

## 目录

目录	1
请求结构	4
公共参数	4
传入方式	4
参数说明	4
返回结果	4
调用成功	5
调用失败	5
签名机制	5
签名算法	5
签名计算完整样例	5
各语言版本的签名	6
PHP (版本5.5+)	6
Python (版本2.7)	6
获取模板列表 (ListTemplates)	6
描述	6
HTTP METHOD	6
请求参数	7
返回结果	7
模板详情	7
示例 (调用成功)	7
请求示例	7
返回示例	7
获取模板详情 (GetTemplateById)	7
描述	8
HTTP METHOD	8
请求参数	8
返回结果	8
模板详情	8
示例 (调用成功)	8
请求示例	8
返回示例	8
创建模板 (CreateTemplate)	9
描述	9
HTTP METHOD	9
请求参数	9
返回结果	9
示例 (调用成功)	9
请求示例	9
返回示例	9
定时发送短信	9
定时发送短信 (SendTimingSms)	9
描述	9
HTTP METHOD	9
请求参数	10
返回结果	10
示例 (调用成功)	10
请求示例	10
返回示例	10
添加签名	10

添加签名 (AddSmsSign)	10
描述	10
HTTP METHOD	10
请求参数	10
返回结果	11
示例 (调用成功)	11
请求示例	11
返回示例	11
修改签名	11
修改签名 (ModifySmsSign)	11
描述	11
HTTP METHOD	11
请求参数	11
返回结果	12
示例 (调用成功)	12
请求示例	12
返回示例	12
查询签名	12
查询签名 (QuerySmsSign)	12
描述	12
HTTP METHOD	12
请求参数	12
返回结果	12
示例 (调用成功)	13
请求示例	13
返回示例	13
删除签名	13
删除签名 (DeleteSmsSign)	13
描述	13
HTTP METHOD	13
请求参数	13
返回结果	13
示例 (调用成功)	13
请求示例	13
返回示例	13
签名列表	14
描述	14
HTTP METHOD	14
请求参数	14
返回结果	14
签名详情	14
示例 (调用成功)	14
请求示例	14
返回示例	14
数据总览	15
国内短信数据总览 (GetInternalSmsOverview)	15
描述	15
HTTP METHOD	15
请求参数	15
返回结果	15
Stats定义	15
示例 (调用成功)	15

---

请求示例	15
响应成功返回示例	16
响应失败返回示例	16

## 请求结构

客户调用金山云短信控制台的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。

短信控制台openAPI的请求结构组成如下：

### 1. 服务地址

短信控制台的服务接入地址为：`sms.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

短信控制台的openAPI仅支持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**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sms
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 <a href="#">签名算法</a>
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' => '1xxx',
  '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' => '1xxxx',
'Service' => 'ksms',
'SignName' => '签名',
'SignatureMethod' => 'HMAC-SHA256',
'SignatureVersion' => '1.0',
'Timestamp' => '2019-08-13T17:18:36Z',
'TplId' => '1xxx',
'TplParams' => '{"key": "v~al"}',
'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' => '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~al%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~al%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~al%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）

```

#-*- 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()

```

## 获取模板列表（ListTemplates）

### 描述

获取模板列表

### HTTP METHOD

GET

## 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：ListTemplates	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
Page	Int	否			当前页	默认1
PageSize	Int	否			每页条数	默认10

## 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	备注
Templates	array	模板列表	
Total	String	模板总数	
RequestId	String	请求唯一标识	

## 模板详情

名称	类型	描述	备注
Id	Int	模板ID	
Status	Int	模板状态	1待审核；2审核通过；3审核未通过
Name	String	模板名称	
Type	String	模板类型	1验证码；2短信通知；3推广短信
CreateTime	String	模板申请时间	
AuditedTime	String	模板审核时间	
Content	String	模板内容	
Description	String	申请描述	
StatusName	String	模板状态名称	
TypeName	String	模板类型名称	

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=ListTemplates&Version=2019-05-01&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "Templates": [
    {
      "Id": 1,
      "Status": 1,
      "Name": "***",
      "Type": 1,
      "CreateTime": "2020-06-05 16:22:34",
      "AuditedTime": "",
      "Content": "您好，您的验证码是{code}",
      "Description": "",
      "StatusName": "待审核",
      "TypeName": "验证码"
    }
  ],
  "Total": 1,
  "RequestId": "68740cac-1b43-4042-ba65-4240fe96963c"
}
```

## 获取模板详情（GetTemplateById）

## 描述

获取模板详情

## HTTP METHOD

GET

## 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：GetTemplateById	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
TemplateId	Int	是			模板ID	

## 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	备注
Template	array	模板列表	
RequestId	String	请求唯一标识	

## 模板详情

名称	类型	描述	备注
Id	Int	模板ID	
UserId	Int	用户ID	
Status	Int	模板状态	1待审核；2审核通过；3审核未通过
Name	String	模板名称	
Type	Int	模板类型	1验证码；2短信通知；3推广短信
CreateTime	String	模板申请时间	
AuditedTime	String	模板审核时间	
Content	String	模板内容	
Description	String	申请描述	
Variable	String	模板变量	

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=GetTemplateById&Version=2019-05-01&TemplateId=1&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "Template": {
    "Id": 1,
    "UserId": 1,
    "Status": 2,
    "Name": "验证码",
    "Type": 1,
    "CreateTime": "2020-02-03 14:23:10",
    "AuditedTime": "2020-02-03 14:23:24",
    "Content": "您的验证码是{code}",
    "Description": "",
    "Variable": "code"
  }
},
```



```
  "RequestId": "7f46b7c0-948d-49b8-a560-bc9595ce2acc"
}
```

