QPlay API

Author: Steven Yan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document Revision History | | | | |
| Version | Date | Status | Author | Comments |
| 0.0.1 | 2016-3-28 |  | Steven Yan | Creation |
| 0.0.2 | 2016-4-28 |  | Steven Yan | 新增General API |
| 1.0.0 | 2016-5-20 |  | Steven Yan | 初稿API |
| 1.0.1 | 2016-05-25 |  | Steven Yan | 新增related tables |
| 1.0.2 | 2016-05-27 |  | Steven Yan | 補上錯誤碼處理建議 |
| 1.0.3 | 2016-05-30 |  | Steven Yan | 補上5.2, 5.3 register, login頁面的redirect\_uri資訊  調整API的順序 |
| 1.0.4 | 2016-05-31 |  | Steven Yan | 整合所有加密方式為sha256, 並新增範例  新增各種程式語言獲取unix時間搓記的方法  統一content-ype為application/json  login()底下新增security\_updated\_at回傳值, 通知APP端seciruty\_level最後異動時間 |
| 1.0.5 | 2016-06-17 |  | Steven Yan | 1)所有API的網址列中, lang參數為必填  2)加入Appendix-Jpush相關資訊  3)修改5.11, 5.13推送平台為Jpush |
| 1.0.6 | 2016-06-22 |  | Steven Yan | 5.2章節新增ldap\_bind的函式用法 |
| 1.0.7 | 2016-06-30 |  | Steven Yan | 5.3 Login部分, 更動security\_updated\_at這個欄位附近的結構, 因為security\_level的異動機率不高,所以一旦有異動,就把異動的app的package name也拋出來,但是只拋出兩個月內的異動清單  異動處如下的紅字  5.2 register API, request時, 網址列參數多一個device\_type |
| 1.0.8 | 2016-07-07 |  | Steven Yan | 第三章節新增999開頭的error code  999010:app-key參數錯誤  999011: signature參數錯誤或誤差超過15分鐘 |
| 1.0.9 | 2016-07-15 |  | Steven Yan | 1) 5.7 新增check point, 說明 qp\_session.last\_message\_time的更新時機  2) 新增000915錯誤碼, 代表package name不存在  3) 新增api—5.14 updateLastMessageTime |
| 1.1.0 | 2016-07-20 |  | Steven Yan | 修改UUID的生成方式, 補充在5.1章節 |
| 1.1.1 | 2016-07-27 |  | Steven Yan | 5.3 login() API的回傳值, 新增員工工號欄位 |
| 1.1.2 | 2016-08-01 |  | Steven Yan | 1. getMessageDetail 回應訊息刪除2個欄位，增加了source\_user 2. getAppList 回應信息中删除android\_url，IOS\_url,增加url 3. Register()和login()    redirect\_uri中可能有特殊字符，若放在URL中还需要编码，所以改为放到Request Header中 4. 5.13 定義$message\_title的最大長度為99, 如果有中文字的話為33, 所以qp\_message.message\_tile格式為varchar(99) 5. 5.7 getMessageList, 5.9 getMessageDetail這兩支PI新增message\_title回傳值 |
| 1.1.3 | 2016-08-03 |  | Steven Yan | 更新Register() API回傳值 |
| 1.1.4 | 2016-08-04 |  | Steven Yan | 1.  5.14, 當qp\_session表內以uuid查找找不到資訊或是token已經失效, 拋出錯誤訊息, 000908  2.  000915的錯誤訊息, 改為”package name不存在”  3.  將文件中, uuid不存在的狀況都改成000911 |
| 1.1.5 | 2016-08-08 |  | Steven Yan | 1.  5.7 getMessageList的回應訊息中多增加了一个message\_send\_row\_id  2.  5.13 "destination\_user\_id"這個節點底下新增"destination\_role\_id"這個大節點  3.  check point處新增許多解說, 請重新看過  4.  getMessageList新增source\_user回傳值, 跟getMessageDetail一樣 |
| 1.1.6 | 2016-08-09 |  | Steven Yan | updateMessage 和 getMessageDetail  message\_row\_id 改为message\_send\_row\_id  以便多次推播辨識用 |
| 1.1.7 | 2016-08-12 |  | Steven Yan | 修正send\_push\_message的API送出參數  destination\_user\_id  destination\_role\_id  兩節點為陣列方式 |
| 1.1.8 | 2016-08-16 |  | Steven Yan | getAppList裡面的multi\_lang這個Container下新增app\_code節點 |
| 1.1.9 | 2016-09-08 |  | Steven Yan | 更新5.13 sendPushMessage SPEC |
| 1.1.10 | 2016-09-08 |  | Steven Yan | 更新sendPushMessage的encode方式 |
| 1.1.11 | 2016-9-30 |  | Steven Yan | qplay的app-key改為appqplay  新增package name規範  com.qplay.[appKay]  且程式需檢查以下規則  [appkey] 必須檢查都是小寫 (目的android package name限制)  [appkey] 不能重複 (目的讓APP當成scheme)  [appkey] 一定要以app開頭, ex: apprrs (目的是讓QSecurity whitelist可以簡單過濾) |
| 1.1.12 | 2016-10-11 |  | Steven Yan | 補充send PushMessage API的說明:   1. sendPushMessage這支API, 不管送出的參數 need\_push是Y或是N, 都會同時往qp\_message, qp\_message\_send, qp\_role\_message/qp\_user\_message裡頭塞資料 2. qp\_user\_mesage, qp\_role\_message這兩張表用意是用來管控手機的用戶是否有權限可以查看此則消息   qp\_message\_send則是用來控制消息推播狀況, 兩者在意義上是完全可以分割的   1. qplay後台的推播管理,顯示歷史推播清單時, 只能顯示qp\_message\_send.need\_push=Y的資料 |
| 1.1.13 | 2016-10-27 |  | Steven Yan | 1. Register及Login 的response body, 新增emp\_no及domain, checksum這三個欄位 2. Register及Login 的response body, 回傳json, 新增loginid欄位 3. Login API¸如果token的有效期限在一小時以上, 則Server端不會更新Token的值   反之, 若token已經過期, 或是時效在一小時內, 則login api的response body會回傳新的token   1. 新增5.15 unregister API |
| 1.1.14 | 2016-11-1 |  | steven yan | 更新5.7 getMessageList, 新增一個參數overwrite\_timestamp |
| 1.1.15 | 2016-11-03 |  | Steven Yan | 1. 新增size節點在5.6 getAppList() API裡面 2. 5.6 getAppList新增check poit   抓取url這個欄位時, 需要判斷qp\_app\_version.external\_app=1, 如果是1, 則串url時不把prefix的字串串上去  qp\_app\_version.external\_app=0, 維持現有邏輯 |
| 1.1.16 | 2016-11-24 |  | Steven Yan | 修改5.6 getAppList, 將response body內app\_list節點下的app\_category改成app\_category\_id, 以便APP端比對 |
| 1.2.0 | 2016-12-05 |  | Steven Yan | 新增sendPushMessage API的處理方式, 直接串到JPUSH |
| 1.2.1 | 2016-12-12 |  | Steven Yan | 1.sendPushMessage, extra裡面的節點, message\_row\_id換成Parameter  2.Badge:”+1”改成Badge:”0”  3.updateMessage改成支援多筆一起處理  4. 處理一人多隻手機的問題,修改getMessageList, getMessageDetail,updateMessage, sendPushMessage, 推播後台  新增欄位, qp\_user\_message新增uuid欄位  [ALTER](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) [TABLE](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) `qp\_user\_message` ADD `uuid` [VARCHAR](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-types.html)(500) [CHARACTER](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/charset.html) [SET](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/charset.html) utf8 [COLLATE](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/charset-collations.html) utf8\_general\_ci [NOT](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL AFTER `message\_send\_row\_id`; |
| 1.2.2 | 2016-12-21 |  | Steven Yan | getMessageDetail, 新增000919錯誤碼, 如果是disable的狀態, 應該顯示000919 |
| 1.2.3 | 2017-01-06 |  | Steven Yan | 更新999012,999013,999014錯誤碼 |
| 1.2.4 | 2017-01-16 |  | Steven Yan | 更新5.13, 加入Tag, 定時推播功能 |
| 1.2.5 | 2017-02-07 |  | Steven Yan | 新增999015, 未上傳APP  checkAppVersion時會拋這個錯誤 |
| 1.2.6 | 2017-02-23 |  | StevenYan | 定義定時推播的name參數  一般定時推播定義name為:  "name":"send763"  秘書定時推播則定義name為  "name":"pushonly763" |
| 1.2.7 | 2017-03-03 |  | StevenYan | 新增ITportal串接app manager畫面的規格在5.16 |
| 1.2.8 | 2017-04-24 |  | StevenYan | 新增錯誤碼999016  調用customAPI的Url參數有誤 |
| 1.2.9 | 2017-04-25 |  | StevenYan | 修改5.13 sendPushMessage裡面的message\_title, 針對iOS推播的長度限制 |
| 1.2.10 | 2017-05-23 |  | StevenYan | 修改getAppList, 新增default\_lang節點在app\_list這個container下 |
| 1.2.11 | 2017-06-30 |  | StevenYan | 修改login, register兩支API, 新增site\_code節點  修改checkAppversion, download\_url需根據呼叫的API網址, 回應不同的download\_url, 隨著FQND變動 |
| 1.2.12 | 2017-07-11 |  | StevenYan | getMessageDetail API新增template\_id節點在content底下 |
| 1.2.13 | 2017-09-04 |  | StevenYan | 新增addDownloadHit API |
| 1.2.14 | 2017-09-20 |  | StevenYan | 加入sendPushMessage, message\_title, message\_text的加解密說明  加入template\_id說明 |
| 1.2.15 | 2017-09-27 |  | StevenYan | 加入5.18 checkAppVersionIntra |
| 1.2.16 | 2017-11-06 |  | StevenYan | 新增5.19 AddAppLog API |
| 1.2.17 | 2017-11-15 |  | StevenYan | 修改5.19 AddAppLog, 新增log\_list 這個container, 底下包一組以上的log  5.19 AddAppLog的post body,新增start\_time節點 |
| 1.2.18 | 2018-2-23 |  | StevenYan | 修改5.19 AddAppLog, 網址列的參數, UUID可以為空  修改5.13, sendPushMessage  URL新增qplay\_message\_list  qplay\_message\_list = Y，event\_type=news，need\_push=Y  會寫入 MySql: qp\_message,qp\_message\_send,  qplay\_message\_list = Y，event\_type=event，need\_push=Y 會寫入 MySql: qp\_message,qp\_message\_send,qp\_user\_message,qp\_role\_message  qplay\_message\_lsit = N，event\_type=news，need\_push=Y  會寫入 Mongo: qp\_message\_send\_pushonly,需紀錄appkey  qplay\_message\_list, = N，event\_type=event，need\_push=Y 會寫入 Mongo: qp\_message\_send\_pushonly,需紀錄appkey  post body新增Extra參數, 如果有填入, 會拿來取代推播時放入的extra節點, 在不同系統, 代表的意義不同, 不過都是key的定義, 用來收到推播後, 除了可以打開指定APP外, 還能開啟到特定頁面或是case, ex:ENS就傳入event id |

