修订记录:

版本	动作	修订内容	修订时间
1.0.0	修订	新增 extra 参数长度校验返回码 10305，10306。 timeToLive 改为最少 60s。 新增 extra 参数长度校验返回码 10305，10306。 新增消息敏感信息错误码 10103。 去掉请求参数中的 regIds、aliases。 新增允许发送时间段判定。 新增消息敏感信息错误码 10103。 新增允许发送时间段判定。 新增消息敏感信息错误码 10103。 取消标签推送接口。	2018-07-19
1.0.1	修订	新增错误码 10072。	2018-08-29

		修改错误码 10153、10154 错误提示信息。	
2.0.0	修订	新增超量限制描述	2018-09-28
2.0.1	修订	删除错误码 10253；新增错误码 10070。 新增错误码 10040	2019-01-09
2.1.0	修订	2.4.4 新增错误码 10307; 2.4.4 新增错误码 10104; 修改错误码 10302 错误信息描述	2019-02-27
2.2.0	修订	单推、批量推新增非法 userId 返回	2019-07-26
2.3.0	新增	新增标签管理相关接口 新增标签分类管理相关接口 新增标签分组管理相关接口 新增标签推接口 新增全局公共业务码详解 新增 10600-10615 返回码详解	2019-08-05
2.4.0	新增	新增查询消息状态接口	2019-08-14

		新增消息是否展示时间字段	
2.5.0	新增	新增订阅解订阅接口	2019-10-09
2.6.0	新增	新增校验无效 useId 接口	2019-10-15
2.7.0	新增	新增 UPA 鉴权接口 新增 UPA 推送接口	2019-12-07
2.8.2	新增	新增消息分类功能 所有推送接口新增消息分类字段 classification	2020-04-27
2.9.0	新增	新增测试模式 单推、批量推新增测试模式功能，测试模式只能给绑定测试用户推送测试消息	2020-05-18
2.9.1	新增	新增配置查询接口 新增错误码：10255, "全量推送接口未开放"	2020-06-04

一. 公共

1.推送超量说明

如果调用接口出现超量限制的返回码，请不要在当天高频次调用服务器相应接口，否则将调低推送量级；

出现“10070：发送总量超出限制”。当天不要调用/message/send 单推接口；

出现“10252：批量发送消息体超出限制”。当天不要调用/message/saveListPayload 保存群推消息公共体接口；

出现“10070：发送总量超出限制”。当天不要调用/message/saveListPayload 保存群推消息公共体接口，以及/message/pushToList 批量推送用户接口；

出现“10254：全量发送超过次数限”。当天不要调用/message/all 全量发送接口；

关于推送量的说明可以参考【vivo 推送常见问题汇总】

2.vivo 服务器地址

https://api-push.vivo.com.cn

3.公共传入参数

HTTP Header 中（推送鉴权接口除外）。

属性名字	类型	是否必填 Y/N	描述
authToken	string	Y	当鉴权成功时会返回该字段，推送消息时需要提供 authToken，有效期默认为 1 天，过期后无法使用

4.高级特性 extra

属性名字	类型	是否必填 Y/N	描述
callback	string	Y	第三方接收回执的 http 接口，最大长度 128 个字符
callback.param	string	N	第三方自定义回执参数，最大长度 64 个字符

二．接口定义

1.鉴权

1.1 推送鉴权接口

接口说明

要想调用 PUSH 接口，任何接入方都要有个鉴权操作。其他接口 header 中必须携带该参数。

接口返回的参数 authToken，一个 appId 可对应多个 token，24 小时过期，业务方做中心缓存，1-2 小时更新一次。

限制：一天限制调用不超过 10000 次。

访问方式

URL	编码方式	协议	method	Content-Type
/message/auth	Utf-8	https	POST	application/json

Curl e.g.:

```
curl -X POST -H    'Content-Type:application/json' -d    '${your_request_body}'    https://api-push.vivo.com.cn/message/auth
```

Request body e.g.:

```
{
      "appId":10004,
        "appKey":"25509283-3767-4b9e-83fe-b6e55ac6243e",
```

```
    "timestamp":1501484120000,
      "sign":"8424f52fd5eaedc16474e4f702d230d2"
}
Response body e.g.:
http status 200:
业务成功:
{
    "result":   0,
    "desc":   "请求成功",
    "authToken":     "24ojds98fu3jqrioeu982134jieds9fq43u09uaf"
}
业务异常:
{
    "result":   xxx,
    "desc":   "xxx 不合法"
```

```
}

http status 500：

Internet server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
appId	int	Y	用户申请推送业务时生成的 appId
appKey	string	Y	用户申请推送业务时获得的 appKey
timestamp	long	Y	Unix 时间戳 做签名用，单位：毫秒，且在 vivo 服务器当前 utc 时间戳前后十分钟区间内。
sign	string	Y	签名 使用 MD5 算法，字符串 trim 后拼接（appId+appKey+timestamp+appSecret），然后通过 MD5 加密得到的值（字母小写）

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败

desc	string	文字描述接口调用情况
authToken	string	当鉴权成功时才会有该字段，推送消息时，需要提供 authToken，有效期默认为 1 天，过期后无法使用。一个 appId 可对应多个 token，24 小时过期，业务方做中心缓存，1-2 小时更新一次。

业务异常码详解：

result	desc
10200	appId 不能为空
10201	appKey 不能为空
10202	appKey 不合法
10203	timestamp 不能为空
10204	sign 不能为空
10205	appId 不存在
10206	sign 不正确
10207	timestamp 不合法
10250	认证接口超过调用次数限制

2.单播

2.1 单推接口

接口说明

接入方携带消息内容以及用户 regId（或 alias）进行通知消息推送。针对每个用户发送不同的通知。

使用场景：如物流、订单状态、游戏预约状态、行程状态、聊天（如微信、评论）等。

限制：根据客户端 SDK 订阅数自动配置，可发送的用户总量可以在开发者后台查看。

访问方式

URL	编码方式	协议	method	Content-Type
/message/send	Utf-8	https	POST	application/json
Curl e.g.: curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://api-push.vivo.com.cn/message/send				

Request body e.g.:

```
{
        "regId":"12345678901234567890123",
    "notifyType":1,
    "title":"标题 1",
    "content":"内容 1",
    "timeToLive":86400,
    "skipType":2,
        "skipContent":"http://www.vivo.com",
      "networkType":"1",
    "clientCustomMap":{
            "key1":"vlaue1",
            "key2":"vlaue2"
    },
    "extra":{
```

```
            "callback":"http://www.vivo.com",
            "callback.param":"vivo"
    },
        "requestId":"25509283-3767-4b9e-83fe-b6e55ac6b123"
}
```

Response body e.g.:

http status 200:

业务成功：

