最后一条数据的游标

错误码（调用不成功）

错误代码	描述	HTTP 状态码	语义

InvalidParameterValue	An invalid or out-of-range value was supplied for the input parameter.	400	参数格式/取值范围不对，具体参数名在错误消息中说明

示例（调用成功）

请求示例

```
https://actiontrail.api.ksyun.com/?Action=ListOperateLogs
&Version=2019-04-01
&AUTHPARAMS
```

返回示例

- json示例


```
{
  "Total": 1,
  "Events": [
    {
      "ErrorMessage": "",
      "CreateTime": "2019-05-06 17:04:57",
      "ServiceName": "iam",
      "EventSource": "iam.inner.api.ksyun.com",
      "ApiVersion": "2015-11-01",
      "RequestParameters": {
        "Version": "2015-11-01",
        "AccountId": "200000000",
        "Action": "SetMemberViewAllProject",
        "ViewAllProject": "1",
        "UserName": "jiming_yu@kingdee.com"
      },
      "SourceIpAddress": "10.69.38.142",
      "EventVersion": "1",
      "EventType": "ApiCall",
      "EventId": "5a82ed6e-510e-4662-9839-cde76aa73c59",
      "EventRw": "write",
      "EventName": "SetMemberViewAllProject",
      "UserIdentity": {
        "AccountId": "200000000",
        "UserType": "Account",
        "UserName": "root",
        "AccessKey": ""
      },
      "ErrorCode": "",
      "Region": "cn-beijing-6",
      "RequestId": "ba7eb142-e896-42d0-95d4-7c711c455f4a",
      "EventTime": "2019-05-06 17:04:57",
      "UserAgent": "GuzzleHttp/6.2.1 curl/7.19.7 PHP/7.0.14"
    }
  ],
  "SearchAfter": "[1565086382000,\"2bf6ac68-9bda-4fc0-8554-be11b89b3fe2\"]",
  "RequestId": "4df19df5-7d5b-4cd1-8721-c9f2cbae1261"
}
```


返回操作日志一览表