目錄

[.1 HTTP Header定義 13](#_Toc507163093)

[.2 HTTP Request URL & Response Body定義 14](#_Toc507163094)

[.3 回應代碼定義 15](#_Toc507163095)

[.4 範例: 透過ID查電話號碼 (QueryPhoneNoByID) 16](#_Toc507163096)

[.5 Qstor API 22](#_Toc507163097)

[5.1檢查移動裝置設備是否已經認證 (isRegister) 22](#_Toc507163098)

[5.2設備認證 (register) 29](#_Toc507163099)

[5.3 登入 (login) 39](#_Toc507163100)

[5.4 登出 (logout) 46](#_Toc507163101)

[5.5 確認APP是否需要更新版本 (checkAppVersion) 49](#_Toc507163102)

[5.6 取得App列表 (getAppList) 52](#_Toc507163103)

[5.7 取得訊息列表 (getMessageList) 58](#_Toc507163104)

[5.8 取得安全性資料列表 (getSecurityList) 64](#_Toc507163105)

[5.9 取得訊息詳細資訊 (getMessageDetail) 67](#_Toc507163106)

[5.10 更新已讀訊息(updateMessage) 72](#_Toc507163107)

[5.11 傳送推播token給Server (sendPushToken) 76](#_Toc507163108)

[5.12 刷新Token有效時間搓記(renewToken) 81](#_Toc507163109)

[5.13 傳送推播訊息給Server (sendPushMessage) 84](#_Toc507163110)

[5.14 刷新獲取信息的時間搓記(updateLastMessageTime) 116](#_Toc507163111)

[5.15設備移除 (Unregister) 120](#_Toc507163112)

[5.16 ITportal串接 (qplay web root directory) 123](#_Toc507163113)

[5.17新增下載次數 (addDownloadHit) 125](#_Toc507163114)

[5.18內網檢查APP版本號(checkAppVersionIntra) 129](#_Toc507163115)

[5.19新增APP行為Log (addAppLog) 133](#_Toc507163116)

[Appendix 139](#_Toc507163117)

[A1.Push API v3 141](#_Toc507163118)

[http://docs.jiguang.cn/server/rest\_api\_v3\_push/ 141](#_Toc507163119)

[推送概述 141](#_Toc507163120)

[推送校验 API 142](#_Toc507163121)

[推送对象 143](#_Toc507163122)

[调用返回 148](#_Toc507163123)

[HTTP状态码定义 149](#_Toc507163124)

[A2.JPush API client library for PHP 152](#_Toc507163125)

[A3. iOS SDK 集成指南 153](#_Toc507163126)

[A4. Android SDK 集成指南 154](#_Toc507163127)

[A5.Android API 154](#_Toc507163128)

[停止与恢复推送服务 API 154](#_Toc507163129)

[接收推送消息Receiver 154](#_Toc507163130)

[别名与标签 API 154](#_Toc507163131)

[获取 RegistrationID API 154](#_Toc507163132)

[A6. iOS API 154](#_Toc507163133)

[标签与别名 API (iOS) 154](#_Toc507163134)

[获取 APNs（通知） 推送内容 154](#_Toc507163135)

[获取自定义消息推送内容 154](#_Toc507163136)

[获取 RegistrationID 154](#_Toc507163137)

[设置Badge 154](#_Toc507163138)

[地理位置统计 154](#_Toc507163139)

## HTTP Header定義

所有的API request, 一定包含以下header四個參數：

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| Content-Type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| App-Key | 必填 | 專案名稱, 例如Qplay平台就是appqplay  此欄位也會用來當作app間識別的scheme |
| Signature-Time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼  ex: 2016/5/31 14:49:24換算為unix time為1464677364   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 如何在不同编程语言中获取现在的Unix时间戳(Unix timestamp)可以  參考以下程式碼   |  |  | | --- | --- | | 程式語言 | 獲取方式 | | Java | time | | JavaScript | Math.round(new Date().getTime()/1000)getTime()返回数值的单位是毫秒 | | Microsoft .NET / C# | epoch = (DateTime.Now.ToUniversalTime().Ticks - 621355968000000000) / 10000000 | | MySQL | SELECT unix\_timestamp(now()) | | Perl | time | | PHP | time() | | PostgreSQL | SELECT extract(epoch FROM now()) | | Python | 先 import time 然后 time.time() | | Ruby | 获取Unix时间戳：Time.now 或 Time.new 显示Unix时间戳：Time.now.to\_i | | SQL Server | SELECT DATEDIFF(s,'1970-01-01 00:00:00',GETUTCDATE()) | | Unix/Linux | date+%s | | VBScript/ASP | DateDiff("s","01/01/1970 00:00:00",Now()) | | 其他操作系统 (如果Perl被安装在系统中) | 命令行状态：perl -e "print time" | | |  | |
| Signature | 必填 | 透過不可逆的演算法加密:  Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  其中AppSecretKey就是yellowpage這個專案的key, 每個專案key是不一樣的  例如定義此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

