

SendCloud 邮件接口 使用指南

版本号 V1.0

武汉闪达科技有限公司
2018. 07

目 录

1.域名操作	3
1.1 添加域名	3
1.2 域名列表	4
1.3 域名修改	5
2.邮件发送	6
2.1 普通发送	6
2.2 添加模板	11
2.3 查询模板	12
2.4 更新模板	14
2.5 删除模板	15
2.6 模板发送	16
2.7 投递回应	18

1. 域名操作

1.1 添加域名

1.1.1 基本请求

Path: /email/domain/add

Method: POST

请求参数:

名称	类型	必须	含义	示例
name	String	Y	域名	test.com

请求示例:

```
POST /email/domain/add HTTP/1.1
```

```
Host: localhost
```

```
Content-Type: application/x-www-form-urlencoded
```

```
name=test.com
```

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200
name	String	域名	test.com
mx.value	String	mx	mx.sendcloud.org
dkim.value	String	dkim	k=rsa;p=....
spf.value	String	spf	v=spf...

返回值示例:

```
{
  "status": true,
  "message": "创建成功",
  "data": null,
  "code": 200,
  "info": {
    "name": "test.com",
    "type": "普通",
    "verify": 0,
    "verifyKey.domain": "test.com",
    "verifyKey.value": "d274e21a-e93f-48dc-87f5-00d0a11fb752-1531218344388",
    "spf.domain": "test.com",
    "spf.value": "v=spf1 include:spf.sendcloud.org ~all",
    "dkim.domain": "mail._domainkey.test.com",
    "dkim.value":
"k=rsa;p=MIGfMA0GCSqGSIlb3DQEBQUAA4GNADCBiQKBgQCpbTH5g7/CBjirjn4ddJFhmWkx\r\nSDZzCzlx
```

```
rrk4C3xciXYShgrjMyxE9yW2eAiYuZgztdRUWO80kkbEc6FhRdJD68Nb\r\nF7sXp0rJkQvusDXuqagqk5NQMt
Gr5jkl0CwWl0aZ0L8cQR9+NaiV2UnxAEidj/hH\r\nJq7pMuZ3Z3zfvPQK+wIDAQAB\r\n",
    "mx.domain": "test.com",
    "mx.value": "mx.sendcloud.org",
    "gmtCreated": "2018-07-10 18:25:44",
    "gmtUpdated": "2018-07-10 18:25:44"
}
}
```

1.1.2 把返回的 spf,dkim,mx 填到自己的域名解析中去，没记住可以通过调用域名查询获得。这三项都配置了就可以进行发信了。



1.2 域名列表

Path: /email/domain/list

Method: POST

请求参数:

名称	类型	必须	含义	示例
name	String	N	域名	test.com

请求示例:

POST /email/domain/listHTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	默认 0, 成功
name	String	域名	Test.com
mx.value	String	mx	mx.sendcloud.org
dkim.value	String	dkim	k=rsa;p=....
spf.value	String	spf	v=spf...

返回值示例:

```

{
  "status": true,
  "message": "查询成功",
  "data": null,
  "code": 0,
  "info": [
    {
      "name": "test.com",
      "type": "普通",
      "verify": 0,
      "verifyKey.domain": "test.com",
      "verifyKey.value": "d274e21a-e93f-48dc-87f5-00d0a11fb752-1531218344388",
      "spf.domain": "test.com",
      "spf.value": "v=spf1 include:spf.sendcloud.org ~all",
      "dkim.domain": "mail._domainkey.test.com",
      "dkim.value":
"k=rsa;p=MIGfMA0GCSqGSIsb3DQEBAQUAA4GNADCBiQKBgQCpbTH5g7/CBjirjn4ddJFhmWkx\r\nSDZzCzlx
rrk4C3xciXYSgrjMyxE9yW2eAiYuZgztdRUWO80kkbEc6FhRdJD68Nb\r\nF7sXp0rJkQvusDXuqagqk5NQMt
Gr5jkl0CwWl0aZ0L8cQR9+NaiV2UnxAEidj/hH\r\nJq7pMuZ3Z3zfvPQK+wIDAQAB\r\n",
      "mx.domain": "test.com",
      "mx.value": "mx.sendcloud.org",
      "gmtCreated": "2018-07-10 18:25:44",
      "gmtUpdated": "2018-07-10 18:25:44"
    }
  ]
}