## 创建模板 (CreateTemplate)

### 描述

创建模板

### HTTP METHOD

GET

### 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：CreateTemplate	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
Type	Int	是			模板类型	1验证码，2短信通知，3推广短信
Name	String	是			模板名称	
Content	String	是			模板内容	
Description	String	否			申请描述	

### 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	备注
TemplateId	Int	模板ID	
RequestId	String	请求唯一标识	

### 示例（调用成功）

#### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=CreateTemplate&Version=2019-05-01&Type=1&Name=验证码模板&Content=您好，您的验证码是{code}&Description=申请描述&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

#### 返回示例

```
{
  "TemplateId": 1,
  "RequestId": "9288f11c-7cd8-4a01-bdf3-da70c537dc52"
}
```

## 定时发送短信

### 定时发送短信 (SendTimingSms)

#### 描述

定时发送短信（注：需开通免模版权限）

#### HTTP METHOD

GET

## 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：SendTimingSms	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
SmsType	Int	是			短信类型	类型值：1验证码，2通知，3营销类短信
SignName	String	是			签名名称	
Mobile	String	是			手机号	如果多个手机号放到list中，例：["1xxxxxxxxxx", "1xxxxxxxxxx"]
Content	String	是			短信内容	如果个性化内容，内容放到list中，手机号与内容顺序一一对应，例：["短信内容1", "短信内容2"]
SendTime	String	是			发送时间，时间范围在24小时以内	例：2020-05-01 12:00:00

## 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	备注
RequestId	String	请求唯一标识	
AssistantId	Int	发送任务ID	
Sid	String	短信ID	

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=SendTimingSms&Version=2019-05-01&SmsType=1&SignName=金山云&Content=您好，您的验证码是{code}&Mobile=135xxx&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "Sid": "xxxx",
  "AssistantId": xxx,
  "RequestId": "ecaeb6ef-fc74-4cb1-8447-d81f33439def"
}
```

# 添加签名

## 添加签名（AddSmsSign）

### 描述

添加签名（此接口需要联系客服单独开通）

### HTTP METHOD

GET

### 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：AddSmsSign	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
SignName	String	是			短信签名名称	
SignType	Int	是			短信签名类型	1. 公司 2. APP 3. 网站 4. 公众号/小程序 5. 商标 6. 政府/单位
Description	String	否			申请描述	

## 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	备注
SignId	Int	短信签名ID	
RequestId	String	请求唯一标识	

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=AddSmsSign&Version=2019-05-01&SignType=1&SignName=短信签名名称&Description=申请描述&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "SignId": 10000,
  "RequestId": "9288f11c-7cd8-4a01-bdf3-da70c537dc52"
}
```

# 修改签名

## 修改签名（ModifySmsSign）

### 描述

修改签名（此接口需要联系客服单独开通）

### HTTP METHOD

GET

### 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：ModifySmsSign	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
SignId	String	是			短信签名ID	
SignName	String	是			短信签名名称	
SignType	Int	是			短信签名类型	1. 公司 2. APP 3. 网站 4. 公众号/小程序 5. 商标 6. 政府/单位

Description	String	否	申请描述
-------------	--------	---	------

## 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	备注
SignId	Int	短信签名ID	
SignName	String	短信签名名称	
SignType	Int	短信签名类型	
Description	String	申请描述	
RequestId	String	请求唯一标识	

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=ModifySmsSign&Version=2019-05-01&SignId=10000&SignType=2&SignName=短信签名名称&Description=申请描述&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "SignId": "10000",
  "SignName": "短信签名名称",
  "SignType": "2",
  "Description": "申请描述",
  "RequestId": "a63b6662-7f0d-4dcf-82c4-169de13b85ba"
}
```

# 查询签名

## 查询签名（QuerySmsSign）

### 描述

查询签名

### HTTP METHOD

GET

### 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：QuerySmsSign	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
SignId	Int	是			短信签名ID	

## 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	备注
SignName	String	短信签名名称	
Status	Int	短信签名审核状态，1-审核中 2-通过 3-未通过	
SignType	Int	短信签名类型	
Reason	String	审核未通过的原因	

RequestId String 请求唯一标识

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=QuerySmsSign&Version=2019-05-01&SignId=10000&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "Description": "金山测试",
  "Status": 1,
  "Reason": "",
  "SignName": "金山",
  "RequestId": "c753ee55-7880-4848-8283-934999a86fe5"
}
```