Sha 256加密方式請參考<http://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages/>

參考上述sample code時請注意加密的參數順序, signature-time會在待加密的字串前面

HTTP回應訊息Header必定至少包含下面參數：

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| Content-Type | 必填 | 訊息體類型 application/json |
| Content-Length | 必填 | 應答訊息長度, 當應答訊息包含訊息體時需要指定訊息體長度 |

## HTTP Request URL & Response Body定義

HTTP Request 時, URL固定包含的參數如下,

URL為 GET /v101/yellowpage/{action}?lang=en-us&……

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| version | Required | API版本,  ex. v 1.0.1表達為101, 以此類推 |
| name | Required | 專案名稱, 每一個APP有專有的name  Ex: qplay, eis, cpcr, amp, rt, gcgs, yellowpage, rrs  此專案名稱為yellowpage |
| action | Required | 調用的Application名稱，ex. QueryPhoneNoByID |
| lang | Required | 5碼, ex. en-us, 如傳空值, 視同於en-us   (保留將來多語系可以使用) |

HTTP Response Body固定包含的參數如下

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| result\_code | Required | 回應碼 (定義為6碼) |
| message | Required | 回應訊息 |
| content | Required | 回應內容(PM自行定義) |

## 回應代碼定義

通用回應代碼如下所示：前三碼999代表所有專案都適用, 001-998則分配給特定的專案使用

專案清單: \\datastore.benq.corp.com\BI30\Qplay\5.Management\Projects.xlsx

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 成功 |
| 999001 | 傳入參數不足或傳入參數格式錯誤 |
| 999002 | version參數錯誤 or 不存在 |
| 999003 | action參數錯誤 or 不存在 |
| 999004 | lang參數錯誤 or 不存在 |
| 999005 | 傳送方法錯誤,  ex.Method應為GET時, 卻傳POST |
| 999006 | Content-Type錯誤  (當Client的傳送方法為POST時, 才會有此result code產生) |
| 999007 | 傳入的json格式錯誤, Server端無法解析  (當Client的傳送方法為POST時, 才會有此result code產生) |
| 999008 | Signature驗證碼不正確 |
| 999009 | 禁止存取API (IP為黑名單) |
| 999010 | app-key參數錯誤 |
| 999011 | signature參數錯誤或誤差超過15分鐘 |
| 999012 | app已經下架 |
| 999013 | push token已經使用 |
| 999014 | company不存在 |
| 999015 | 未上傳app |
| 999016 | 調用customAPI的Url參數有誤 |
| 999999 | 其他未知錯誤 |

針對yellowpage專案所會使用的回應代碼可以再特別規範

這份清單就需要各PM自行提供

例如前三碼為專案代號, 假設yellowpage的專案代號為001：

第四碼代表訊息類別, 其中9代表可以顯示給前台查看

|  |  |  |  |
| --- | --- | --- | --- |
| **Result Code** | **Description - TW** | **Description - EN** | **Description - CN** |
| 1 | 成功 | OK | 成功 |
| 001901 | 查無此人 | Cannot find the result | 查无此人 |

以下我簡單寫一個電話查詢API的實作方法, (一樣紅字是客製)

但實際上, 這些電話簿資料應該是在e04系統裡面, 實質上我們的平台只是提供API規範

這個API的開發應該還是要下放到每個系統的PM

PM開發時, 必須遵循規則開發完畢, 放到Qplay平台上

將來撰寫客製系統API SPEC時可以參考以下寫法

## 範例: 透過ID查電話號碼 (QueryPhoneNoByID)

#### 接口描述

APP端透過API拿到電話號碼

#### 接口訊息定義

請求方法：

GET /v101/yellowpage/QueryPhoneNoByID?lang=en-us&ad\_id=steven.yan

**請求訊息：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| ad\_id | Required | string | AD的帳號 |

**回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | Integer | 回應碼 |
| message | NA | 1 | String | 回應訊息 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

**回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 001901 | 查無此人 |

EX:

Request **Url**：http://mproject\_api.benq.com/v101/yellowpage/QueryPhoneNoByID?lang=en-us&ad\_id=steven.yan  
Request **Header**：Array( [0]=> Content-Type:application/json[1]=> App-Key:yellowpage[2]=> Signature-Time:1458900578 [3]=> Signature: WsnMjPaCnVTJmUk0tIkeT9bEIng= )

Response **Data**HTTP/1.1 200 OK  
Date: Fri, 25 Mar 2016 10:22:18 GMT  
Server: Apache/2.2.3 (CentOS)  
X-Powered-By: PHP/5.2.10  
Content-Length: 116  
Content-Type: application/json ;   
Response **Body**:  
{

    "result\_code": "1",

    "message": "OK",

"token\_valid": "1458884996",

    "content": {

      "ad\_id": "steven.yan",

      "phone\_number": "7758"

    }

}

**其中紅色框起來的content部分就給PM自行定義**

**平台只提供格式確認**

**Content裡面再包的巢狀JSON包什麼就不管了**

HTTP請求訊息Header包含下面參數：

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| Content-Type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| App-Key | 必填 | 專案名稱, 例如Qplay平台就是appqplay  此欄位也會用來當作app間識別的scheme |
| Signature-Time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼  ex: 2016/5/31 14:49:24換算為unix time為1464677364   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 如何在不同编程语言中获取现在的Unix时间戳(Unix timestamp)可以  參考以下程式碼   |  |  | | --- | --- | | 程式語言 | 獲取方式 | | Java | time | | JavaScript | Math.round(new Date().getTime()/1000)getTime()返回数值的单位是毫秒 | | Microsoft .NET / C# | epoch = (DateTime.Now.ToUniversalTime().Ticks - 621355968000000000) / 10000000 | | MySQL | SELECT unix\_timestamp(now()) | | Perl | time | | PHP | time() | | PostgreSQL | SELECT extract(epoch FROM now()) | | Python | 先 import time 然后 time.time() | | Ruby | 获取Unix时间戳：Time.now 或 Time.new 显示Unix时间戳：Time.now.to\_i | | SQL Server | SELECT DATEDIFF(s,'1970-01-01 00:00:00',GETUTCDATE()) | | Unix/Linux | date+%s | | VBScript/ASP | DateDiff("s","01/01/1970 00:00:00",Now()) | | 其他操作系统 (如果Perl被安装在系统中) | 命令行状态：perl -e "print time" | | |  | |
| Signature | 必填 | 透過不可逆的演算法加密:  Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  其中AppSecretKey就是yellowpage這個專案的key, 每個專案key是不一樣的  例如定義此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

其中, Signature-Time是unix時間搓

2016年3月25日 下午 01:49:56 換算為1458884996

Signature則用以下的encode方式

Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )

其中appSecretKey是固定的, 每個APP有自己的32bits的secretkey

每個APP專案申請的時候, 由Qplay平台管理人員生成後配發

SignatureTime則是隨時間變化

EX:

Base64(HMAC-SHA256 (1458884996, swexuc453refebraXecujeruBraqAc4e))

會得到GUZkDX9UdbF3L28ZeqLiJ6z1/C/QHWJRI2WDRdlEo64=

Sha 256加密方式請參考<http://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages/>