```
{
      "result":    0,
      "desc":    "请求成功",
"taskId": "121397329"
}
```

推送使用的 regid 或 alias 不合法，无法送达：

```
{
```

```
        "result": 10302,
        "desc": "regId 不合法",
        "invalidUser": {
            "status": 1,
            "userid":    "15638535410301000000001"
        }
}
```

业务异常：

```
{
        "result":    xxx,
"desc": "xxx 不合法"
}
```

http status 500：

Internet server error!

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
regId	string	N	应用订阅 PUSH 服务器得到的 id 长度 23 个字符（regId，alias 两者需一个不为空，当两个不为空时，取 regId）
alias	string	N	别名 长度不超过 40 字符（regId，alias 两者需一个不为空，当两个不为空时，取 regId）
notifyType	int	Y	通知类型 1:无，2:响铃，3:振动，4:响铃和振动
title	string	Y	通知标题（用于通知栏消息） 最大 20 个汉字（一个汉字等于两个英文字符，即最大不超过 40 个英文字符）
content	string	Y	通知内容（用于通知栏消息） 最大 50 个汉字（一个汉字等于两个英文字符，即最大不超过 100 个英文字符）
timeToLive	int	N	消息保留时长 单位：秒，取值至少 60 秒，最长 7 天。当值为空时，默认一天
skipType	int	Y	点击跳转类型 1：打开 APP 首页 2：打开链接 3：自定义 4:打开 app 内指定页面

skipContent	string	N	跳转内容 跳转类型为 2 时，跳转内容最大 1000 个字符，跳转类型为 3 或 4 时，跳转内容最大 1024 个字符，skipType 传 3 需要在 onNotificationMessageClicked 回调函数中自己写处理逻辑。关于 skipContent 的内容可以参考【vivo 推送常见问题汇总】
networkType	int	N	网络方式 -1：不限，1：wifi 下发送，不填默认为-1
classification	int	N	消息类型 0：运营类消息，1：系统类消息。不填默认为 0
clientCustomMap	JSON Object	N	客户端自定义键值对 自定义 key 和 Value 键值对个数不能超过 10 个，且长度不能超过 1024 字符, key 和 Value 键值对总长度不能超过 1024 字符。app 可以按照客户端 SDK 接入文档获取该键值对
extra	JSON Object	N	高级特性（详见目录：一.公共——5.高级特性 extra）
requestId	string	Y	用户请求唯一标识 最大 64 字符
pushMode	int	N	推送模式 0：正式推送；1：测试推送，不填默认为 0（测试推送，只能给 web 界面录入的测试用户推送；审核中应用，只能用测试推送）

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败

desc	string	文字描述接口调用情况
taskId	string	任务编号
invalidUser	JSON Object	非法用户信息，包括 status 和 userid，userid 为接入方传的 regid 或者 alias，status 有三种情况：1.userId 不存在；2.卸载或者关闭了通知；3.90 天不在线；4.非测试用户

业务异常码详解：

result	desc
10050	alias 和 regId 不能都为空
10051	暂不支持该消息类型
10054	notifyType 不合法
10055	title 不能为空
10056	title 长度不能超过 40 个字符
10057	content 不能为空
10058	content 长度不能超过 100 个字符
10059	timeToLive 不合法
10060	skipType 不合法

10061	skipType = 2, skipContent 不能为空
10062	skipType = 2, skipContent 不能超过 1000 个字符
10063	skipType = 3, skipContent 不能为空
10064	skipType = 3, skipContent 不能超过 1024 个字符
10065	networkType 不合法
10066	自定义 key 和 Value 键值对个数不能超过 10 个
10067	自定义 key 和 value 键值对不合法
10068	skipType = 4, skipContent 不能为空
10069	skipType = 4, skipContent 不能超过 1024 个字符
10070	发送量总量超出限制
10072	推送速度过快，请稍后再试
10301	alias 长度不能超过 40 个字符
10302	regId 不合法, regId 为无效的 regId, regId 可能已经失效
10352	requestId 不能为空
10353	requestId 长度不能超过 64 个字符

10303	requestId 已存在
10304	extra 包含不支持的 key
10305	extra callback 长度不能超过 128 个字符
10306	extra　callback.param 长度不能超过 64 个字符
10307	alias 不合法，alias 为无效的 alias，　alias 可能已经失效
~~10103~~	~~推送内容含敏感信息（目前单推没有同步返回审计信息）~~
10104	请发送正式信息，请检查 content，不要发送测试内容，正式应用发送的 content 里面不能是纯数字，纯英文，纯符号，符号加数字，包含“测试”字样、大括号、中括号
10040	资源已达上限，稍后重试
10085	title 或 content 不能为纯表情

3.广播

3.1 保存群推消息公共体接口

接口说明

同一条信息覆盖多个用户。此接口需与批量推送用户接口 3.2 配套使用，批量推送接口用此接口生成 taskId 批量发送用户。

使用场景：活动、系统升级提醒等。

限制：默认根据客户端 SDK 订阅数自动配置。

访问方式

URL	编码方式	协议	method	Content-Type
/message/saveListPayload	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://api-push.vivo.com.cn/message/saveListPayload

Request body e.g.:

```
{
    "title":"标题 1",
"content":"内容 1",
    "notifyType":1,
```

```
    "timeToLive":86400,
    "skipType":2,
      "skipContent":"http://www.vivo.com",
    "networkType":"1",
    "clientCustomMap":{
          "key1":"vlaue1",
          "key2":"vlaue2"
},
      "requestId":"25509283-3767-4b9e-83fe-b6e55ac6b123"
}
```

Response body e.g.:

http status 200:

业务成功：

