

目录

目录	1
请求结构	2
公共参数	2
传入方式	2
参数说明	2
返回结果	2
调用成功	3
调用失败	3
签名机制	3
签名算法	3
签名计算完整样例	3
各语言版本的签名	4
PHP (版本5.5+)	4
Python (版本2.7)	4
携号转网号码查询	4
描述	4
HTTP METHOD	4
HTTP BODY编码格式	5
请求参数	5
返回结果	5
示例 (调用成功)	5
空号号码查询	5
描述	5
HTTP METHOD	5
HTTP BODY编码格式	6
请求参数	6
返回结果	6
示例 (调用成功)	6
黑名单号码查询	6
描述	6
HTTP METHOD	6
HTTP BODY编码格式	6
请求参数	6
返回结果	7
示例 (调用成功)	7
API错误码	7
功能说明	7
短信发送错误码	7

请求结构

客户调用金山云短信服务的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。

短信openAPI的请求结构组成如下：

1. 服务地址

短信的服务接入地址为：`smsopen.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

短信的openAPI仅支持POST请求方式。

注意

- 使用 POST 方式，参数均从 请求Body中取得，需要使用x-www-form-urlencoded方式进行编码。

4. 请求参数

金山云openAPI请求包含两类参数：**公共请求参数**和**接口请求参数**。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；接口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

传入方式

- GET请求, 放在url的query里面. e.g: `xx.api.ksyun.com?{业务参数}&{公共参数}`
- POST请求, 放在http body里面. e.g: `{业务参数}&{公共参数}`

参数说明

公共请求参数(必须)是每个OpenAPI接口都需要使用的请求参数。

名称	类型	是否必须参数	描述
Accesskey	String	是	ak 用户在控制台创建的accesskey
Service	String	是	服务名称，固定值smsopen
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，固定值：2019-05-01
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 签名算法
Region	String	否	区域，例如：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

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)。

金山云的短信服务的调用返回的数据格式仅支持json。

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)，在调用方找不到错误原因时，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。

- 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."
  }
}
```

签名机制

签名算法

1、根据请求参数(公共参数和业务参数，不包含公共参数Signature)构造规范化请求字符串：CanonicalizedQueryString

第一步：请求参数排序。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字符集对每个请求参数的名称和参数取值进行URLEncode，一般在URLEncode后需对三种字符替换：加号(+)替换成%20、星号(*)替换成%2A、%7E替换成波浪号(~)

第三步：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：

```
88b203541ce8c757d7d554af2a25de036d3d9a636d91fb44d01bf82dc67a6941
```

计算签名时使用的sk为Accesskey对应的密钥，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。

签名计算完整样例

1、构造规范化请求字符串样例

请求参数数组：

```
array (
  'Mobile' => '1xxxxxxxxx',
  'TplId' => '1xxx',
  'TplParams' => '{"key": "v~al"}',
  'SignName' => '签名',
  'Action' => 'SendSms',
  'Version' => '2019-05-01',
  'SignatureVersion' => '1.0',
  'SignatureMethod' => 'HMAC-SHA256',
  'Timestamp' => '2019-08-13T17:18:36Z',
  'Service' => 'ksms',
  'Accesskey' => 'xxx',
)
```

第一步，请求参数排序：排序结果如下。

```
array (
  'Accesskey' => 'xxx',
  'Action' => 'SendSms',
  'Mobile' => '1xxxxxxxxx',
  'Service' => 'ksms',
  'SignName' => '签名',
  'SignatureMethod' => 'HMAC-SHA256',
  'SignatureVersion' => '1.0',
  'Timestamp' => '2019-08-13T17:18:36Z',
  'TplId' => '1xxx',
  'TplParams' => '{"key": "v~a"}',
  'Version' => '2019-05-01',
)
```

第二步请求参数和参数值URLEncode，结果如下：

```
array (
  'Accesskey' => 'xxx',
  'Action' => 'SendSms',
  'Mobile' => '1xxxxxxxxx',
  'Service' => 'ksms',
  'SignName' => '%E7%AD%BE%E5%90%8D',
  'SignatureMethod' => 'HMAC-SHA256',
  'SignatureVersion' => '1.0',
  'Timestamp' => '2019-08-13T17%3A18%3A36Z',
  'TplId' => '1xxx',
  'TplParams' => '%7B%22key%22%3A%22v~a%22%7D',
  'Version' => '2019-05-01',
)
```

第三步字符串拼接如下：

```
Accesskey=xxx&Action=SendSms&Mobile=1xxxx&Service=ksms&SignName=%E7%AD%BE%E5%90%8D&SignatureMethod=HMAC-SHA256&SignatureVersion=1.0&Timestamp=2019-08-13T17%3A18%3A36Z&TplId=1xxx&TplParams=%7B%22key%22%3A%22v~a%22%7D&Version=2019-05-01
```

2、根据待签串计算签名