參考上述sample code時請注意加密的參數順序, signature-time會在待加密的字串前面

如果要測試的話

可以到提供的測試網站

<http://vodtest.benq.com/examples/Signature_sha256.php>

完成三步驟就可以拿到signature去測試RD寫好的演算法



## Qstor API

### 5.1檢查移動裝置設備是否已經認證 (isRegister)

#### 5.1.1接口描述

移動裝置提供uuid給server確認是否已經認證過, 原則上一個移動裝置只需認證一次。

#### 5.1.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/isRegister?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| is\_register | content | 0-1 | Boolean | 1:true, 已經註冊  0:false, 未註冊 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |

**Related Tables:**

qp\_register

qp\_error\_code

**Check point:**

1. UUID生成方式, 分別以Android及iOS說明

**ANDROID**

可以選用的ID大概有幾下5種: **DEVICE\_ID, MAC\_Address, Serial Number, ANDROID\_ID, Installtion ID : UUID**

每一種都各有優缺點, 但我們有盡可能保留原ID的需求

所以還是以ANDROID\_ID為主

萬一是特別的設備取不到ANDROID\_ID, 則用TelephonyManager.getDeviceId()拿DEVICE\_ID, 如若還拿不到的話, 只好自行產生UUID

因此取得順序: ANDROID\_ID->DEVICE\_ID->UUID

以上拿到的資料皆必須保存到SharedPreferences

參考資料來自:

<http://www.cnblogs.com/lqminn/p/4204855.html>

參考代碼:

1 import android.content.Context;

2 import android.content.SharedPreferences;

3 import android.provider.Settings.Secure;

4 import android.telephony.TelephonyManager;

5

6 import java.io.UnsupportedEncodingException;

7 import java.util.UUID;

8

 9 public class DeviceUuidFactory {

10     protected static final String PREFS\_FILE = "device\_id.xml";

11     protected static final String PREFS\_DEVICE\_ID = "device\_id";

12

13     protected static UUID uuid;

14

15     /\*\*

16     \* Returns a unique UUID for the current android device. As with all UUIDs, this unique ID is "very highly likely"

17     \* to be unique across all Android devices. Much more so than ANDROID\_ID is.

18     \*

19     \* The UUID is generated by using ANDROID\_ID as the base key if appropriate, falling back on

20     \* TelephonyManager.getDeviceID() if ANDROID\_ID is known to be incorrect, and finally falling back

21     \* on a random UUID that's persisted to SharedPreferences if getDeviceID() does not return a

22     \* usable value.

23     \*

24     \* In some rare circumstances, this ID may change. In particular, if the device is factory reset a new device ID

25     \* may be generated. In addition, if a user upgrades their phone from certain buggy implementations of Android 2.2

26     \* to a newer, non-buggy version of Android, the device ID may change. Or, if a user uninstalls your app on

27     \* a device that has neither a proper Android ID nor a Device ID, this ID may change on reinstallation.

28     \*

29     \* Note that if the code falls back on using TelephonyManager.getDeviceId(), the resulting ID will NOT

30     \* change after a factory reset. Something to be aware of.

31     \*

32     \* Works around a bug in Android 2.2 for many devices when using ANDROID\_ID directly.

33     \*

34     \* @see <http://code.google.com/p/android/issues/detail?id=10603>

35     \*

36     \* @return a UUID that may be used to uniquely identify your device for most purposes.

37     \*/

38     public DeviceUuidFactory(Context context) {

39         if( uuid ==null ) {

40             synchronized (DeviceUuidFactory.class) {

41                 if( uuid == null) {

42                     final SharedPreferences prefs = context.getSharedPreferences(PREFS\_FILE, 0);

43                     final String id = prefs.getString(PREFS\_DEVICE\_ID, null );

44

45                     if (id != null) {

46                         // Use the ids previously computed and stored in the prefs file

47                         uuid = id;

48                     } else {

49                         final String androidId = Secure.getString(context.getContentResolver(), Secure.ANDROID\_ID);

50

51                         // Use the Android ID unless it's broken, in which case fallback on deviceId,

52                         // unless it's not available, then fallback on a random number which we store

53                         // to a prefs file

54                         try {

55                             if (!"9774d56d682e549c".equals(androidId)) {

56                                 uuid = UUID.nameUUIDFromBytes(androidId.getBytes("utf8"));

57                             } else {

58                                 final String deviceId = ((TelephonyManager) context.getSystemService( Context.TELEPHONY\_SERVICE )).getDeviceId();

59                                 uuid = deviceId != null ? UUID.nameUUIDFromBytes(deviceId.getBytes("utf8")) : UUID.randomUUID();

60                             }

61                         } catch (UnsupportedEncodingException e) {

62                             throw new RuntimeException(e);

63                         }

64

65                         // Write the value out to the prefs file

66                         prefs.edit().putString(PREFS\_DEVICE\_ID, uuid.toString()).commit();

67

68                     }

69                 }

70             }

71         }

72     }

73 }

**iOS**

以向APNS取得的DeviceToken為UUID

**所以呼叫Qplay的Register頁面時, 就需要先取得DeviceToken了**

也就是安裝Qplay app後, 要馬上跟APNS取得DeviceToken, 拿來向Qplay註冊UUID

然後同時還必須向Jpush(極光)註冊取得RegisterID, 拿來推播用

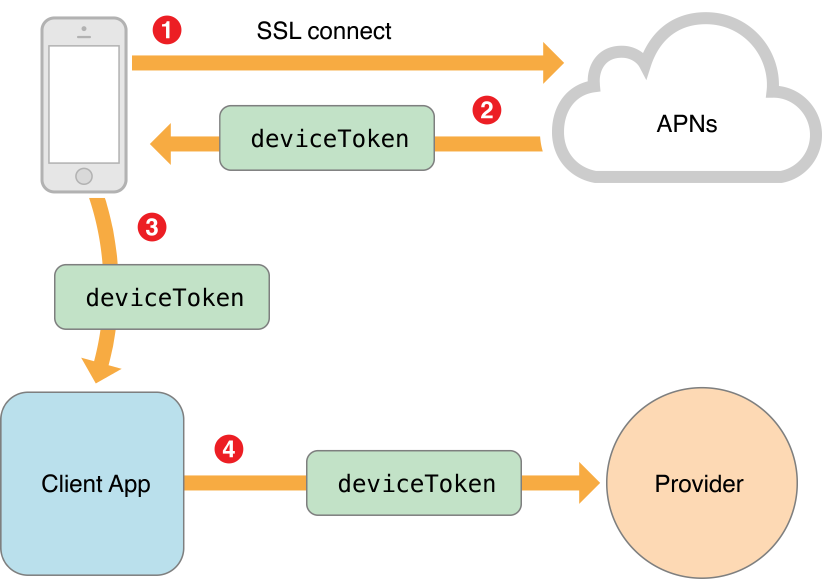
接著註冊時送出來的是APNS註冊的DeviceToken

呼叫sendPushToken() API時, 再送出Jpush的RegisterID

以下是APNS取得DeviceToken的方式

第一步第二步拿到DeviceToken後

第四步驟, 呼叫Register() API時, 就把deviceToken當UUID拿來註冊



參考代碼:

// Device Token取得

    func application( application: UIApplication!, didRegisterForRemoteNotificationsWithDeviceToken deviceToken: NSData! ) {

        var characterSet: NSCharacterSet = NSCharacterSet( charactersInString: "<>" )

        println(characterSet)

        var deviceTokenString: String = ( deviceToken.description as NSString )

.stringByTrimmingCharactersInSet( characterSet )

            .stringByReplacingOccurrencesOfString( " ", withString: "" ) as String

         //Print出Device Token

        println( deviceTokenString )

    }

    //向遠端APNS註冊失敗的事件

    func application( application: UIApplication!, didFailToRegisterForRemoteNotificationsWithError error: NSError! ) {

        println( error.localizedDescription )