```

1.3 域名修改

Path: /email/domain/update

Method: POST

请求参数:

名称	类型	必须	含义	示例
name	String	Y	域名(旧)	test.com
newName	String	Y	域名(新)	newtest.com

请求示例:

POST /email/domain/update HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

name=test.com&newName=newtest.com

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200
name	String	域名	test.com
mx.value	String	mx	mx.sendcloud.org
dkim.value	String	dkim	k=rsa;p=...

spf.value	String	spf	v=spf...
-----------	--------	-----	----------

返回值示例:

```
{
  "status": true,
  "message": "修改成功",
  "data": null,
  "code": 200,
  "info": {
    "name": "newtest.com",
    "type": "普通",
    "verify": 0,
    "verifyKey.domain": "newtest.com",
    "verifyKey.value": "d274e21a-e93f-48dc-87f5-00d0a11fb752-1531218344388",
    "spf.domain": "newtest.com",
    "spf.value": "v=spf1 include:spf.sendcloud.org ~all",
    "dkim.domain": "mail._domainkey.newtest.com",
    "dkim.value":
"k=rsa;p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCckL4bp8u2mW2pifxu7tOvY+Hz\r\nppohm
hXCm7G5VKwVeteyM7Aq82gm7c9P+xNs4a+28W5vzBJ+MY1t6OF9vQBBMS6\r\nSMCHD61EDoxNSqTT7
seVn8kffRfDwssNnKgsQvM6ppqAmH7kEDDwrkqVQzBHftfxL\r\nRyiDtZcu8tgYCEoSbWIDAQAB\r\n",
    "mx.domain": "newtest.com",
    "mx.value": "mx.sendcloud.org",
    "gmtCreated": "2018-07-10 18:25:44",
    "gmtUpdated": "2018-07-10 18:48:34"
  }
}
```

2. 邮件发送

邮件发送之前，必须配置域名（调用域名添加接口），并配置相关 mx、spf、dkim

2.1 普通发送

Path: /email/send

Method: POST

请求参数:

名称	类型	必须	含义	示例
xsmtpapi	String	*	SMTP 扩展字段	
emailType	Integer	Y	邮件类型: 0(触发), 1(批量)	1
cc	String	N	抄送地址. 多个地址使用','分隔	A@qq.com;B@qq.com
html	String	Y	邮件的内容.	

			邮件格式为 text/html	
fromName	String	N	发件人名称	ifaxin 客服支持 <support@ifaxin.com>
to	String	*	收件人地址. 多个地址使用';'分隔	ben@ifaxin.com;joe@ifaxin.com
from	String	Y	发件人地址, 为了更高的送达率, 建议 from 域名后缀与发信域名一致。	support@ifaxin.com
subject	String	Y	邮件主题	生日祝福
contentSummary	String	N	邮件摘要. 该字段传入值后, 若原邮件已有摘要, 会覆盖原邮件的摘要; 若原邮件中没有摘要将会插入摘要。	
bcc	String	N	密送地址. 多个地址使用';'分隔	
replyTo	String	N	设置用户默认的回复邮件地址. 多个地址使用';'分隔, 地址个数不能超过 3 个. 如果 replyTo 没有或者为空, 则默认的回复邮件地址为 from	
headers	String		邮件头部信息. JSON 格式, 比如: {"header1": "value1", "header2": "value2"}	

特别说明: 当你使用 xsmtapi, to 参数将会失效 (两者必传一个)

参数说明:

1. 假设 from 为爱发信支持 <support@ifaxin.com>. 如果 fromName 为空, 则系统会将 fromName 设置为"爱发信支持"; 如果 fromName 为非空, 则不作处理.

2. 地址列表发送时, 使用参数 to 指定地址列表, 地址列表中的每个地址是单独发送, 地址列表

的个数不能超过 5. 此时参数 `cc`, `bcc`, `xsmtpapi` 失效.