```
{
    "result":   0,
```

```
    "desc":   "请求成功",
"taskId": "342982232646905856"
}
业务异常:
{
    "result":   xxx,
"desc": "xxx 不合法"
}
http status 500:
Internet server error!
```

接口定义

输入参数:

属性名字	类型	是否必填 Y/N	描述
notifyType	int	Y	通知类型 1:无，2:响铃，3:振动，4:响铃和振动

title	string	Y	通知标题（用于通知栏消息） 最大 20 个汉字（一个汉字等于两个英文字符，即最大不超过 40 个英文字符）
content	string	Y	通知内容（用于通知栏消息） 最大 50 个汉字（一个汉字等于两个英文字符，即最大不超过 100 个英文字符）
timeToLive	int	N	消息保留时长 单位：秒，取值至少 900 秒，最长 7 天。当值为空时，默认一天。
skipType	int	Y	点击跳转类型 1：打开 APP 首页 2：打开链接 3：自定义 4:打开 app 内指定页面
skipContent	string	N	跳转内容 跳转类型为 2 时，跳转内容最大 1000 个字符，跳转类型为 3 或 4 时，跳转内容最大 1024 个字符，skipType 传 3 需要在 onNotificationMessageClicked 回调函数中自己写处理逻辑。关于 skipContent 的内容可以参考【vivo 推送常见问题汇总】
networkType	int	N	网络方式 -1：不限，1：wifi 下发送，不填默认为-1
classification	int	N	消息类型 0：运营类消息，1：系统类消息。不填默认为 0
clientCustomMap	JSON Object	N	客户端自定义键值对 自定义 key 和 Value 键值对个数不能超过 10 个，且长度不能超过 1024 字符, key 和 Value 键值对总长度不能超过 1024 字符。app 可以按照客户端 SDK 接入文档获取该键值对。
requestId	string	Y	用户请求唯一标识 最大 64 字符

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
taskId	string	任务编号

业务异常码详解：

result	desc
10054	notifyType 不合法
10055	title 不能为空
10056	title 长度不能超过 40 个字符
10057	content 不能为空
10058	content 长度不能超过 100 个字符
10059	timeToLive 不合法
10060	skipType 不合法
10061	skipType = 2, skipContent 不能为空

10062	skipType = 2, skipContent 不能超过 1000 个字符
10063	skipType = 3, skipContent 不能为空
10064	skipType = 3, skipContent 不能超过 1024 个字符
10065	networkType 不合法
10066	自定义 key 和 Value 键值对个数不能超过 10 个
10067	自定义 key 和 value 键值对不合法
10068	skipType = 4, skipContent 不能为空
10069	skipType = 4, skipContent 不能超过 1024 个字符
10101	消息内容审核不通过
10252	批量发送消息体超出限制
10352	requestId 不能为空
10353	requestId 长度不能超过 64 个字符
10303	requestId 已存在
10071	超出发送时间允许范围
10103	推送内容含敏感信息

10104	请发送正式信息，请检查 content，不要发送测试内容，正式应用发送的 content 里面不能是纯数字，纯英文，纯符号，符号加数字，包含“测试”字样、大括号、中括号
10253	批量发送 regId 或 alais 超过总量限制
10070	发送量总量超出限制
10040	资源已达上限，稍后重试
10085	title 或 content 不能为纯表情

3.2 批量推送用户接口

接口说明

批量发送用户。此接口与群推消息接口 3.1 配套使用，首先通过群推消息接口 3.1 返回 taskId，然后此接口带上 taskId 批量发送用户，可以根据 regIds 或者 aliases 分批调用。每次调用时，regIds 或者 aliases 的个数必须大于等于 2，小于等于 1000。

限制：根据客户端 SDK 订阅数自动配置，可发送的用户总量可以在开发者后台查看。

访问方式

URL	编码方式	协议	method	Content-Type
/message/pushToList	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://api-push.vivo.com.cn/message/pushToList

Request body e.g.:

```
{
    "regIds":[
            "1234567890123456789012",
            "1234567890123456789012"
],
        "taskId":"342982232646905856",
        "requestId":"25509283-3767-4b9e-83fe-b6e55ac6b123"
}
```

Response body e.g.:

http status 200:

业务成功：


```
{
    "requestId":    "25509283-3767-4b9e-83fe-b6e55ac6b123",
    "result": 0,
    "desc": "请求成功",
    "invalidUsers": [
      {
          "status": 1,
          "userid": "12345678901234567890122"
      },
      {
          "status": 1,
          "userid":   "15554239157791000000009"
      }
```

```
    ]
}
```

请求成功，但是请求中的部分 userid 不合法无法送达

```
{
    "requestId":       "25509283-3767-4b9e-83fe-b6e55ac6b123",
    "result": 0,
    "desc": "请求成功",
    "invalidUsers": [
        {
            "status": 1,
            "userid":    "15579140943441000000113"
        },
        {
            "status": 1,
            "userid":    "15613679456951000000074"
```

```
        }
    ]
}

业务异常：
{
        "result":    xxx,
        "desc":    "xxx 不合法"
}
http status 500：
Internet server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述

regIds	string[]	N	regId 列表 个数大于等于 2，小于等于 1000，regId 长度 23 个字符（regIds，aliases 两者需一个不为空，两个不为空，取 regIds）
aliases	string[]	N	别名列表个数大于等于 2，小于等于 1000，长度不超过 40 字符（regIds，aliases 两者需一个不为空，两个不为空，取 regIds）
taskId	string	Y	公共消息任务号，取 saveListPayload 返回的 taskId
requestId	string	Y	请求唯一标识，最大 64 字符
pushMode	int	N	推送模式 0：正式推送；1：测试推送，不填默认为 0（测试推送，只能给 web 界面录入的测试用户推送；审核中应用，只能用测试推送）

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
invalidUser	JSON Object	非法用户信息，包括 status 和 userid，userid 为接入方传的 regid 或者 alias，status 有三种情况：1.userId 不存在；2.卸载或者关闭了通知；3.90 天不在线；4.非测试用户

业务异常码详解：

result	desc
10150	aliases 和 regIds 不能都为空
10154	aliases 个数不在指定范围[2-1000]
10301	alias 长度不能超过 40 个字符
10153	regIds 个数不在指定范围[2-1000]
10302	regId 不合法
10151	taskId 不能为空
10152	taskId 不合法
10253	批量发送 regId 或 alais 超过总量限制
10155	消息不存在或已过期
10352	requestId 不能为空
10353	requestId 长度不能超过 64 个字符
10303	requestId 已存在
10070	发送量总量超出限制
10040	资源已达上限，稍后重试

3.3 全量发送接口

接口说明

向所有设备推送某条消息。

使用场景：活动、系统升级提醒等。

限制：默认是每个 app 每日可发送一条。

访问方式

URL	编码方式	协议	method	Content-Type
/message/all	Utf-8	https	POST	application/json
Curl e.g.: curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://api-push.vivo.com.cn/message/all Request body e.g.: {				

```
       "notifyType":1,
       "title":"标题 1",
"content":"内容 1",
       "timeToLive":86400,
       "skipType":2,
         "skipContent":"http://www.vivo.com",
       "networkType":"1",
       "clientCustomMap":{
            "key1":"vlaue1",
            "key2":"vlaue2"
       },
         "requestId":"25509283-3767-4b9e-83fe-b6e55ac6b123"
}
```

Response body e.g.:

http status 200:

业务成功：

```
{
    "result":  0,
    "desc":  "请求成功",
"taskId":  "12139732"
}
```

业务异常：

```
{
    "result":  xxx,
"desc": "xxx 不合法"
}
```

http status 500：

Internet server error!

