T-UPA0002-2019 V1.1.0 (2019-6)

统一推送通道层接口规范

Interface Standard for Unified Push Channel Layer

Unified Push Alliance
统 一 推 送 联 盟

目录

前言

这一技术文稿是由统一推送联盟（UPA）技术标准组攥写。

项目过程中采用三位编码的原则，格式如下：

Vx.y.z，初始版本号为V0.0.0。

其中，x 在每一次大版本发布的时候加 1；

y 在有技术性变更的时候加 1，如增加删减功能模块；

z 在有编辑性改动的时候加 1，如格式、段落调整。

本标准起草单位：中国信息通信研究院、华为技术有限公司、北京小米移动软件有限公司、广东欧珀移动通信有限公司、维沃移动通信有限公司、浙江每日互动网络科技股份有限公司、深圳市和讯华谷信息技术有限公司、珠海市魅族通讯设备有限公司、阿里巴巴（中国）有限公司、北京百度网讯科技有限公司深圳市腾讯计算机系统有限公司、北京爱奇艺科技有限公司、北京北信源软件股份有限公司、平安科技（深圳）有限公司、深圳兆日科技股份有限公司、北京云中融信网络科技有限公司、努比亚技术有限公司、联想（北京）有限公司、北京奇虎 360 科技有限公司、展讯通信（上海）有限公司、深圳市金立通信设备有限公司、锤子软件（北京）有限公司、深圳市万普拉斯科技有限公司、厦门美图移动科技有限公司、青岛海信移动通信技术股份有限公司、中国移动通信集团终端有限公司、新浪网技术（中国）有限公司、联通宽带在线有限公司、中国电信股份有限公司-综合平台开发运营中心、网易（杭州）网络有限公司、小沃科技有限公司、中国移动通信集团公司、北京字节跳动科技有限公司。

1 范围

安卓生态圈一直为 App 开发者提供了一个开放的运行环境，用于实现各种创新的想法，然而相应地也产生了相应的性能问题。消息推送是 App 运营的重要一环，为了优化消息推送成功率，降低电量和流量消耗，改善用户的使用体验，系统级的推送服务显得尤为重要，各大手机厂商也已经提供或者正在研发基于各自系统平台的系统推送服务解决方案。

故中国信息通信研究院泰尔终端实验室联手各大手机厂商和提供推送服务的互联网厂家成立“统一推送联盟”，通过标准化统一通道层，降低终端功耗、提升用户体验、支撑开发者生态建设，促进安卓推送服务行业健康发展，为终端用户提供更好的手机使用体验，为应用开发者解决消息推送需求。

本文档适用于 Push 通道厂商、App 开发者、第三方 Push 服务提供商。

Push 通道厂商：指提供终端设备，并在终端设备操作系统层内置 Push 通道，能够对 Push 通道进行管控，在黑屏后能够保持 Push 通道的厂商。

App 开发者: 指经有效申请并经过 Push 通道厂商/第三方 Push 服务提供商同意，使用 Push 服务，具备民事行为能力的个人、法人或其他组织。以下简称开发者。

第三方 Push 服务提供商：与 Push 通道厂商对应，指不提供终端设备，提供 Push 集成服务的厂商。

本接口规范建议文档包含以下建议内容：

- 整体技术架构
- 基础功能清单
- UPS-Server API 接口规范
- 终端 UPS 服务接口规范

2 定义和缩略语

2.1 定义

术语名称	含义
推送服务器 UPS-SERVER	推送服务器接收并处理从应用开发者发起的请求消息，向 Push 客户端发送 Push 内容。
Push 客户端 UPS-Client	安装终端上的 Push 客户端，负责统一与 UPS-SERVER 交互的，将消息转给发其他 App。
App 开发者 App Developer	App 的所有者，App 开发者应为具备民事行为能力的个人、法人或其他组织。
App 服务器 App Provider Server	App 开发者的服务器，调用 UPS-SERVER 推送消息