```
sk = 123456
CanonicalizedQueryString = "Accesskey=xxx&Action=SendSms&Mobile=1xxxx&Service=ksms&SignName=%E7%AD%BE%E5%90%8D&SignatureMethod=HMAC-SHA256&SignatureVersion=1.0&Timestamp=2019-08-13T17%3A18%3A36Z&TplId=1xxx&TplParams=%7B%22key%22%3A%22v~a%22%7D&Version=2019-05-01"
//sign = hash_hmac('sha256', CanonicalizedQueryString, sk)
sign: e2925c6745e11b06107920591b318c883b3b825bbc47fded40489bfbff6e660e
```

各语言版本的签名

PHP（版本5.5+）

```
function sign($params, $secret_key)
{
    ksort($params, SORT_STRING);
    $str_encode = '';
    foreach($params as $k=>$v) {
        $str_encode .= rawurlencode($k).'=' . rawurlencode($v).'&';
    }
    $str_encode = substr($str_encode, 0, -1);
    return hash_hmac("sha256", $str_encode, $secret_key);
}
//rawurlencode 实现的urlencode，符合上述签名URLEncode的要求，所以无需做替换
```

Python（版本2.7）

```
#!/usr/bin/env python
#-*- coding:utf-8 -*-
import sys, os, base64, datetime, hashlib, hmac
import urllib
def sign(params, secret_key):
    str_encode = ''
    param_keys = sorted(params.keys())
    for key in param_keys:
        str_encode += urllib.quote(key, '~') + '=' + urllib.quote(str(params[key]), '~') + '&'
    print(str_encode[:-1])
    return hmac.new(secret_key, str_encode[:-1], hashlib.sha256).hexdigest()
```

携号转网号码查询

描述

查询提交手机号码中是否存在携号转网号码

HTTP METHOD

POST

HTTP BODY编码格式

application/x-www-form-urlencoded

请求参数

公共参数列表 详情	名称	类型	必须	长度限制	参数备注
Action	String	是	确定值	a-zA-Z+	操作接口名，系统规定参数，取值：Switch Number
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01
Mobile	String	是	国内手机号：11位数字，例如 1xxxxxxxxx		只支持国内手机号，多个手机号码使用英文逗号(,)连接，单次提交最多支持200个手机号码

返回结果

通过指定 Accept:application/json，返回的数据格式为json

名称	类型	描述
data	array	携号转网列表
RequestId	String	请求唯一标识

示例（调用成功）

请求示例

```
curl -X POST \
'http://ksmsapi.api.ksyun.com/' \
-H 'accept: application/json' \
-H 'content-type: application/x-www-form-urlencoded' \
-H 'host: ksmsapi.api.ksyun.com' \
-d 'Mobile=159xxxx,139xxxx&Action=PortabilityNumber&Version=2019-05-01&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=ksmsapi&Accesskey=xxx'
```

响应成功返回示例

```
{
  "data": [
    {
      "original_operator": "电信",
      "current_operator": "联通",
      "mobile": "1xxxxxxxxx"
    },
    {
      "original_operator": "联通",
      "current_operator": "电信",
      "mobile": "1xxxxxxxxx"
    }
  ],
  "RequestId": "21ca0d6f-3120-4eb5-b707-6e53e3e85a92"
}
```

响应失败返回示例

```
{
  "RequestId": "B6D67E30-B7E2-4225-A53E-C3CC8D2CDB64",
  "Error": {
    "Type": "server",
    "Message": "The Account do not call service",
    "Code": "AccountServiceUnavailable"
  }
}
```

空号号码查询

描述

查询提交手机号码中是否存在空号号码

HTTP METHOD

POST

HTTP BODY编码格式

application/x-www-form-urlencoded

请求参数

公共参数列表 详情	名称	类型	必须	长度限制	参数备注
Action	String	是	确定值	a-zA-Z+	操作接口名，系统规定参数，取值：EmptyMobile
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01
Mobile	String	是	国内手机号：11位数字，例如 1xxxxxxxxx		只支持国内手机号，多个手机号码使用英文逗号(,)连接，单次提交最多支持200个手机号码

返回结果

通过指定 Accept:application/json，返回的数据格式为json

名称	类型	描述
data	array	空号列表
RequestId	String	请求唯一标识

示例（调用成功）

请求示例

```
curl -X POST \
'http://ksmsapi.api.ksyun.com/' \
-H 'accept: application/json' \
-H 'content-type: application/x-www-form-urlencoded' \
-H 'host: ksmsapi.api.ksyun.com' \
-d 'Mobile=1xxxxxxxxx,1xxxxxxxxx&Action=EmptyMobile&Version=2019-05-01&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=ksmsapi&Accesskey=xxx'
```

响应成功返回示例

```
{
  "data": [
    "1xxxxxxxxx"
  ],
  "RequestId": "4931af9f-76ee-40a0-814b-606615b141a3"
}
```

响应失败返回示例