    }

### 5.2設備認證 (register)

#### 5.2.1接口描述

透過此API進行設備認證, 將uuid註冊到Qplay Server, 認證時需比對AD帳號密碼是否正確

目前設備認證暫時不做email認證, 也不做近端同網域認證

只要帳密正確就可以認證新的設備

#### 5.2.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/register?lang=en-us
3. header請求參數:

Register需要將帳號密碼欄位透過HTTP Header送出, 所以包含下面參數：

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| domain | 必填 | AD domain |
| loginid | 必填 | AD login id |
| password | 必填 | AD login password |
| redirect\_uri | 必填 | app提供的轉址URI, 待qplay auth web頁面呼叫register api取回token後, 需要轉址回這個頁面 |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| ~~redirect\_uri~~ | ~~Required~~ | ~~string~~ | ~~app提供的轉址URI, 待qplay auth web頁面呼叫register api取回token後, 需要轉址回這個頁面~~ |
| device\_type | Required | string | ios/android |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| redirect\_uri | content | 1 | String | app提供的轉址URI, 待qplay auth web頁面呼叫register api取回token後, 需要轉址回這個頁面  若取得token失敗, 也會透過這個URI轉址將錯誤訊息帶回給app |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | content | 0-1 | String | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| token | content | 0-1 | String | 由server配發的token, 用來識別是否已經登入qplay, 時效為48hour |
| loginid | content | 0-1 | String | 員工AD帳號  ex:steven.yan |
| emp\_no | content | 0-1 | String | 員工工號 |
| domain | content | 0-1 | String | 員工所屬domain  ex:Qgroup, BenQ |
| site\_code | content | 0-1 | String | 員工所屬地區  ex:QTY,QTT,QCS |
| checksum | content | 0-1 | String | 將密碼欄位用md5加密後當作checksum |
| security\_update\_list | NA | 0-1 | Container | 回應訊息內容Container |
| app\_key | security\_update\_list | 0-N | String | 兩個月內有異動security\_level的app |
| security\_updated\_at | security\_update\_list | 0-N | Integer | 這個欄位表示security level的最後更新時間, 採用Unix時間搓記  ex: 2016年3月25日 下午 01:49:56 換算為1458884996 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000901 | 帳號不存在 |
| 000902 | 密碼錯誤 |
| 000903 | 設備已被認證, uuid重複 |
| 000914 | 帳號已被停權 |

**000901:離職或是帳號資訊打錯**

**000914:在職但是被停權**

**Related Tables:**

qp\_register

qp\_user

qp\_push\_token

qp\_project

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000901 | 帳號不存在 | 提示錯誤, 點擊確認鍵後,回到認證畫面並清空密碼欄位 |
| 000902 | 密碼錯誤 | 提示錯誤, 點擊確認鍵後,回到認證畫面並清空密碼欄位 |
| 000903 | 設備已被認證, uuid重複 | 停留在認證畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號問題 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

Check Point:

1. Qplay提供認證網頁, client端開啟時必須透過在url上輸入參數的方式提供uuid, 同時, user在web畫面上輸入domain, 帳號密碼, web會透過http get方式將domain, 帳密置於header區塊, 一併送出給qplay server
2. 以下說明client跟此認證網頁互動的方式供app調用

* Client端與Auth web及Qplay平台互動流程

|  |  |  |
| --- | --- | --- |
| APP端 | Auth Web Page端 | Qplay Server端 |
| 1.調用QPlay Auth web頁面, 並帶入app\_key, uuid及redirect的URI |  |  |
|  | 2. User在Qplay Auth頁面輸入domain, user id, password |  |
|  | 3. Qplay auth頁面將, uuid, redirect\_uri帶至參數列, domain, user\_id, password放在header送出register api request |  |
|  |  | 4.處理register api並回傳response:  成功:回傳token及redirect\_uri  失敗:回傳錯誤訊息及redirect\_uri |
|  | 5. 收到response後, 將資訊掛在URL參數列並轉址到redirect\_uri上的網址 |  |
| 6. 取得token或是失敗訊息 |  |  |

* 調用qplay auth頁面說明

|  |  |
| --- | --- |
| 調用網址 | ex:https://qplay.benq.com/qplayauth\_register?app\_key=qplay&uuid={uuid}&redirect\_uri={redirect\_uri} |
| 網址列輸入參數 | |
| app\_key | 統一填入qplay  將來有其他平台才需填入其他APP\_key |
| uuid | 填入app的unique id, 也就是uuid, 每一個device有唯一值 |
| redirect\_uri | 用來收回傳值的URI, app透過該網址收回token或是錯誤訊息 |
| 輸出值 | |
| 認證成功 | {redirect\_uri}?token={token}&token\_valid={token\_valid} |
| 認證失敗 | {redirect\_uri}?result\_code={result\_code}&message={message} |

需討論是否認證錯誤就不導回app畫面, 只停留在auth web畫面

或是有條件導回

Ex:只有在00901跟00902以外的error code才導回

1. 該網頁與AD溝通時採用LDAP API, 可以參考以下資訊

<http://php.net/manual/en/book.ldap.php>

連線的LDAP網址:

LDAP://BenQ.corp.com

**5.3的登入畫面, 也採用同樣方式, 透過LDAP API檢查AD帳密**

1. 透過Ldap\_bind的函式就可以知道帳密是否正確

參考網址

<http://icodding.blogspot.tw/2015/10/php-php-ad-server.html>

<http://bbs.mychat.to/reads.php?tid=137319>

<http://ithelp.ithome.com.tw/questions/10167400>

1. Sample code1

<?php

/\*\*

\* Created by Joe of ExchangeCore.com

\*/

if(isset($\_POST['username']) && isset($\_POST['password'])){

$adServer = "ldap://domaincontroller.mydomain.com";

$ldap = ldap\_connect($adServer);

$username = $\_POST['username'];

$password = $\_POST['password'];

$ldaprdn = 'mydomain' . "\\" . $username;

ldap\_set\_option($ldap, LDAP\_OPT\_PROTOCOL\_VERSION, 3);

ldap\_set\_option($ldap, LDAP\_OPT\_REFERRALS, 0);

$bind = @ldap\_bind($ldap, $ldaprdn, $password);

if ($bind) {

$filter="(sAMAccountName=$username)";

$result = ldap\_search($ldap,"dc=MYDOMAIN,dc=COM",$filter);

ldap\_sort($ldap,$result,"sn");

$info = ldap\_get\_entries($ldap, $result);

for ($i=0; $i<$info["count"]; $i++)

{

if($info['count'] > 1)

break;

echo "<p>You are accessing <strong> ". $info[$i]["sn"][0] .", " . $info[$i]["givenname"][0] ."</strong><br /> (" . $info[$i]["samaccountname"][0] .")</p>\n";

echo '<pre>';

var\_dump($info);

echo '</pre>';

$userDn = $info[$i]["distinguishedname"][0];

}

@ldap\_close($ldap);

} else {

$msg = "Invalid email address / password";

echo $msg;

}

}else{

?>

<form action="#" method="POST">

<label for="username">Username: </label><input id="username" type="text" name="username" />

<label for="password">Password: </label><input id="password" type="password" name="password" /> <input type="submit" name="submit" value="Submit" />

</form>

<?php } ?>

1. Sample code2

Retrieve information for all entries where the surname starts with "S" from a directory server, displaying an extract with name and email address.