App 客户端 App Client	安装在手机上的 App 客户端，与 Push 客户端交互，接受消息。
UPS-SDK	提供有 App 客户端集成的 SDK，提供接口给有 App 客户端，与 Push 客户端交互。
通知栏消息	通知栏消息是指消息通过 UPS-SERVER 发送到 Push 客户端时，使用 Push 通道厂商默认的消息呈现和点击动作
留存用户	针对具体 App，在 3 个月内注册的，并且登陆过系统，并且未被明确告知卸载的逻辑用户数（非独立终端数）。
活跃用户	针对具体 App，在 1 个月内登陆过系统，并且未被明确告知卸载的逻辑用户数。
无效用户	针对具体 App，没有对应 Registration token 的逻辑用户或者不再活跃用户列表中的逻辑用户。
当日活跃用户	针对具体 App，在查询当天或者指定具体某日期中登陆过系统的逻辑用户。
消息接收总数	针对具体 App，通过单推和 to-List API 接口或者 UPS-Web 中的单推和 to-List 推送请求中包含的逻辑用户数。
有效接收消息数	针对［消息接收总数］中去除［无效用户］的消息数
在线下发数	在针对某逻辑用户下发时，此用户正有效登陆在系统上，并且消息下发成功的消息总数。
离线下发数	在针对某逻辑用户下发时，此用户未有效登陆在系统上，并且消息无法在即时下发成功，而后用户在消息有效期内登陆到系统并成功下发该消息的总数。
实际下发数	[在线下发数] + [离线下发数]
终端接收消息数	推送系统服务器成功将消息传递给终端，并且收到终端回执的消息总数。
终端展示消息数	终端在接收到消息后，成功展示在通知栏的消息总数。
消息点击数	消息展示在通知栏后，被用户最终点击的消息总数。

2.2 缩略语

UPS	Unified Push Service
API	Application Program Interface
SDK	Software Development Kit

3 概要

3.1 整体技术架构

本标准的技术原则为：

- 减轻 SDK，尽可能只定义接口，将实现隐藏到服务侧
- 屏蔽不同厂商（包括第三方推送）的底层技术差异
- 基本行为一致性，保证不同的系统上有相同的基本行为，降低不必要的维护成本技术架构：

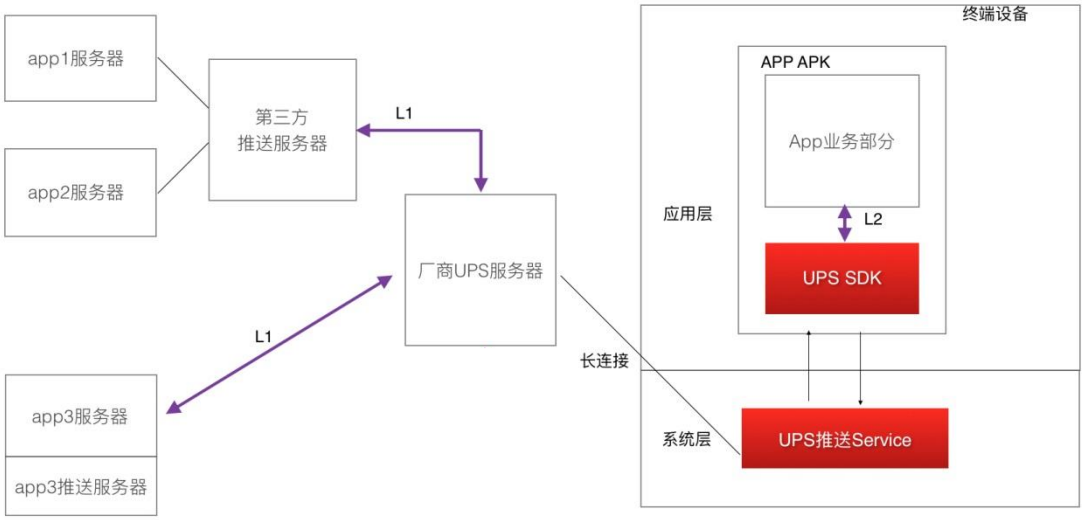


图 1 统一推送的系统框图

图 1 中，厂商 UPS 服务器通过统一接口接收 APP 的推送请求。并和终端系统中的 UPS 推送 Service 保持唯一长连接。UPS SDK 提供了标准的 API，实现 APP 与 UPS 服务器之间的交互。