3.非地址列表发送时, 使用参数 `to` 指定收件人, 多个收件人是广播发送 (收件人会全部显示). 使用参数 `cc` 指定抄送人, 参数 `bcc` 指定密送人.

4.非地址列表发送时, 使用 `xsmtpapi` 指定收件人, 多个收件人是单独发送. 此时参数 `to`, `cc`, `bcc` 失效.

5.参数 `to`, `cc`, `bcc` 的收件人个数不能超过 100, `xsmtpapi` 中的 `to` 的收件人个数不能超过

6.`subject`, `html`, 中都可以使用变量. 由于变量的 `'%'` 为特殊字符, 做 HTTP 请求时请注意处理.

7.X-SMTPAPI 是 SendCloud 为开发者提供的邮件个性化定制的处理方式, 开发者通过这个特殊的 信头扩展字段, 可以设置邮件处理方式的很多参数. 一般的, 开发者在使用 SMTP 接入时会使用此字段. 不过, API 的方式也支持此参数.

SMTP 调用时, 开发者可以在邮件中自行插入各种头域信息, 这是 SMTP 协议所允许的. 而 SendCloud 会检索 key 为 X-SMTPAPI 的头域信息, 如果发现含有此头域, 则其 value 的值可以被解析, 用来改变邮件的处理方式.

X-SMTPAPI 代码示例如下:

```
x_smtpapi = {  
  
    "to": ["d@163.com", 'i@163.com'],  
  
    "sub": {  
  
        "%content%": ['nihao0', 'nihao1']  
  
    },  
  
}  
  
msg['SC-Custom-test_key1'] = "value1";  
  
msg['SC-Custom-test_key2'] = "value2";  
  
msg['X-SMTPAPI'] = Header(base64.b64encode(simplejson.dumps(x_smtpapi)))
```

SMTP 服务器会对邮件中 key 为 X-SMTPAPI 的头域信息做格式检查. 如果不符合上述要求, 则会报 `xsmtpapi error` 的错误.

需要注意的是:

1. SMTP 调用时, X-SMTPAPI 必须是头域字段的最后一个. 否则, 可能导致 `xsmtpapi error` 的错误.

2. API 调用时, 直接传入 JSON 字符串即可, 无需 base64 编码封装.
3. X-SMTPAPI 的总长度不能超过 1M.

value 封装的 JSON 字符串的结构和用途见下:

to 含有收件人地址的数组, 指定邮件的收件人.

```
{  
  
  "to": ["ben@ifaxin.com", "joe@ifaxin.com"]  
  
}
```

注意:

- 这里的 **to** 会覆盖收件人参数 **to**
- 这里的 **to** 的收件人个数不能超过 100

sub 是一个关联数组. 它的 **key** 是「变量」, **value** 是「替换值数组」.

用法解释: 每一个「变量」对应一个「替换值数组」, 在做邮件内容替换时, 每一个「收件人」按其在「收件人数组」中出现的位置使用「替换值数组」中相应位置的值来替换「变量」的值.

请求示例

POST: /email/send HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

emailType=1&html=%3Chtml%3Ehello%3C%2Fhtml%3E&fromName=zhangsan&to=receiver%40qq.com&from=sender%40newtest.com&subject=%E7%94%9F%E6%97%A5%E7%A5%9D%E7%A6%8F

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功

code	Integer	返回码	200
emailIdList	String[]	emailId 数组, 有返回值表明请求成功	

返回值示例:

```
{
  "status": true,
  "message": "",
  "data": null,
  "code": 200,
  "info": {
    "emailIdList":
    [ "1447054895514_15555555_32350_1350.sc-10_10_126_221-inbound0$ben@ifaxin.com",
      "1447054895514_15555555_32350_1350.sc-10_10_126_221-inbound1$joe@ifaxin.com" ]
  }
}
```

2.2 添加模板

Path: /email/template/add

Method: POST

请求参数:

名称	类型	必须	含义	示例
invokeName	String	Y	邮件模板调用名称	abcde
templateType	Integer	Y	邮件模板类型: 0(触发), 1(批量)	1
subject	String	Y	模板标题	生日祝福
html	String	Y	html 格式内容	<html>生日快乐</html>
name	String	Y	邮件模板名称	生日祝福模板