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
notifyType	int	Y	通知类型 1:无，2:响铃，3:振动，4:响铃和振动
title	string	Y	通知标题（用于通知栏消息） 最大 20 个汉字（一个汉字等于两个英文字符，即最大不超过 40 个英文字符）
content	string	Y	通知内容（用于通知栏消息） 最大 50 个汉字（一个汉字等于两个英文字符，即最大不超过 100 个英文字符）
timeToLive	int	N	消息保留时长 单位：秒，取值至少 900 秒，最长 7 天。当值为空时，默认一天
skipType	int	Y	点击跳转类型 1：打开 APP 首页 2：打开链接 3：自定义 4:打开 app 内指定页面
skipContent	string	N	跳转内容 跳转类型为 2 时，跳转内容最大 1000 个字符，跳转类型为 3 或 4 时，跳转内容最大 1024 个字符，skipType 传 3 需要在 onNotificationMessageClicked 回调函数中自己写处理逻辑。关于 skipContent 的内容可以参考【vivo 推送常见问题汇总】
networkType	int	N	网络方式 -1：不限，1：wifi 下发送，不填默认为-1
classification	int	N	消息类型 0：运营类消息，1：系统类消息。不填默认为 0
clientCustomMap	JSON Object	N	客户端自定义键值对 自定义 key 和 Value 键值对个数不能超过 10 个，且长度不能超过 1024 字符, key 和 Value 键值对总长度不能超过 1024 字符。app 可以按照客户端 SDK 接入文档获取该键值对
requestId	string	Y	用户请求唯一标识 最大 64 字符

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
taskId	string	任务编号

业务异常码详解：

result	desc
10054	notifyType 不合法
10055	title 不能为空
10056	title 长度不能超过 40 个字符
10057	content 不能为空
10058	content 长度不能超过 100 个字符
10059	timeToLive 不合法
10060	skipType 不合法

10061	skipType = 2, skipContent 不能为空
10062	skipType = 2, skipContent 不能超过 1000 个字符
10063	skipType = 3, skipContent 不能为空
10064	skipType = 3, skipContent 不能超过 1024 个字符
10065	networkType 不合法
10066	自定义 key 和 Value 键值对个数不能超过 10 个
10067	自定义 key 和 value 键值对不合法
10068	skipType = 4, skipContent 不能为空
10069	skipType = 4, skipContent 不能超过 1024 个字符
10352	requestId 不能为空
10353	requestId 长度不能超过 64 个字符
10303	requestId 已存在
10101	消息内容审核不通过
10254	全量发送超过次数限制
10071	超出发送时间允许范围

10103	推送内容含敏感信息
10104	请发送正式信息，请检查 content，不要发送测试内容，正式应用发送的 content 里面不能是纯数字，纯英文，纯符号，符号加数字，包含“测试”字样、大括号、中括号
10040	资源已达上限，稍后重试
10085	title 或 content 不能为纯表情

3.4 获取消息推送的统计值接口

接口说明

获取 taskId 对应的统计信息，taskIds 最多 100 个。

访问方式

URL	编码方式	协议	method	Content-Type
/report/getStatistics	Utf-8	https	GET	application/json
Curl e.g.:				

curl -X GET -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' https://api-push.vivo.com.cn/report/getStatistics?taskIds=123138209,23498102,3240910940,109283018

Request body e.g.:

Response body e.g.:

http status 200:

业务成功:


```
{
    "result":0,
    "desc":"请求成功",
    "statistics":[
  {
     "taskId":  "298475091219",
     "target": 10000,
     "valid": 9500,
     "send": 9000,
     "receive": 9000,
```

```
    "display": 8000,
    "click": 200,
    "targetInvalid":    200,
    "targetUnSub":    200,
    "targetInActive": 100,
    "covered": 200,
    "controlled":    200,
    "targetOffline":    100
},
{
    "taskId":    "298475091220",
    "target": 10000,
    "valid": 9500,
    "send": 9000,
    "receive": 9000,
```

```
        "display": 8000,
        "click": 200,
        "targetInvalid":    200,
        "targetUnSub":    200,
         "targetInActive": 100,
        "covered": 200,
        "controlled":    200,
        "targetOffline":    100
    }
]
}
```

业务异常：

```
{
      "result":    xxx,
      "desc":    "xxx 不合法"
```

```
}

http status 500：

Internet server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
taskIds	string	Y	查询的任务列表 taskIds 用，分隔开 e.g.: "234567,234568"

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
statistics	json array	统计内容

statistics：

属性名字	类型	描述
taskId	string	任务号
target	long	目标总数
targetInvalid	long	不存在的用户总数，根据 id 找不到任何信息
targetUnSub	long	解订阅的用户总数，用户已卸载或者客户端主动调用 turnOffPush()解订阅
targetInActive	long	90 天不在网的用户数，手机 90 天没有联网，信息被后台删除
valid	long	有效目标总数
covered	long	被覆盖的用户消息总数，群推、全推、标签推消息属于营销消息，会被覆盖
controlled	long	被管控的用户消息总数，例如接收的群推消息总数一天超过 5 条
targetOffline	long	推送后不在线的用户数，截止到当前统计时间为止，没有联网的用户数，呈下降趋势
send	long	下发总数
receive	long	到达总数
display	long	展示总数
click	long	点击总数

业务异常码详解：

result	desc
10471	taskIds 数量不能超过 100 个
10472	taskIds 格式错误
10473	taskIds 为空

4.回执

4.1 消息送达回执

接口说明

vivo 推送服务器将已送达或和对应设备的 alias 或者 regId 通过调用第三方设置的回调，http 接口传给开发者服务器。（每次调用后，推送服务器会清空这些数据。）

使用：

① 发送消息接口设置扩展参数 extra（详见目录：一.公共——5.高级特性 extra），包含 callback、callback.param。

②vivo 推送服务器调用第三方设置的 callback url 接口。

访问方式

URL	编码方式	协议	method	Content-Type
第三方设置的 callback url（详见：高级特性 extra）	Utf-8	https/http	POST	application/json

Curl e.g.:

curl -X POST -H 'Content-Type:application/json' -d '${your_request_body}' https://callbackurl

Request body e.g.:

```
{
    "taskId1":{
            "param":"param1",
            "targets":"alias1,alias2,alias3"
    },
    "taskId2":{
```

```
            "param":"param2",
            "targets":"regId1,regId2,regId3"
    }
}
Response body e.g.:
http status 200:
http status 500:
Internet server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
taskId	string	Y	任务编号
param	string	Y	开发者上传的自定义参数值
targets	string	Y	一批 alias 或者 regId 列表，之间是用逗号分割

5.标签管理

5.1 新增标签接口

接口说明

为应用方增加标签，用于后续标签推送。

访问方式

URL	编码方式	协议	method	Content-Type
/tag/add	Utf-8	https	POST	application/json
Curl e.g.: curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tag/add Request body e.g.: { "name": "shenzhen",				

```
        "desc":"深圳用户标签",
      "group":"city"
}
Response body    e.g.:
http status    200:
{
        "result": 0,
        "desc": "成功",
}
业务异常:
{
        "result": xxx,
"desc":    "xxx"
}
http status    500:
```

Internet server error!