3.2 标准接口

本标准接口分为两个部分：

L1 接口：连接 APP-Server 和 UPS-Server. 主要用于应用进行推送请求的鉴权、提交推送请求。

L2 接口：让 App 业务部分可以通过 UPS SDK/第三方接口层来使用在手机上的推送模块，最终连接到推送服务系统，完成使用推送所需要的基础工作。

3.3 基础功能要求

3.3.1 UPS-Server API 功能清单

- 采用 HTTPS 或者 socket 形式的接口，建议支持 HTTP2.0，避免大量推送时造成连接数压力，提高推送效率和稳定性；
- 支持指定单个 Registration_token 或一组 Registration_token 列表进行消息推送

3.3.2 终端侧 UPS 服务功能清单

- UPS SDK 提供初始化接口，传入所需的应用信息（如 AppId），实现设备注册，并返回 Registration_token 给 APP，用于后续进行消息推送；
- UPS 服务可以独立完成通知消息展示
- UPS SDK 可以打开，关闭和注销推送通道的功能。
- UPS SDK 提供查询当前 Registration_token 的接口；

4 UPS -server API 相关接口（L1）

4.1 基本功能接口

接口编号	接口描述	备注
L1	App 服务端和推送服务端之间的 API 接口集	
L1/Auth	推送鉴权	操作接口
L1/Send	推送	操作接口

4.1.1 推送鉴权(L1/ Auth)

用于应用推送用户身份鉴定，通过后可获得 access_token 用于后续请求身份校验。

该 access_token 具有一定的时效性，以保证安全性。

后续请求做为请求参数，如使用 HTTP 协议的可以做为 HTTP 头携带，本文后续接口默认都认为已携带该参数，并推荐使用 HTTPS。

本接口定义参考标准 Oauth 2.0 规范[1]。

● 接口规范定义

```
swagger: "2.0"

info:

  description: "本接口描统一推送联盟建议的发送 PUSH API 定义"

  title: "Push Open API"

  version: "1.0.0"

basePath: "/v1/L1"
```

[1] https://tools.ietf.org/html/rfc6749

```
produces:
  - "application/json"
schemes:
- "https"
paths:
  /auth:
    post:
      tags:
        - "push open service"
      summary: "get access token"
      operationId: "pushMetadata"
      schemes: ["https"]
      produces:
        - "application/json"
      parameters:
        - name: ups
          in: body
          schema:
            type: object
            properties:
              app_id:
                type: string
              app_secret:
                type: string
              grant_type:
```

```
                type: string
                enum: [client_credentials]
              timestamp:
                type: string
            required:
              - grant_type
              - app_id
              - app_secret
              -  timestamp
      responses:
        200:
          description: "successful operation"
          schema:
            $ref: "#/definitions/auth_rsp"
        405:
          description: "permission denied"
        500:
          description: "internal server error"
        503:
          description: "flow control"



definitions:
  auth_rsp:
    type: object
```

```
    properties:
      error:
        type: string
      error_description:
        type: string
      access_token:
        type: string
      expires_in:
        type: integer
```

● 本接口定义参考标准 Oauth 2.0 规范请求参数

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

http 200 响应结果说明

● 响应结果

result	描述
-1	系统未知
0	成功
1	无效的 app_id
2	无效的 app_secret

>100	厂商预留错误码

● 响应数据

属性	类型	说明
result	string	鉴权结果
desc	string	结果描述
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。

4.1.2 推送(L1/ Send)