# 删除签名

## 删除签名（DeleteSmsSign）

### 描述

删除签名

### HTTP METHOD

GET

### 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：DeleteSmsSign	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
SignId	Int	是			短信签名ID	

### 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	备注
DeletedTime	String	删除时间	
RequestId	String	请求唯一标识	

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=DeleteSmsSign&Version=2019-05-01&SignId=10000&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "DeletedTime": "2020-12-02 16:56:43",
  "RequestId": "d6cd4ac1-bf18-444a-b62d-27755b79002f"
}
```

# 签名列表

## 描述

获取签名列表

## HTTP METHOD

GET

## 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：ListSigns	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
Page	Int	否			当前页	默认1
PageSize	Int	否			每页条数	默认10

## 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	备注
Signs	array	签名列表	
Total	String	签名总数	
RequestId	String	请求唯一标识	

## 签名详情

名称	类型	描述	备注
Id	Int	签名ID	
Status	Int	签名状态	1待审核；2审核通过；3审核未通过
Name	String	签名名称	
Type	String	签名类型	1公司；2 APP；3网站；4 公众号；5 商标 255 其他
CreatedTime	String	签名申请时间	
AuditedTime	String	签名审核时间	

## 示例（调用成功）

### 请求示例

```
curl -X GET \
  'http://sms.api.ksyun.com/?Action=ListSigns&Version=2019-05-01&Signature=xxxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'
```

### 返回示例

```
{
  "Signs": [
    {
      "Id": 79046,
      "Name": "金山云",
      "Status": 1,
      "Type": 2,
      "CreatedTime": "2021-08-17 10:53:51",
      "UpdatedTime": "",
      "AuditedTime": "",
      "DeletedTime": "",
      "StatusName": "待审核",
      "TypeName": "APP"
    }
  ]
}
```

```

    ],
    "Total": 1,
    "RequestId": "68740cac-1b43-4042-ba65-4240fe96963c"
}

```

## 数据总览

### 国内短信数据总览 (GetInternalSmsOverview)

#### 描述

国内短信数据总览

#### HTTP METHOD

GET

#### 请求参数

公共参数列表 [详情](#)

名称	类型	必须	长度限制	参数格式	描述	备注
Action	String	是	确定值	[a-zA-Z]+	操作接口名，系统规定参数，取值：GetInternalSmsOverview	
Version	String	是	确定值		版本号，系统规定参数，取值：2019-05-01	
BeginDate	String	是			开始日期，例：2020-05-01	
EndDate	String	是			结束日期，例：2020-05-03	
SignId	Int	否			签名ID，控制台《短信服务》国内短信《短信签名》可查	
TemplateId	Int	否			模板ID，控制台《短信服务》国内短信《短信模板》可查	
SmsType	Int	否		类型值：1验证码，2通知，3营销类短信	短信类型	

#### 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	备注
RequestId	String	请求唯一标识	
Stats	列表	短信数据统计列表	

#### Stats定义

名称	类型	描述	备注
SendAmount	Int	发送量	
SuccessAmount	Int	成功量	
ChargingAmount	Int	计费量	
FailAmount	Int	失败量	
SuccessRate	String	成功率	

#### 示例（调用成功）

##### 请求示例

```

curl -X GET \
  'http://sms.api.ksyun.com/?Action=GetInternalSmsOverview&Version=2019-05-01&BeginDate=2020-05-01&EndDate=2020-05-01&Signature=xxx&SignatureVersion=1.0&Timestamp=2019-08-13T17:18:36Z&Service=sms&Accesskey=xxx' \
  -H 'accept: application/json' \
  -H 'Authorization: ****'

```

### 响应成功返回示例

```
{
  "Stats": {
    "2020-05-01": {
      "SendAmount": 1,
      "SuccessAmount": 1,
      "ChargingAmount": 1,
      "FailAmount": 0,
      "SuccessRate": "100.00%"
    },
    "2020-05-02": {
      "SendAmount": 0,
      "SuccessAmount": 0,
      "ChargingAmount": 0,
      "FailAmount": 0,
      "SuccessRate": "0.00%"
    },
    "2020-05-03": {
      "SendAmount": 0,
      "SuccessAmount": 0,
      "ChargingAmount": 0,
      "FailAmount": 0,
      "SuccessRate": "0.00%"
    }
  },
  "RequestId": "4083a4c2-fd22-4f4c-a898-790d7f5c66b1"
}
```

### 响应失败返回示例

```
{
  "RequestId": "ED9B3DF4-75AF-4573-8DD4-7611C57292F2",
  "Error": {
    "Type": "sender",
    "Message": "Action not found",
    "Code": "ActionNotFound"
  }
}
```

访问[API错误码](#)查看更多错误码