Example #1 LDAP search example

<?php  
// basic sequence with LDAP is connect, bind, search, interpret search  
// result, close connection  
  
echo "<h3>LDAP query test</h3>";  
echo "Connecting ...";  
$ds=ldap\_connect("LDAP://BenQ.corp.com");  // must be a valid LDAP server!  
echo "connect result is " . $ds . "<br />";  
  
if ($ds) {   
    echo "Binding ...";   
    $r=ldap\_bind($ds);     // this is an "anonymous" bind, typically  
                           // read-only access  
    echo "Bind result is " . $r . "<br />";  
  
    echo "Searching for (sn=S\*) ...";  
    // Search surname entry  
    $sr=ldap\_search($ds, "o=My Company, c=US", "sn=S\*");    
    echo "Search result is " . $sr . "<br />";  
  
    echo "Number of entries returned is " . ldap\_count\_entries($ds, $sr) . "<br />";  
  
    echo "Getting entries ...<p>";  
    $info = ldap\_get\_entries($ds, $sr);  
    echo "Data for " . $info["count"] . " items returned:<p>";  
  
    for ($i=0; $i<$info["count"]; $i++) {  
        echo "dn is: " . $info[$i]["dn"] . "<br />";  
        echo "first cn entry is: " . $info[$i]["cn"][0] . "<br />";  
        echo "first email entry is: " . $info[$i]["mail"][0] . "<br /><hr />";  
    }  
  
    echo "Closing connection";  
    ldap\_close($ds);  
  
} else {  
    echo "<h4>Unable to connect to LDAP server</h4>";  
}  
?>

1. Register及Login 的response body, 新增loginid, emp\_no及domain, checksum這4個欄位

loginid, emp\_no跟domain請從qp\_user表抓取

其中checksum是為了讓Security level較高的app

從背景resume時, 不再開啟Server提供的login page

直接在Local手機端驗證帳密即可

因此, 在規則上直接用md5運算即可:

      只要在Register或是Login頁面輸入帳密, 經過LDAP驗證通過後,

      就把密碼欄位做md5加密後放在response body回傳給Client端

      loginid跟domain就不放入運算

### 5.3 登入 (login)

#### 5.3.1接口描述

登入qplay, 取得token

#### 5.3.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/login?lang=en-us
3. header請求參數:

login需要將帳號密碼欄位透過HTTP Header送出, 所以包含下面參數：

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| domain | 必填 | AD domain |
| loginid | 必填 | AD login id |
| password | 必填 | AD login password |
| redirect\_uri | 必填 | app提供的轉址URI, 待qplay auth web頁面呼叫login api取回token後, 需要轉址回這個頁面 |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| ~~redirect\_uri~~ | ~~Required~~ | ~~string~~ | ~~app提供的轉址URI, 待qplay auth web頁面呼叫login api取回token後, 需要轉址回這個頁面~~ |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| redirect\_uri | content | 1 | String | app提供的轉址URI, 待qplay auth web頁面呼叫login api取回token後, 需要轉址回這個頁面  若取得token失敗, 也會透過這個URI轉址將錯誤訊息帶回給app |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | content | 0-1 | String | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| token | content | 0-1 | String | 由server配發的token, 用來識別是否已經登入qplay, 時效為48hours |
| loginid | content | 0-1 | String | 員工AD帳號  ex:steven.yan |
| emp\_no | content | 0-1 | String | 員工工號 |
| domain | content | 0-1 | String | 員工所屬domain  ex:Qgroup, BenQ |
| site\_code | content | 0-1 | String | 員工所屬地區  ex:QTY,QTT,QCS |
| checksum | content | 0-1 | String | 將密碼欄位用md5加密後當作checksum |
| security\_update\_list | NA | 0-1 | Container | 回應訊息內容Container |
| app\_key | security\_update\_list | 0-N | String | 兩個月內有異動security\_level的app |
| security\_updated\_at | security\_update\_list | 0-N | Integer | 這個欄位表示security level的最後更新時間, 採用Unix時間搓記  ex: 2016年3月25日 下午 01:49:56 換算為1458884996 |

**備註1:**

因為security\_level的異動機率不高

所以一旦有異動

就把異動的app的package name也拋出來

但是只拋出兩個月內的異動清單

異動處如下的紅字

**備註2:**

如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000901 | 帳號不存在 |
| 000902 | 密碼錯誤 |
| 000904 | 登入帳號與設備認證時的帳號不同 |
| 000905 | 設備未認證 |
| 000906 | token取得失敗 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_session

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000901 | 帳號不存在 | 提示錯誤, 點擊確認鍵後, 回到登入畫面並清空密碼欄位 |
| 000902 | 密碼錯誤 | 提示錯誤, 點擊確認鍵後, 回到登入畫面並清空密碼欄位 |
| 000904 | 登入帳號與設備認證時的帳號不同 | 停留在登入畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號問題 |
| 000905 | 設備未認證 | 直接走register() API |
| 000906 | token取得失敗 | 正常情況下不會顯示00906這個訊息, 若出現, 請顯示錯誤訊息, 並加上請聯絡系統管理員確認Token問題 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

Check Point:

1. Qplay提供登入網頁, client端開啟時必須透過在url上輸入參數的方式提供uuid, 同時, user在web畫面上輸入domain, 帳號密碼, web會透過http get方式將domain, 帳密置於header區塊, 一併送出給qplay server
2. 以下說明client跟此登入網頁互動的方式供app調用

* Client端與Auth web及Qplay平台互動流程

|  |  |  |
| --- | --- | --- |
| APP端 | Auth Web Page端 | Qplay Server端 |
| 1.調用QPlay Auth web頁面, 並帶入app\_key, uuid及redirect的URI |  |  |
|  | 2. User在Qplay Auth頁面輸入domain, user id, password |  |
|  | 3. Qplay auth頁面將, uuid, redirect\_uri帶至參數列, domain, user\_id, password放在header送出login api request |  |
|  |  | 4.處理login api並回傳response:  成功:回傳token及redirect\_uri  失敗:回傳錯誤訊息及redirect\_uri |
|  | 5. 收到response後, 將資訊掛在URL參數列並轉址到redirect\_uri上的網址 |  |
| 6. 取得token或是失敗訊息 |  |  |

* 調用qplay auth頁面說明

|  |  |
| --- | --- |
| 調用網址 | ex:https://qplay.benq.com/qplayauth\_login?app\_key=qplay&uuid={uuid}&redirect\_uri={redirect\_uri} |
| 網址列輸入參數 | |
| app\_key | 統一填入qplay  將來有其他平台才需填入其他APP\_key |
| uuid | 填入app的unique id, 也就是uuid, 每一個device有唯一值 |
| redirect\_uri | 用來收回傳值的URI, app透過該網址收回token或是錯誤訊息 |
| 輸出值 | |
| 認證成功 | {redirect\_uri}?token={token}&token\_valid={token\_valid} |
| 認證失敗 | {redirect\_uri}?result\_code={result\_code}&message={message} |

需討論是否認證錯誤就不導回app畫面, 只停留在auth web畫面

或是有條件導回

Ex:只有在00901跟00902以外的error code才導回

1. Register及Login 的response body, 新增loginid, emp\_no及domain, checksum這4個欄位

loginid, emp\_no跟domain請從qp\_user表抓取

其中checksum是為了讓Security level較高的app

從背景resume時, 不再開啟Server提供的login page

直接在Local手機端驗證帳密即可

因此, 在規則上直接用md5運算即可:

      只要在Register或是Login頁面輸入帳密, 經過LDAP驗證通過後,

      就把密碼欄位做md5加密後放在response body回傳給Client端

      loginid跟domain就不放入運算

1. Login API¸如果token的有效期限在一小時以上, 則Server端不會更新Token的值

反之, 若token已經過期, 或是時效在一小時內, 則login api的response body會回傳新的token

      這個修改比較單純

      只要檢查token\_valid欄位在一小時內的才刷新Token並回傳

      否則正常情況下, 就直接到qp\_session表抓舊有的token回傳給Client

### 5.4 登出 (logout)

#### 5.4.1接口描述

登出qplay, 刪除server端token

#### 5.4.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/logout?lang=en-us
3. header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| domain | Required | string | user的domain |
| loginid | Required | string | user的AD帳號id |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| ~~token\_valid~~ | ~~NA~~ | ~~0-1~~ | ~~Integer~~ | ~~登出API不會有此element~~ |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | content | 0-1 | String | 設備唯一識別碼 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000901 | 帳號不存在 |
| 000904 | 登入帳號與設備認證時的帳號不同 |
| 000905 | 設備未認證 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_session