```
{
  "RequestId": "B6D67E30-B7E2-4225-A53E-C3CC8D2CDB64",
  "Error": {
    "Type": "server",
    "Message": "The Account do not call service",
    "Code": "AccountServiceUnavailable"
  }
}
```

黑名单号码查询

描述

查询提交手机号码中是否存在黑名单号码

HTTP METHOD

POST

HTTP BODY编码格式

application/x-www-form-urlencoded

请求参数

公共参数列表 详情	名称	类型	必须	长度限制	参数备注
Action	String	是	确定值	a-zA-Z+	操作接口名，系统规定参数，取值：BlackList
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01
Mobile	String	是	国内手机号：11位数字，例如 1xxxxxxxxx		只支持国内手机号，多个手机号码使用英文逗号(,)连接，单次提交最多支持200个手机号码

返回结果

通过指定 Accept:application/json，返回的数据格式为json	名称	类型	描述
data	array		黑名单列表
RequestId	String		请求唯一标识

示例（调用成功）

请求示例

```
curl -X POST \
'http://ksmsapi.api.ksyun.com/' \
-H 'accept: application/json' \
-H 'content-type: application/x-www-form-urlencoded' \
-H 'host: ksmsapi.api.ksyun.com' \
-d 'Mobile=1xxxxxxxxx,1xxxxxxxxx&Action=BlackList&Version=2019-05-01&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=ksmsapi&Accesskey=xxx'
```

响应成功返回示例

```
{
  "data": [
    "1xxxxxxxxx",
    "1xxxxxxxxx"
  ],
  "RequestId": "611f081b-fd69-43bf-8241-2fded8e8c6d0"
}
```

响应失败返回示例

```
{
  "RequestId": "B6D67E30-B7E2-4225-A53E-C3CC8D2CDB64",
  "Error": {
    "Type": "server",
    "Message": "The Account do not call service",
    "Code": "AccountServiceUnavailable"
  }
}
```

API 错误码

功能说明

若 API 调用失败，用户可以根据错误码列表查询相关的错误信息。

短信发送错误码

错误码	说明	HTTP状态码	处理措施
Unknow	未知错误	500	请联系客服人员沟通解决
ServiceUnavailable	服务临时不可用	500	请联系客服人员沟通解决
InvalidAccesskey	accesskey参数错误	400	请联系检查与控制台创建的accesskey是否一致
FlowLimitExceeded	调用次数过多	400	请联系客服人员沟通解决
NotWhiteList	请求账号不在白名单中	401	请检查账号是否有短信服务权限，如还有问题请联系客服人员沟通解决
SendSmsFailed	发送短信失败	500	请联系客服人员沟通解决

InvalidAccountId	账号错误	400	请遵守API接口说明规范
InvalidMobile	手机号格式错误	400	请遵守API接口说明规范
InvalidSignName	签名不存在	400	请遵守API接口说明规范
InvalidTplId	模板ID不存在	400	请遵守API接口说明规范
InvalidTplParams	模板参数不存在或者错误	400	请遵守API接口说明规范
InvalidExtId	外部流水扩展格式错误	400	请遵守API接口说明规范
InvalidContent	短信内容不存在	400	请遵守API接口说明规范
InvalidContentLength	超过短信内容最大长度	400	不能超过500字符
MobileCountLimit	批量发送短信手机号超过规定的数量	400	不能超过500个
NotSupportInternational	不支持国际短信	400	请联系客服人员沟通解决
EmptySmsAccount	短信账号不存在	400	请联系客服人员沟通解决
InvalidSmsSendTime	营销短信发送时间限制	400	为避免骚扰用户，营销短信只允许在8点到22点发送
TplContainUrl	短信模板不能有url	400	请遵守API接口说明规范
InvalidTplLen	短信模板长度不能超过短信内容最大长度	400	不能超过500字符
MobileFrequencyLimit	往同个手机号下发短信的条数超过控制台上设定的频度控制上限	400	可自行到控制台调整短信频率限制策略
InvalidNationCode	国际手机号格式错误	400	请遵守API接口说明规范
NotAllowNation	该账号不允许在此国家发送	400	请联系管理员沟通解决
SignatureNotMatch	签名不匹配	403	请查看接口的签名实现算法
ServiceUnavailable	请求被拒绝，因其引用的InnerAPI无内部服务	404	请联系客服人员沟通解决
InnerServiceUnavailable	openAPI或服务暂不可用	500	请联系客服人员沟通解决
NoSuchEntity	请求被拒绝，因其引用的InnerAPI不存在	404	请联系客服人员沟通解决
FlowLimitExceeded	请求被拒绝，因该openAPI接口访问速度已达到流控上限	409	请控制调用API的速率
MissingParameter	输入参数 %s 的值不能为空	400	请仔细阅读接口文档
InvalidTimestamp	请保持请求时间戳在15分钟以内	400	请注意接口传参值范围
InvalidTimestampFormat	请求时间戳格式不合法(%s)，格式：YYYY-MM-DDThh:mm:ssZ	400	请仔细阅读接口文档
ServiceTimeout	内部服务由于超时暂不可用	500	请联系客服人员沟通解决