对 app 中单个用户，单独推送消息。

消息总长度不超过 4K 字节。

● 接口规范定义

```
swagger: "2.0"
info:
  description: "本接口描统一推送联盟建议的发送 PUSH API 定义"
  title: "Push Open API"
  version: "1.0.0"
basePath: "/v1/L1"
produces:
  - "application/json"
schemes:
- "https"
paths:
  /send:
    post:
      tags:
        - "push open service"
```

```
    summary: "sending push messages"
    operationId: "pushMetadata"
    schemes: ["https"]
    produces:
      - "application/json"
    parameters:
      - name: Authorization
        in: header
        type: string
        description: "取值样例 Bearer access_token"
      - name: Content-type
        in: header
        type: string
        description: "application/json"
      - name: ups
        in: body
        schema:
          type: object
          properties:
            registration_tokens:
              type: array
              items:
                type: string
              minItems: 1
              maxItems: 100
            notification:
              $ref: "#/definitions/Notification"
            notification_channel:
              type: string
            original_source_name:
              type: string
            original_source_ip:
```

```
                type: string
            ttl:
             type: string
            option:
              $ref: "#/definitions/Option"
          required:
            - registration_tokens
            - ttl
            - notification
            - original_source_name
            - original_source_ip


    responses:
      200:
        description: "successful operation"
        schema:
          $ref: "#/definitions/send_msg_rsp"
      405:
        description: "permission denied"
      500:
        description: "internal server error"
      503:
        description: "flow control"



definitions:
  Notification:
    type: object
    properties:
      title:
        type: string
      content:
```

```
      type: string
    click_action:
      $ref: "#/definitions/Action"
  required:
   - title
   - content
Option:
  type: object
  properties:
    key:
      type: string
    value:
      type: string
Action:
  type: object
  properties:
    url:
      type: string
    intent:
      type: string
send_msg_rsp:
  type: object
  properties:
    error:
      type: string
    error_description:
      type: string
    message_id:
        type: string
```

● 请求数据

HTTP 参数头

属性	类型	是否必传	说明
Authorization	string	是	申请的 access_token
Content-type	string	是	application/json

请求参数

属性	类型	父项目	是否必传	说明
registration_tokens	JSON Array		是	多个 registration_token 列表,建议最大值不超过 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14天）
option	JSONObject		否	消息扩展信息
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。）
notification_channel	string		否	为满足安卓O/P的特性，增加通知通道的特性（不超过64字节）
title	String	notifaication	是	通知标题（不超过128字节）
content	String	notifaication	是	通知内容（不超过256字节）
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
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

请求样例

```
Post /v1/L1/send Http 1.1

Host: xxxxx.com
```

```
Connection: Upgrade, HTTP2-Settings
Upgrade: h2c
Accept: application/json, text/plain, */*
Accept-encoding: gzip, deflate, br
Accept-language: zh-CN,zh;q=0.9
Authorization:asdadsddddddxxxxxxxxx
Content-type: application/json; charset=UTF-8
{
  "registration_tokens": [
    "1234567890123"
  ],
  "notification": {
    "title": "通知栏消息",
    "content": "这是个通知栏消息",
    "click_action": {
      "intent": "Test#TestIntent"
    }
  },
  "notification_channel": "test",
  "original_source_name": "huawei",
  "original_source_ip": "10.12.3.52",
  "ttl": "10000",
  "option": {
    "key1": "value1 ",
    "key2": "value2"
  }
}
```