请求示例:

POST /email/template/add HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

```
invokeName=abcde&templateType=1&subject=%E7%94%9F%E6%97%A5%E7%A5%9D%E7%A6%8F&html=%3Chtml%3E%E7%94%9F%E6%97%A5%E5%BF%AB%E4%B9%90%3C%2Fhtml%3E&name=%E7%94%9F%E6%97%A5%E7%A5%9D%E7%A6%8F%E6%A8%A1%E6%9D%BF
```

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200
invokeName	String	邮件模板调用名称	
templateType	Integer	邮件模板类型：0(触发), 1(批量)	
subject	String	模板标题	
html	String	html 格式内容	
name	String	邮件模板名称	

返回值示例：

```
{
  "status": true,
  "message": "修改成功",
  "data": null,
  "code": 200,
  "info": {
    "data": {
      "name": "生日祝福",
      "invokeName": "abcde",
      "templateType": 1,
      "gmtCreated": "2015-10-16 10:42:01",
      "gmtUpdated": "",
      "html": "<html>生日快乐</html>",
      "subject": "生日祝福"
    }
  }
}
```

2.3 查询模板

Path: /email/template/list

Method: POST

请求参数：

名称	类型	必须	含义	示例
invokeName	String	N	邮件模板调用名称	abcde
templateType	Integer	N	邮件模板类型：0(触发), 1(批量)	1

start	Integer	N	查询起始位置, 取值区间 [0-], 默认为 0	
limit	String	N	查询个数, 取值区间 [0-100], 默认为 100	

请求示例:

POST /email/template/list HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200
invokeName	String	邮件模板调用名称	
templateType	String	邮件模板类型: 0(触发), 1(批量)	
subject	String	模板标题	
html	String	html 格式内容	
name	String	邮件模板名称	
templateStat	String	邮件模板状态: -2(未提交审核), -1(审核不通过), 0(待审核), 1(审核通过)	
total	Integer	总条数	
count	Integer	当前返回条数	

返回值示例:

```
{
  "status": true,
  "message": "修改成功",
  "data": null,
  "code": 200,
  "info": {
    "total": 2,
```

```

"dataList": [
  {
    "templateType": 1,
    "templateStat": 1,
    "gmtCreated": "2013-11-21 16:37:41",
    "name": "ifaxin 账单",
    "gmtUpdated": "2015-06-12 19:44:07",
    "invokeName": "****"
  },
  {
    "templateType": 0,
    "templateStat": 1,
    "gmtCreated": "2013-11-21 16:39:53",
    "name": "ifaxin 密码找回",
    "gmtUpdated": "2013-11-21 16:39:53",
    "invokeName": "****"
  }
],
"count": 2
}
}
}

```

2.4 更新模板

Path: /email/template/update

Method: POST

请求参数:

名称	类型	必须	含义	示例
invokeName	String	Y	邮件模板调用名称	abcde
templateType	Integer	N	邮件模板类型: 0(触发), 1(批量)	1
subject	String	N	模板标题	生日祝福
html	String	N	html 格式内容	<html>生日快乐</html>
name	String	N	邮件模板名称	生日祝福模板

请求示例:

POST /email/template/update HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

invokeName=abcde

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200
count	Integer	更新成功数	

返回值示例:

```
{
  "status": true,
  "message": "请求成功",
  "data": null,
  "code": 200,
  "info": {
    "count": 1
  }
}
```

2.5 删除模板

Path: /email/template/delete

Method: POST

请求参数:

名称	类型	必须	含义	示例
invokeName	String	Y	邮件模板调用名称	abcde

请求示例:

POST /email/template/delete HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

invokeName=abcde

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200

count	Integer	删除成功数	
-------	---------	-------	--

返回值示例:

```
{
  "status": true,
  "message": "请求成功",
  "data": null,
  "code": 200,
  "info": {
    "count": 1
  }
}
```

2.6 模板发送

进行模板发送前，必须先添加模板（调用添加模板接口）

Path: /email/sendtemplate

Method: POST

请求参数:

名称	类型	必须	含义	示例
templateInvokeName	String	Y	邮件模板调用名称	abcde
from	String	Y	发件人地址	support@ifaxin.com
to	String	*	地址列表	
xsmtpapi	String	*	SMTP 扩展字段	
subject	String	Y	邮件标题	
fromName	String	N	发件人名称	ifaxin 客服支持 <support@ifaxin.com>
replyTo	String	N	设置用户默认的回复邮件地址. 多个地址使用';'分隔, 地址个数不能超过 3 个. 如果 replyTo 没有或者为	

			空, 则默认的回 复邮件地址 为 from	
headers	String	N	邮件头部信 息. JSON 格 式, 比 如: {"header1 ": "value1", "header2": "value2"}	

特别说明: 当你使用 xsmtapi, to 参数将会失效 (两者必传一个)

参数说明:

- 1.配置好域名之后, 添加相应模板, 即可进行模板发送.
- 2.假设 from 为 爱发信支持 <support@ifaxin.com>. 如果 fromName 为空, 则系统会将 fromName 设置为"爱发信支持"; 如果 fromName 为非空, 则不作处理.
- 3.使用 xsmtapi 指定收件人, 详见 “普通发送” xsmtapi 格式
- 4.subject, html, 中都可以使用变量. 由于变量的 '%' 为特殊字符, 做 HTTP 请求时请注意处理.

请求示例:

```
POST /email/template/delete HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
X-APIGW-MKT-INSTANCEID: test
X-Apig-Uid: test
```

```
templateInvokeName=abcde&from=sender%40newtest.com&xsmtapi=%7B%22to%
22%3A+%5B%22ben%40ifaxin.com%22%2C+%22joe%40ifaxin.com%22%5D%2C%22sub
%22%3A%7B%22%25name%25%22%3A+%5B%22Ben%22%2C+%22Joe%22%5D%2C%22%25mon
ey%25%22%3A%5B%2288%2C+497%5D%7D%7D
```

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200
emailIdList	String[]	emailId 数组, 有返回值表明 请求成功	

返回值示例:

```
{
  "status": true,
```

```

    "message": "请求成功",
    "data": null,
    "code": 200,
    "info": {
        "emailIdList": [

"1447054895514_15555555_32350_1350.sc-10_10_126_221-inbound0$ben@ifaxin.com",

"1447054895514_15555555_32350_1350.sc-10_10_126_221-inbound1$joe@ifaxin.com"

        ]

    }
}
}
}

```

2.7 投递回应

Path: /email/status

Method: POST

请求参数:

名称	类型	必须	含义	示例
email	String	N	收件人地址	
limit	Integer	N	查询个数, 取值区间 [0-100], 默认为 100	
start	Integer	N	查询起始位置, 取值区间 [0-], 默认为 0	
endDate	String	N	结束日期, 格式为 yyyy-MM-dd, 和起始时间间隔不超过 30 天	
startDate	String	N	开始日期, 格式为 yyyy-MM-dd, 和结束时间间隔不超过 30 天	

days	String	N	过去 days 天内的投递数据，(days=1 表示今天), 时间不超过 30 天
emailIds	String	N	调用 api 发送邮件成功返回的 emailId. 多个地址使用 ';' 分隔，如:emailIds=a;b;c

1. 获取投递数据时，必须指定时间区间. 即 startDate 与 endDate 的组合 或者 days 参数需二者取一.
2. 查询的天数不超过三个月.
3. 每次查询默认只查询 100 条记录，如果用户查询满足条件的记录数超过 100 时候，需要人为指定 start 和 limit 参数.
4. 每次查询得到的对应的记录会根据用户所传参数缓存 10 分钟.

请求示例:

POST /email/status HTTP/1.1

Host: localhost

Content-Type: application/x-www-form-urlencoded

返回参数:

名称	类型	含义	示例
status	boolean	操作是否成功	true
message	String	返回描述	创建成功
code	Integer	返回码	200
total	Integer	总条数	
voListSize	Integer	当前返回条数	
emailId	String	邮件 id	
sendLog	String	发送日志	
status	String	发送状态	
recipients	String	收件人	

返回值示例:

```
{
  "status": true,
  "message": "请求成功",
```