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000901 | 帳號不存在 | 仍然登出, 差別只是server端有紀錄log |
| 000904 | 登入帳號與設備認證時的帳號不同 | 仍然登出, 差別只是server端有紀錄log |
| 000905 | 設備未認證 | 仍然登出, 差別只是server端有紀錄log |
| 000914 | 帳號已被停權 | 仍然登出, 差別只是server端有紀錄log |

Check Point:

1. 登出沒有web頁面, 直接由client呼叫api完成
2. 登出後就把這個帳號底下device底下的token清空
3. 需討論帳號若登出了是否還要進行推播

### 5.5 確認APP是否需要更新版本 (checkAppVersion)

#### 5.5.1接口描述

取得APP最新版號資訊, 並由Server主動確認是否需要更新

#### 5.5.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/checkAppVersion?lang=en-us&package\_name=com.qplay.appqplay&device\_type=android&version\_code=1

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| package\_name | Required | string | app的package name |
| device\_type | Required | string | ios  android  其中一個 |
| vension\_code | Required | string | 填入目前版號 |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼  resul\_code=1, 表示需要更新  result\_code =000913, 表示不需要更新 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| version\_code | content | 0-1 | String | 新版本版號 |
| download\_url | content | 0-1 | String | 直接提供iOS或是Android的下載路徑  下載的路經需偵測詢問的路徑, 如果是qplay.benq.com/…..  則回應的下載網址為<https://qplay.benq.com/>....  如果下的API路徑為10.82.121.4/…則回應的下載網址為<https://10.82.121.4/>.... |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000913 | 不需更新 |

**Related Tables:**

qp\_app\_head

qp\_app\_version

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000913 | 不需更新 | 收到此訊息app不需另外處理, 代表APP是新版本, 直接進行下一個步驟即可 |

### 5.6 取得App列表 (getAppList)

#### 5.6.1接口描述

取得有權限的App清單及Detail資訊

#### 5.6.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/getAppList?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 qplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  token值需要使用sha256加密, 加密時需要添加salt值, salt值為request header內的Signature-Time參數  $token = base64\_encode (hash\_hmac(‘sha256’, $Signature-Time, $token’, true)); |

Sha 256加密方式請參考<http://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages/>

參考上述sample code時請注意加密的參數順序, signature-time會在待加密的字串前面

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| app\_category\_list | content | 0-N | Container | app分類的Container |
| category\_id | app\_category\_list | 0-1 | String | app分類的Unique流水號  qp\_app\_category.row\_id |
| app\_category | app\_category\_list | 0-1 | String | app分類名稱 |
| sequence | app\_category\_list | 0-1 | Integer | app分類排序 |
| app\_list | content | 0-N | Container | app清單的Container |
| app\_id | app\_list | 0-1 | String | APP在DB上的Unique流水號  qp\_app\_head.row\_id |
| app\_code | app\_list | 0-1 | String | APP專案代號 |
| package\_name | app\_list | 0-1 | String | Package name |
| app\_category\_id | app\_list | 0-1 | String | app編號的流水號 |
| default\_lang | app\_list | 0-1 | String | app預設語系en-us |
| app\_version | app\_list | 0-1 | String | APP版號 |
| app\_version\_name | app\_list | 0-1 | String | APP版本名稱 |
| security\_level | app\_list | 0-1 | String | 數字越少security越高  目前先定義三種  1:最高security, on\_resume時必須重新輸入密碼  2:次級security, 關閉app重新開啟時才需要重新輸入密碼  3:一般security, 只要token沒有過期都不需要重新輸入密碼 |
| avg\_score | app\_list | 0-1 | String | APP評分 |
| user\_score | app\_list | 0-1 | String | 這個User的個人評分 |
| sequence | app\_list | 0-1 | Integer | APP排序 |
| ~~android\_url~~ | ~~app\_list~~ | ~~0-1~~ | ~~String~~ | ~~Android版本下載URL~~ |
| ~~ios\_url~~ | ~~app\_list~~ | ~~0-1~~ | ~~String~~ | ~~iOS版本下載URL~~ |
| url | app\_list | 0-1 | String | Android/iOS版本下載URL |
| icon\_url | app\_list | 0-1 | String | Icon圖標下載URL |
| size | app\_list | 0-1 | Integer | 檔案大小, 單位是Bytes |
| multi\_lang | content | 0-N | Container | Container |
| app\_code | multi\_lang | 0-1 | String | APP專案代號 |
| lang | multi\_lang | 0-1 | String | 採用en-us這樣的格式  Ex:  en-us  zh-tw  zh-cn |
| app\_name | multi\_lang | 0-1 | String | APP名稱 |
| app\_summary | multi\_lang | 0-1 | String | APP短摘要 |
| app\_description | multi\_lang | 0-1 | String | APP長描述 |
| pic\_list | multi\_lang | 0-N | Container | Container |
| ~~lang~~ | ~~multi\_lang~~ | ~~0-1~~ | ~~String~~ | ~~採用en-us這樣的格式~~  ~~Ex:~~  ~~en-us~~  ~~zh-tw~~  ~~zh-cn~~ |
| pic\_type | pic\_list | 0-1 | String | 縮圖類別  B:Banner  S:screenshot |
| pic\_url | pic\_list | 0-1 | String | Picture下載url |
| sequence | pic\_list | 0-1 | Integer | Picture的排序 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_user\_app

qp\_app\_head

qp\_app\_line

qp\_app\_version

qp\_app\_pic

qp\_app\_evaluation

qp\_user\_score

qp\_session

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000907 | token過期 | 代表token已經超過使用期限, APP直接切換到登入頁面讓User重新登入  如果過期前有call renewToken API()就不會發生過期狀況 |
| 000908 | token已經失效 | 代表token不存在於系統中, APP直接切換到登入頁面讓User重新登入 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

Check point:

1. qp\_app\_version新增一個欄位external\_app, varchar(1)

1代表是外部app, 例如在google play或是itunes上

0代表檔案在公司SERVER上

1. 抓取url這個欄位時, 需要判斷qp\_app\_version.external\_app=1, 如果是1, 則串url時不把prefix的字串串上去
2. qp\_app\_version.external\_app=0, 維持現有邏輯
3. app端開啟規範