● 响应数据

属性	类型	说明
result	int	推送结果
desc	string	结果描述

message_id	string	任务编号

httpcode 结果码说明

result	描述
200	请求成功
405	授权失败
500	系统内部错误
503	系统流控

http 200 响应结果说明

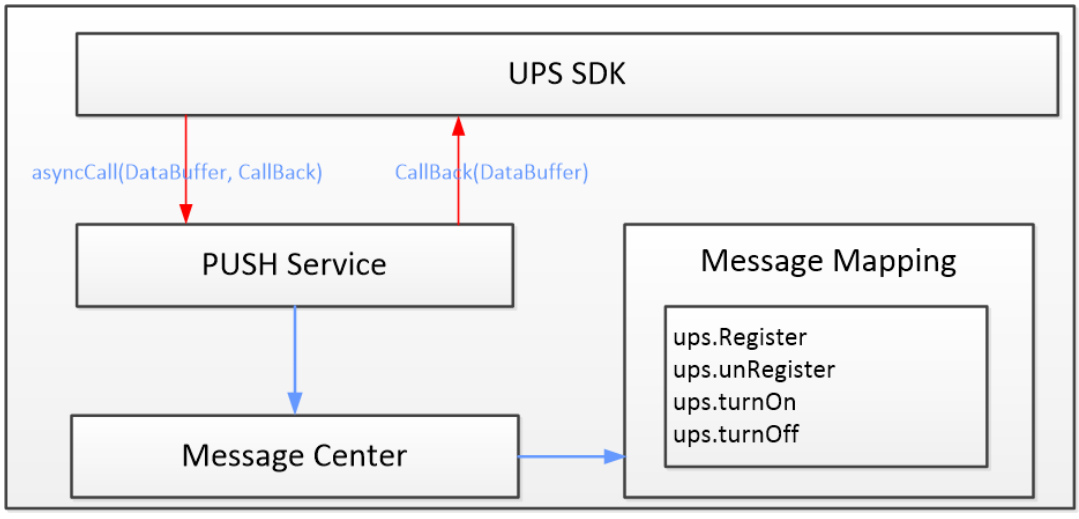
● 响应结果

result	描述
0	成功
>100	厂商自定义参数

5 移动端接口规范（L2）

5.1 总体方案

定义通用的请求，通过方法名进行实现扩展



5.2 申请 Token

public void registerToken (Context context, String appID, String appKey,String appSecret,UPSRegisterCallBack callback)

- **参数：**

字段	说明
context	应用的 Application Context
appID	在设备商开发者网站上创建应用时生成的唯一标识
appKey	在设备商开发者网站上创建应用时生成的 appKey，可以为 null
appSecret	在设备商开发者网站上创建应用时生成的应用客户端密钥，可以为 null
callback	接口回调，获取 token 接口

- **返回：**

无

- **说明：**

注册 UPS 服务，获取设备上应用的 token，作为后续的消息推送目标，结果异步回调返回，建议在 app 启动的时候调用。

- **UPS 实现设计：**

UPS SDK 里需要传输如下参数给厂商 Push：

字段	类型	说明
command	String	请求类型。 申请 token："ups.register"
pkgName	String	包名
appid	String	应用的 AppID
firstRequest	Boolean	是否首次调用，卸载重装，清空缓存都算是首次调用

```
public interface ICallbackResult<R> {

    void onResult(R result);

}
```

```
public interface UPSRegisterCallBack extends ICallbackResult<TokenResult> {

}
```

```
public void registerToken (context, new UPSRegisterCallBack(){
                        @Override
                        public void onResult(TokenResult result) {
                                result.getToken();
                        }
                });
```

首次调用时，在本地写一个标志位，标识后续的调用为非首次

5.3 注销 token

public void unRegisterToken (Context context, UPSUnRegisterCallBack callback)

● **参数：**

字段	类型	说明
context	Context	应用的 Application Context
callback	UPSUnRegisterCallBack	接口回调

● **返回：**

无

说明：

进行反注册

● **UPS 实现设计：**

UPS SDK 里需要传输如下参数给厂商 Push：

字段	类型	说明
command	String	请求类型。 申请 token："ups.unregister"
pkgName	String	包名
userId	int	用户标识

```
public interface UPSUnRegisterCallBack extends ICallbackResult<TokenResult> {

}
```

```
public void unRegisterToken (context, new UPSRegisterCallBack(){
```

```
                @Override
                public void onResult(TokenResult result) {
                        result.getReturnCode();
                }
        });
```

5.4 开启推送

public void turnOnPush(Context context, UPSTurnCallBack callback)