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
name	string	Y	标签名称
desc	string	N	标签描述
group	string	N	标签分类名

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况

5.2 更新单个标签接口

接口说明

更新一个标签的信息,更新标签名字、描述、所属分类。

访问方式

URL	编码方式	协议	method	Content-Type
/tag/update	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tag/update

Request body e.g.:

```
{
        "oldName":"shenzhen",
        "newName":"shenzhenbaoan"
}
```

Response body e.g.:

http status 200:

```
{
    "result": 0,
    "desc": "成功"
}
业务异常：
{
       "result": xxx,
"desc":    "xxx"
}
http status    500：
Internet    server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述

oldName	string	Y	旧标签名称
newName	string	Y	新标签名称
desc	string	N	更新标签描述
group	string	N	更新标签所属分类

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况

5.3 给标签添加用户设备接口

接口说明

给标签添加用户设备信息，单次不超过 1000 个设备。

访问方式

URL	编码方式	协议	method	Content-Type
/tag/addMembers	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tag/addMembers

Request body e.g.:

```
{
        "name":"shenzhen",
      "type":1,
        "ids":["15549635570031000000216","15549635980031000001086"]
}
```

Response body e.g.:

http status 200:

```
{
    "result": 0,
    "desc": "success",
```

```
    "data": [
        {
            "status": 1,
            "userid":    "00000000000000000000004"
        },
        {
            "status": 1,
            "userid":    "00000000000000000000005"
        }
    ]
}
```

业务异常：

```
{
       "result": xxx,
"desc":    "xxx"
```

```
}

http status    500:

Internet    server error!
```

接口定义

输入参数:

属性名字	类型	是否必填 Y/N	描述
name	string	Y	标签名字
type	int	Y	用户类型：1 是 regId ,2 是别名
ids	string[]	Y	用户 id

输出参数:

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
data	JSON Array	userid 是非法用户 id，status 有 3 种状态：1 是用户不存在，2 是用户 push 开关关闭，3 是用户 7 天未联网。

5.4 移除标签中的用户设备接口

接口说明

移除标签中的用户设备信息，单次不超过 1000 个设备。

访问方式

URL	编码方式	协议	method	Content-Type
/tag/removeMembers	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tag/removeMembers

Request body e.g.:

```
{
    "name":"shenzhen",
    "type":1,
    "ids":["1555423915779100000009","1555423915779100000008"]
```

```
}

Response body    e.g.:

http status    200:

{

    "result": 0,

    "desc": "success",

    "data": [

       {

            "status": 1,

            "userid":    "1555423915779100000008"

       },

       {

            "status": 1,

            "userid":    "1555423915779100000009"

       }
```

```
    ]
}
业务异常：
{
        "result": xxx,
"desc":    "xxx"
}
http status    500：
Internet    server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
name	string	Y	标签名字
type	int	Y	用户类型：1 是 regId ,2 是别名

ids	string[]	Y	用户 id

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
data	JSON Array	userid 是非法用户 id，status 有 3 种状态：1 是用户不存在，2 是用户 push 开关关闭，3 是用户 7 天未联网。

6.标签分类管理

6.1 新增标签分类接口

接口说明

为应用方增加标签分类，用于后续标签分类管理。

访问方式

URL	编码方式	协议	method	Content-Type
/tagGroup/add	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tagGroup/add

Request body e.g.:

```
{
    "name": "guangzhou",
"type":"city"
}
```

Response body e.g.:

http status 200:

```
{
    "result": 0,
    "desc": "成功",
```

```
}

业务异常：

{

        "result": xxx,

"desc":     "xxx"

}

http status     500：

Internet     server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
name	string	Y	标签分类名称
type	int	Y	标签分类类型：1 是普通标签，2 是互斥标签
desc	string	N	标签分类描述

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况

6.2 更新单个标签分类接口

接口说明

更新一个标签分类的信息，包括标签分类名称和描述。

访问方式

URL	编码方式	协议	method	Content-Type
/tagGroup/update	Utf-8	https	POST	application/json
Curl e.g.： curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tagGroup/update				

Request body e.g.:

```
{
    "oldName": "city",
    "newName":  "province",
    "desc": "这是更新后的描述"
}
```

Response body e.g.:

http status 200:

```
{
    "result": 0,
    "desc": "成功"
}
```

业务异常:

```
{
      "result": xxx,
```

```
"desc":    "xxx"

}

http status    500:

Internet    server error!
```

接口定义

输入参数:

属性名字	类型	是否必填 Y/N	描述
oldName	string	Y	旧标签分类名称
newName	string	Y	新标签分类名称
desc	string	N	新标签分类的描述

输出参数:

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况

6.3 添加标签到标签分类接口

接口说明

把标签添加进标签分类里面，一次可添加一个或者多个标签，最多不超过 100 个。

访问方式

URL	编码方式	协议	method	Content-Type
/tagGroup/addToGroup	Utf-8	https	POST	application/json

Curl e.g.:

```
curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tagGroup/addToGroup
```

Request body e.g.:

```
{
    "name":"city",
      "tagList":["shenzhen","guangzhou"]
}
```

```
Response body    e.g.:

http status    200:

{

    "result": 0,

    "desc": "成功"

}

业务异常：

{

       "result": xxx,

"desc":    "xxx"

}

http status    500：

Internet    server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
name	string	Y	标签分类名称
tagList	string[]	Y	一个或多个标签

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况

7.标签组合管理

7.1 新增标签组合接口

接口说明

为应用方增加标签组合。 一个标签组合由多个标签组成。

访问方式

URL	编码方式	协议	method	Content-Type
/tagSegment/add	Utf-8	https	POST	application/json

Curl e.g.:

```
curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tagSegment/add
```

Request body e.g.:

```
{
    "name": "segment1",
        "expression":{
                "notTags":["shenzhen"],
                "andTags":[],
                "orTags":[]
        }
}
```

Response body e.g.:

```
http status    200:

{

       "result": 0,

       "desc": "成功",

}

业务异常:

{

       "result": xxx,

"desc":    "xxx"

}

http status    500:

Internet    server error!
```

接口定义

输入参数:

属性名字	类型	是否必填 Y/N	描述
name	string	Y	标签组合名称

expression	JSON Object	Y	标签组合表达式：包含三种 JSON 数组：notTags 表示非操作、andTags 表示与操作，orTags 表示或操作，例如："tagExpression":{
			"notTags":["琴","棋","书","画"],
			"andTags":["深圳","广东"],
			"orTags":["本科生","研究生"]
			}
			可以表示为，不会琴、棋、书、画的广东省深圳市的本科生或研究生的所有用户。
			当然这些标签需要先在平台或者接口创建后再使用。

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况

7.2 更新单个标签组合接口

接口说明

更新一个标签组合的信息

访问方式

URL	编码方式	协议	method	Content-Type
/tagSegment/update	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/tagSegment/update

Request body e.g.:

```
{
        "oldName":"segment1",
        "newName":"segment2",
"expression":   {
                "orTags": ["shenzhen"],
                "andTags":   [],
```

```
                "notTags": []
    }
}
Response body    e.g.:
http status    200:
{
    "result": 0,
    "desc": "成功"
}
业务异常:
{
       "result": xxx,
"desc":    "xxx"
}
http status    500:
```

Internet server error!