● **参数：**

字段	类型	说明
context	Context	应用的 Application Context
callback	UPSTurnCallBack	接口回调

● **返回：**

无

● **说明：**

开启 Push 推送，关闭状态则收不到推送。

如果已经调用了 turnOffPush 接口关闭了推送，则调用 turnOnPush 之后才能正常推送，或者应用重新卸载重装，再次调用 registerToken 时恢复推送。

● **UPS 实现设计：**

UPS SDK 里需要传输如下参数给厂商 Push：

字段	类型	说明
command	String	请求类型。 申请 token： "ups.turnOn"
pkgName	String	包名

```
public interface UPSTurnCallBack extends ICallbackResult<CodeResult> {

}
```

```
public void turnOnPush (context, new UPSTurnCallBack(){
                @Override
                public void onResult(CodeResult result) {
                        result.getReturnCode();
                }
        });
```

5.5 关闭推送

public void turnOffPush(Context context, UPSTurnCallBack callback)

● **参数：**

字段	类型	说明
context	Context	应用的 Application Context
Callback	UPSTurnCallBack	接口回调

● **返回：**

无

● **说明：**

关闭 Push 推送，关闭后则无法收到推送消息

如果已经调用了 turnOffPush 接口关闭了推送，则调用 turnOnPush 之后才能正常推送，或者应用重新卸载重装，再次调用 registerToken 时恢复推送。

● **UPS 实现设计：**

UPS SDK 里需要传输如下参数给厂商 Push：

字段	类型	说明
command	String	请求类型。 申请 token： "ups.turnOff"
pkgName	String	包名

```
public interface UPSTurnCallBack extends ICallbackResult<CodeResult> {

}
```

```
public void turnOffPush (context, new UPSTurnCallBack(){
                        @Override
                        public void onResult(CodeResult result) {
                                result.getReturnCode();
                        }
                });
```

5.6 开启/关闭推送的回调接口

public void onCommandResult (Context context, Bundle extras)

● 参数：

字段	类型	说明
context	Context	应用的 Application Context
extras	Bundle	返回接口调用结果，可扩展
	String	1.“command”
	String	2. ″resultCode″ 接口调用结果，如果成功，返回 ErrorCode.Sussess 即 0；否则返回错误类型值。
	String	3. “reason” 表示调用命令失败的原因。如果失败，则返回失败原因，否则返回为 null。

● 返回：

无

● 说明：

返回调用结果

● UPS 实现设计：

监听 com.ups.push.PUSH_RESPONSE 广播

附录 A:修订记录

日期	会议	修改章节	修改描述	修订版本
2017-11			初稿，搭建文档框架	0.0.1
2017-12		5	修改部分接口设计	0.1.0
2018-1		4-6	修改参数接口格式	0.2.0
2018-3		5、6	减少接口，只保留最基本功能	0.3.0
2018-4		5	删除不必要接口	0.8.0
2018-6		5、6	细化函数说明	0.9.0
2018-7		5、6	补充相关参数长度，调整全文格式	1.0.0
2019-5		4	1. 修改参数命名，根据各家厂商统一采用 xxx_xxx 命名方式 2. 授权接口返回值修改成 access_token 符合授权命名。 3. 发送接口，授权令牌在 http 头传入 4. 修改发送入参，通知栏消息发在一个结构体中，后续易于扩展 5. clickActionIntent 和 clickActionIntentParameterMap 参数去除使用 click_action 中 url 和 intent 承载。添加 option 扩展参数，便于各大厂商差异化参数传入	1.0.1
2019-5		5	1. 修改 registerToken 接口的入参，添加回调接口，token 直接在回调里返回，不使用广播返回 2. 修改 unRegisterToken 接口入参，添加回调接口参数，返回调用结果 3. 修改 turnOn 、turnoff 的接口入参，添加回调接口参数，返回调用结果	1.0.1
2019-5		5	1. 申请 token 时加入 appkey，appsecret 字段	1.0.1
2019-6		5	1. 补充接口回调参数	1.0.1
2019-6		无	1. 将 20190618 版本修改为 1.1.0 版本，以方便管理	1.1.0