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
oldName	string	Y	旧标签组合名称
newName	string	Y	新标签组合名称

expression	JSON Object	N	新标签组合表达式：包含三种 JSON 数组：notTags 表示非操作、andTags 表示与操作，orTags 表示或操作，例如："tagExpression":{
			"notTags":["琴","棋","书","画"],
			"andTags":["深圳","广东"],
			"orTags":["本科生","研究生"]
			}
			可以表示为，不会琴、棋、书、画的广东省深圳市的本科生或研究生的所有用户。
			当然这些标签需要先在平台或者接口创建后再使用。

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况

8.标签推

8.1 标签推接口

接口说明

接入方携带消息内容、标签信息进行通知消息推送。

访问方式

URL	编码方式	协议	method	Content-Type
/message/tagPush	Utf-8	https	POST	application/json
Curl e.g.:				

```
curl -X POST     -H 'authToken:${your_auth_token}' -H     'Content-Type:application/json'   -d '${your_request_body}'        https://host:port/message/tagPush

Request     body e.g.:

{

        "tagExpression":{

                "notTags":["tag1","       tag2","tag3"," tag4"],

                "andTags":["tag5","     tag6"],

                "orTags":["tag7","tag8"]

        },

  "segmentName": "组合名",

    "notifyType": 4,

    "title": "123",

    "content": "456",

    "timeToLive": 86400,

    "skipType": 1,

    "skipContent":     "assda",
```

```
    "networkType": -1,
    "requestId" :       "25509283-3767-4b9e-83fe-b6e55ac6b123"
}
Response body     e.g.:
http status     200:
业务成功:
{
    "result": 0,
    "desc": "请求成功",
    "taskId":     "2199912448"
}
业务异常:
{
       "result": xxx,
"desc":     "xxx 不合法"
```

```
}

http status    500:

Internet    server error!
```

接口定义

输入参数:

属性名字	类型	是否必填 Y/N	描述

tagExpression	JSON Object	Y	标签表达式。包含三种 JSON 数组：notTags 表示非操作、andTags 表示与操作，orTags 表示或操作，例如："tagExpression":{
			"notTags":["琴","棋","书","画"],
			"andTags":["深圳","广东"],
			"orTags":["本科生","研究生"]
			}
			可以表示为，不会琴、棋、书、画的广东省深圳市的本科生或研究生的所有用户。
			当然这些标签需要先在平台或者接口创建后再使用。

segmentName	string	N	之前创建好的组合名，如果传递组合名，默认用组合名推送，否则如果传递 tagExpression，按照标签表达式推送
notifyType	int	Y	通知类型 1:无，2:响铃，3:振动，4:响铃和振动
title	string	Y	通知标题（用于通知栏消息） 最大 20 个汉字（一个汉字等于两个英文字符，即最大不超过 40 个英文字符）
content	string	Y	通知内容（用于通知栏消息） 最大 50 个汉字（一个汉字等于两个英文字符，即最大不超过 100 个英文字符）
timeToLive	int	N	消息保留时长 单位：秒，取值至少 60 秒，最长 7 天。当值为空时，默认一天
skipType	int	Y	点击跳转类型 1：打开 APP 首页 2：打开链接 3：自定义 4:打开 app 内指定页面
skipContent	string	N	跳转内容 跳转类型为 2 时，跳转内容最大 1000 个字符，跳转类型为 3 或 4 时，跳转内容最大 1024 个字符，skipType 传 3 需要在 onNotificationMessageClicked 回调函数中自己写处理逻辑。
networkType	int	N	网络方式 -1：不限，1：wifi 下发送，不填默认为-1
classification	int	N	消息类型 0：运营类消息，1：系统类消息。不填默认为 0
clientCustomMap	JSON Object	N	客户端自定义键值对 自定义 key 和 Value 键值对个数不能超过 10 个，且长度不能超过 1024 字符, key 和 Value 键值对总长度不能超过 1024 字符。
requestId	string	Y	用户请求唯一标识 最大 64 字符

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
taskId	string	任务编号

9.消息状态查询

9.1 消息状态查询

接口说明

调用接口传递消息 id，查询消息的状态。目前只返回审计相关信息，单推没有同步返回消息审计状况，其他类型的推送有同步返回。目前有调用频率限制，单个 app 1 次/s，建议一批类似的消息体查询一次。

使用：

① 发送消息后查询消息的状态(延时在一分钟内)。

② ‘审计通过’ 或 ‘审计不通过的状态还未入库’ ，返回 【消息状态未知】，审计不通过状态已入库，返回【消息包含敏感词】

访问方式

URL	编码方式	协议	method	Content-Type
https://api-push.vivo.com.cn/search/msgStatus	Utf-8	https/http	GET	application/json

Curl e.g.:

curl -X GET -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://api-push.vivo.com.cn/search/msgStatus?taskIds=taskId1,taskId2

Request body e.g.:

```
{
    "result": 0,
    "desc": "success",
    "data": [
        {
```

```
            "taskId": taskId1,
            "desc": "消息包含敏感词"
        },
        {
            " taskId ": taskId2,
            "desc": "消息状态未知"
        }
    ]
}Response body e.g.:
http status 200:
http status 500：
Internet server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
taskIds	string	Y	由多个 taskId 组成的字符串，多个 taskId 之间用逗号分隔，一次最多查询 100 个

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
data	json array	查询数据

data：

属性名字	类型	描述
taskId	string	消息 id
desc	string	消息状态描述

10. 查询失效 id

10.1 失效 id 查询

接口说明

由于用户卸载、客户端主动调用 turnOffPush()解订阅、设备 90 天不联网还有一部分历史原因，可能造成推送的 id 无效，导致在推送时消耗额度，开发者可以根据推送时返回的 InvalidUser 对失效 id 做标记或者清除处理，或者单独查询本接口，将无效的 id 过滤，避免占用推送额度。

使用：

① 传递要查询的 id，以及 id 的类型，后台根据保存的 id，返回无效的 id。

② 建议在推送低峰期，例如晚上或者凌晨调用，将保存的 id 一批批查询，避免影响应用的推送性能。勿高频调用。

访问方式

URL	编码方式	协议	method	Content-Type
/invalidUser/check	Utf-8	https/http	POST	application/json
Curl e.g.： curl -X GET -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://api-push.vivo.com.cn/invalidUser/check				

Request body e.g.:

```
{
    "userType": 1,
    "userIds": [
        "15547801220021000000000",
        "15547801220021000000001"
    ]
}
```

Response body e.g.:

http status 200:

```
{
    "result": 0,
    "desc": "请求成功",
    "invalidUsers": [
        "15547801220021000000000",
        "15547801220021000000001"
```

```
    ]

}

http status 500：

Internet server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
userIds	string[]	Y	用户 id 数组 ， 单次最大 100 个
userType	int	Y	userId 的类型 1:regId,2:alias

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
invalidUsers	json array	无效用户列表

11.UPA（统一推送联盟）鉴权

11.1 UPA 推送鉴权接口

接口说明

要想调用 PUSH 接口，任何接入方都要有个鉴权操作。其他接口 header 的 Authorization 设置该接口返回的参数 access_token。

限制：一天限制调用不超过 10000 次。

访问方式

URL	编码方式	协议	method	Content-Type
/v3/message/auth	Utf-8	https	POST	application/json
Curl e.g.: curl -X POST -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/v3/message/auth Request body e.g.: { "grant_type":"client_credentials",				

```
        "app_id":10004,
        "app_secret":"25509283-3767-4b9e-83fe-b6e55ac6243e",
        "timestamp":1501484120000
}
Response body e.g.:
http status 200:
业务成功：
{
          "result":     0,
          "desc":     "请求成功",
"access_token":     "24ojds98fu3jqrioeu982134jieds9fq43u09uaf",
"expires_in": 86400
}
业务异常：
{
```

```
        "result":    xxx,

        "desc":    "xxx 不合法"

}



http status 500：

Internet server error!
```

接口定义

输入参数：

属性名字	类型	是否必填 Y/N	描述
grant_type	string	Y	依据 Oauth 2.0 规范，本值必须为“client_credentials”
app_id	string	Y	用户申请推送业务时生成的 AppID（最长 24 字节）
timestamp	string	Y	时间戳（单位 ms，Unix 标准，从 1970 年 1 月 1 日开始）
app_secret	string	Y	用户申请推送业务时获得对应 AppID 的秘钥（最长 128 字节）

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
access_token	string	当鉴权成功时才会有该字段，权限令牌推送消息时，需要提供 access_token
expires_in	int	access_token 的有效期，以秒为单位。

12.UPA（统一推送联盟）推送

12.1 UPA 推送接口

接口说明

接入方携带消息内容以及用户 registration_token（同 regId）进行消息推送。当传递多个 registration_token，将进行群推，群推有消息体条数限制。

限制：由我司运营人员配置。

访问方式

URL	编码方式	协议	method	Content-Type
/v3/message/send	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H 'Authorization:${your_access_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/v3/message/send

Request body e.g.:

```
{
    "registration_tokens": [
      "1234567890123"
    ],
    "notification": {
      "title": "通知栏消息",
      "content": "这是个通知栏消息",
```

```
      "click_action": {
        "intent": "Test#TestIntent"
      }
    },
    "original_source_name": "vivo",
    "original_source_ip": "10.12.3.52",
    "ttl": "10000",
    "option": {
      "key1": "value1 ",
      "key2": "value2"
    }
}
Response body e.g.:
http status 200:
业务成功:
```

```
{
        "result":    0,
        "desc":    "请求成功",
"message_id":    "121397329"
}
业务异常:
{
        "result":    xxx,
"desc": "xxx 不合法"
}

http  status  500:
Internet  server  error!
```

输入参数:

属性	类型	父项目	是否必传	描述

registration_tokens	JSON Array		是	多个 registration_token 列表,最大值不超过 100.
notification	JSONObject		否	通知栏消息结构体
ttl	string		是	消息保留时长（单位：秒，最长不超过 7 天）
option	JSONObject		否	消息扩展信息，自定义 key 和 Value 键值对个数不能超过 10 个，且长度不能超过 1024 字符, key 和 Value 键值对总长度不能超过 1024 字符。
original_source_name	string		是	推送消息的来源，品牌名称（英文或拼音）（不超过 128 字节）
original_source_ip	string		是	如果消息由第三方代发， 消息中要挟带消息原始来源的 ip 地址（举例：CP 1 为原始推送消息发出方，将消息发送给了 CP 2，由 CP 2 代为发送至厂商，则 CP 2 发出的消息中需要增加 CP 1 的 IP 地址。）
title	string	notifaication	是	通知标题（用于通知栏消息） 最大 20 个汉字（一个汉字等于两个英文字符，即最大不超过 40 个英文字符）
content	string	notifaication	是	通知内容（用于通知栏消息） 最大 50 个汉字（一个汉字等于两个英文字符，即最大不超过 100 个英文字符）
click_action	JSONObject	notifaication	否	点击通知栏后自定义行为
url	string	click_action	否	点击通知栏消息，打开指定的 URL 地址（不能超过 1024 个字符）
intent	string	click_action	否	点击通知栏消息，用户收到通知栏消息后点击通知栏消息打开应用定义的这个 Intent 页面（不能超过 1024 个字符）

输出参数：

属性	类型	描述
result	int	推送结果
desc	string	结果描述
message_id	string	任务编号（消息 id）

13.消息回收

13.1 消息回收接口

接口说明

此接口用于回收已发送给用户的消息。通过传递推送的消息 id 和具体的用户 id、用户类型进行消息回收，消息回收接口调用量和调用速度会占用推送量级和推送速度。

限制：由我司运营人员配置。

消息撤回功能是平台提供给开发者紧急处理运营事故的工具，切勿日常频繁使用，以免对用户体验不当影响。

对于推送内容、应用行为等存在违规的，平台将按《vivo 推送运营规则》进行相应处罚。

访问方式

URL	编码方式	协议	method	Content-Type
/message/recycle	Utf-8	https	POST	application/json

Curl e.g.:

curl -X POST -H ' authToken:${your_auth_token}' -H 'Content-Type:application/json' -d '${your_request_body}' https://host:port/message/recycle

Request body e.g.:

```
{
        "taskId":656079923690082304,
        "userIds":
[
"15762048080021000000007",
"15762048080021000000008",
"15762048080021000000009"
```

```
],
        "userType":1
}
Response body e.g.:
http status 200:
业务成功:
{
        "result":    0,
        "desc":    "请求成功",
"invalidUsers": [
        {
          "status": 1,
          "userid":    "15762048080021000000008"
          },
        {
```

```
            "status": 1,
            "userid":    "15762048080021000000008"
        }
      ]
}
业务异常:
{
        "result":    xxx,
"desc": "xxx 不合法"
}

http status 500:
Internet server error!
```

接口定义

输入参数:

属性名字	类型	是否必填 Y/N	描述
taskId	string	Y	公共消息任务号，取 saveListPayload 返回的 taskId
userIds	string[]	Y	userId 列表。个数不能大于 100
userType	int	Y	用户类型。1：regId；2：alias

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
invalidUsers	JSON Array	非法用户信息，包括 status 和 userid，userid 为接入方传的 regid 或者 alias，status 有三种情况：1 userId 不存在 2 卸载或者关闭了通知 3 七天不在线

14.应用配置信息查询

14.1 应用配置信息查询接口

接口说明

此接口用于查询应用配置信息，系统消息总量、运营消息总量、群推消息体总量、配置速度等信息、SDK 订阅数。此接口限制了调用频率，请勿频繁调用。

访问方式

URL	编码方式	协议	method	Content-Type
/report/getAppConfig	Utf-8	https	GET	application/json

Curl e.g.:

```
curl -X GET -H 'authToken:${your_auth_token}' -H 'Content-Type:application/json'   -d '${your_request_body}'    https://host:port/report/getAppConfig
```

Response body e.g.:

http status 200:

业务成功:

```
{
    "result": 0,
    "desc": "请求成功",
```

```
    "data": {
        "sysMsgCount": 10000,
        "marketMsgCount": 100000,
        "sdkSubCount": 124,
        "speed": 3000,
        "groupMsgBodyCount": 12000
    }
}业务异常:
{
      "result":    xxx,
       "desc": "xxx 不合法"
}

http status 500:
Internet server error!
```

接口定义

输出参数：

属性名字	类型	描述
result	int	接口调用是否成功的状态码 0 成功，非 0 失败
desc	string	文字描述接口调用情况
data	json object	详细内容

data：

属性名字	类型	描述
sysMsgCount	long	系统消息配置量
marketMsgCount	long	运营消息配置量
sdkSubCount	long	sdk 订阅数
speed	long	速度配置
groupMsgBodyCount	long	群推消息体配置量

全局公共返回码详解

result	desc
0	请求成功
10000	权限认证失败
10040	资源已达上限，稍后重试
10044	该接口未开放测试模式
10045	应用审核中不可发送正式消息
10050	alias 和 regId 不能都为空
10051	暂不支持该消息类型
10054	notifyType 不合法
10055	title 不能为空
10056	title 长度不能超过 40 个字符
10057	content 不能为空
10058	content 长度不能超过 100 个字符
10059	timeToLive 不合法

10060	skipType 不合法
10061	skipType = 2, skipContent 不能为空
10062	skipType = 2, skipContent 不能超过 1000 个字符
10063	skipType = 3, skipContent 不能为空
10064	skipType = 3, skipContent 不能超过 1024 个字符
10065	networkType 不合法
10066	自定义 key 和 Value 键值对个数不能超过 10 个
10067	自定义 key 和 value 键值对不合法
10068	skipType = 4, skipContent 不能为空
10069	skipType = 4, skipContent 不能超过 1024 个字符
10070	运营消息发送量总量超出限制
10071	超出发送时间允许范围
10072	推送速度过快，请稍后再试
10073	系统消息发送量总量超出限制
10082	系统消息未开通

10084	查询速度过快，请稍后再试
10085	title 或 content 不能为纯表情
10101	消息内容审核不通过
10102	服务器端未知异常
10103	推送内容含敏感信息
10104	请发送正式信息，请检查 content，不要发送测试内容
10110	请配置商业化消息发送频率
10111	今天已发送商业化消息
10112	商业化消息发送超出周次数限制
10150	aliases 和 regIds 不能都为空
10151	taskId 不能为空
10152	taskId 不合法
10153	regIds 个数不在指定范围[2-1000]
10154	aliases 个数不在指定范围[2-1000]
10155	消息不存在或已过期

10200	appId 不能为空
10201	appKey 不能为空
10202	appKey 不合法
10203	timestamp 不能为空
10204	sign 不能为空
10205	appId 不存在
10206	sign 不正确
10207	timestamp 不合法
10250	认证接口超过调用次数限制
10252	批量发送创建消息体超出限制
10253	批量发送 regId 或 alais 超过总量限制
10254	全量发送超过次数限制
10301	alias 长度不能超过 40 个字符
10302	regId 不合法, regId 为无效的 regId, regId 可能已经失效
10303	requestId 已存在

10304	extra 包含不支持的 key
10305	extra callback 长度不能超过 128 个字符
10306	extra callback.param 长度不能超过 64 个字符
10307	alias 不合法
10352	requestId 不能为空
10353	requestId 长度不能超过 64 个字符
10354	Internet server error
10471	taskIds 数量不能超过 100 个
10472	taskIds 格式错误
10473	taskIds 为空
10600	name 参数不合法,只能使用中文、英文字母、阿拉伯数字和下划线命名,且不能以数字开头，且长度不超过 50 字符
10601	标签名字不能为空
10602	标签描述不能超过 300 字符
10603	oldName 参数不能为空
10604	newName 参数不能为空

10605	oldName 参数不合法,只能使用中文、英文字母、阿拉伯数字和下划线命名，且不能以数字开头，且长度不超过 50 字符
10606	newName 参数不合法,只能使用中文、英文字母、阿拉伯数字和下划线命名，且不能以数字开头，且长度不超过 50 字符
10607	id 参数不能为空
10608	type 参数不能为空
10609	ids 参数不能为空
10610	标签名已经存在
10611	标签分类名已经存在
10612	标签组合名已经存在
10613	group 参数不合法,只能使用中文、英文字母、阿拉伯数字和下划线命名，且不能以数字开头，且长度不超过 50 字符
10614	ids 数量不能超过 1000
10615	tag 数量不能超过 100
10616	type 参数不合法
10700	userids 不能为空
10701	userid 个数超过限制
10702	optionType 错误

10703	register 接口调用超频，稍后再试
10704	register 调用过快，稍后再试
10705	app 不支持
10706	userType 错误
10800	registration_tokens 个数不在指定范围
10801	notification 不能为空
10802	original_source_name 不能为空
10803	original_source_name 长度非法
10804	original_source_ip 不能为空
10806	click_action 非法
10807	url 长度超过限制
10808	intent 长度超过限制
10255	全量推送接口未开放