* 如果url節點是market://開頭

表示需要以google market開啟

* 如果url節點是<https://itunes>開頭

表示需要以safari開啟

### 5.7 取得訊息列表 (getMessageList)

#### 5.7.1接口描述

取得Message清單, 將user未刪除過的訊息提供在Qplay app上

#### 5.7.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/getMessageList?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  token值需要使用sha256加密, 加密時需要添加salt值, salt值為request header內的Signature-Time參數  $token = base64\_encode (hash\_hmac(‘sha256’, $Signature-Time, $token’, true)); |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| date\_from | Optional | Integer | 填入取得訊息的起始時間  不填則由SERVER直接抓取上次存取的時間搓, 抓差異化的訊息提供出去  有填date\_from就一定要填上date\_to |
| date\_to | Optional | Integer | 填入取得訊息的結束時間  不填則由SERVER直接抓”當前”的時間搓  有填date\_to就一定要填date\_from |
| count\_from | Optional | Integer | 如果有下date\_time, date\_to, 則這個欄位才有作用, 可以下載指定數量的訊息  沒有下參數, 則抓時間區間內所有資料 |
| count\_to | Optional | Integer | 如果有下date\_time, date\_to, 則這個欄位才有作用, 可以下載指定數量的訊息  沒有下參數, 則抓時間區間內所有資料 |
| overwrite\_timestamp | Optional | Integer | 如果有下date\_time, date\_to, 則這個欄位才有作用  如果是1, 則代表要更新獲取訊息的時間搓記  會更新qp\_register裡面的last\_message\_time  如果是0, 代表不更新不時間搓記  不會更新qp\_register裡面的last\_message\_time  如果不提供這個參數, 代表是0 |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| message\_count | content | 0-1 | Integer | 此次查詢的訊息總數量 |
| message\_list | content | 0-N | Container | 訊息列表container, 有多個訊息 |
| message\_row\_id | message\_list | 0-1 | Integer | 訊息unique id |
| message\_send\_row\_id | message\_list | 0-1 | Integer | 訊息send的unique id |
| message\_type | message\_list | 0-1 | String | 訊息類別, 目前分類:  news  event |
| message\_title | message\_list | 0-1 | String | 訊息標題 |
| message\_txt | message\_list | 0-1 | String | 訊息純文字格式 |
| message\_html | message\_list | 0-1 | String | 訊息html格式 |
| message\_url | message\_list | 0-1 | String | 訊息網址, 如果是空白, 表示該訊息沒有網址可以查看 |
| read | message\_list | 0-1 | Boolean | 是否已經看過該訊息  只有event類別會有這個flag, news類別不會傳送此欄位 |
| message\_source | message\_list | 0-1 | String | 訊息是從哪個平台傳遞過來  目前平台未定義  現階段:qplay |
| read\_time | message\_list | 0-1 | String | 查看訊息時間 |
| create\_user | message\_list | 0-1 | String | 訊息發布者 |
| create\_time | message\_list | 0-1 | Integer | 訊息發布時間  直接抓qp\_user\_message或qp\_role\_message裡面的create\_time |
| source\_user | content | 0-1 | String | 這筆推播是誰送出來 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_message

qp\_user\_message

qp\_session (by uuid抓取上次最後詢問MESSAGE的時間)

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000907 | token過期 | 代表token已經超過使用期限, APP直接切換到登入頁面讓User重新登入  如果過期前有call renewToken API()就不會發生過期狀況 |
| 000908 | token已經失效 | 代表token不存在於系統中, APP直接切換到登入頁面讓User重新登入 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

Check Point:

1. 如果date\_from, date\_to有填資料的時候(通常一次查詢的資料太多, 要下分批count查詢, 此時就會丟時間), 代表是APP端自行下區間來查找資料, 因此不更新qp\_register裡面的last\_message\_time
2. 如果date\_from, date\_to沒有填資料時, 通常就是手機一進入Qplay app, 拿取getMessageList時, 這時候不丟參數, 則由API默認帶入查詢, 並且此時要更新qp\_register.last\_message\_time
3. Overwrite\_timestamp:

如果有下date\_time, date\_to, 則這個欄位才有作用

如果是1, 則代表要更新獲取訊息的時間搓記

會更新qp\_register裡面的last\_message\_time

如果是0, 代表不更新不時間搓記

不會更新qp\_register裡面的last\_message\_time

如果不提供這個參數, 代表是0

1. 會有一人多隻手機的問題

getMessageList, getMessageDetail這兩隻API, 獲得資訊時

請調整為加入where條件

**qp\_user\_message.uuid=XXXXXX and deleted\_at <>'0000-00-00 00:00:00'**

### 5.8 取得安全性資料列表 (getSecurityList)

#### 5.8.1接口描述

通常在login成功後, 再透過token來取, 除了白名單外(也就是允許存許的URL), 還會提供security level資訊

Block List則不在這支API上取得, 如果是在黑名單內, 會直接由API server reject回應在錯誤碼上, 錯誤碼為999009:禁止存取API

#### 5.8.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/getSecurityList?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  token值需要使用sha256加密, 加密時需要添加salt值, salt值為request header內的Signature-Time參數  $token = base64\_encode (hash\_hmac(‘sha256’, $Signature-Time, $token’, true)); |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| app\_key | Required | string | 請填入app的app\_key  ex:qplay, phonebook |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| security\_level | NA | 0-1 | Integer | 數字越少security越高  目前先定義三種  1:最高security, on\_resume時必須重新輸入密碼  2:次級security, 關閉app重新開啟時才需要重新輸入密碼  3:一般security, 只要token沒有過期都不需要重新輸入密碼 |
| allow\_url | content | 0-N | String | 允許存取的url |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000909 | app key不存在 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_while\_list

qp\_app\_head

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000907 | token過期 | 代表token已經超過使用期限, APP直接切換到登入頁面讓User重新登入  如果過期前有call renewToken API()就不會發生過期狀況 |
| 000908 | token已經失效 | 代表token不存在於系統中, APP直接切換到登入頁面讓User重新登入 |
| 000909 | app key不存在 | 一般狀況下不會出現此訊息, 若出現此訊息表示該專案可能已被下架或不提供服務, 因此需提示錯誤後並關閉APP  提示錯誤:  此系統服務已被關閉, 若需要進一步訊息, 請聯絡系統管理員 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

### 5.9 取得訊息詳細資訊 (getMessageDetail)

#### 5.9.1接口描述

 透過訊息id取得訊息詳細資料

#### 5.9.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/getMessageDetail?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  token值需要使用sha256加密, 加密時需要添加salt值, salt值為request header內的Signature-Time參數  $token = base64\_encode (hash\_hmac(‘sha256’, $Signature-Time, $token’, true)); |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| message\_send\_row\_id | Required | Integer | message的唯一識別碼 |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| ~~message\_count~~ | ~~content~~ | ~~0-1~~ | ~~Integer~~ | ~~此次查詢的訊息總數量~~ |
| ~~message\_list~~ | ~~content~~ | ~~0-N~~ | ~~Container~~ | ~~訊息列表container, 有多個訊息~~ |
| message\_send\_row\_id | content | 0-1 | Integer | 訊息unique id |
| message\_type | content | 0-1 | String | 訊息類別, 目前分類:  news  event |
| template\_id | content | 0-1 | Integer | 樣板編號  目前合法範圍是1-16 |
| message\_title | content | 0-1 | String | 訊息標題 |
| message\_txt | content | 0-1 | String | 訊息純文字格式 |
| message\_html | content | 0-1 | String | 訊息html格式 |
| message\_url | content | 0-1 | String | 訊息網址, 如果是空白, 表示該訊息沒有網址可以查看 |
| read | content | 0-1 | Boolean | 是否已經看過該訊息  只有event類別會有這個flag, news類別不會傳送此欄位 |
| message\_source | content | 0-1 | String | 訊息是從哪個平台傳遞過來  目前平台未定義 |
| read\_time | content | 0-1 | String | 查看訊息時間 |
| create\_user | content | 0-1 | String | 訊息發布者 |
| create\_time | content | 0-1 | Integer | 訊息發布時間  抓的是qp\_message.create\_time |
| source\_user | content | 0-1 | String | 這筆推播是誰送出來 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000914 | 帳號已被停權 |
| 000910 | 訊息不存在 |
| 000919 | 訊息被移除 |

**Related Tables:**

qp\_message

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000907 | token過期 | 代表token已經超過使用期限, APP直接切換到登入頁面讓User重新登入  如果過期前有call renewToken API()就不會發生過期狀況 |
| 000908 | token已經失效 | 代表token不存在於系統中, APP直接切換到登入頁面讓User重新登入 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |
| 000910 | 此訊息不存在 | 代表要獲取的message不存在DB中, 因此會造成資訊不對稱, 須清除APP端該則訊息 |
| 000919 | 此訊息已被移除 | 代表要獲取的message不存在DB中, 因此會造成資訊不對稱, 須清除APP端該則訊息 |

Check Pint:

1. 會有一人多隻手機的問題

因此DB結構上針對read, delete需要再另外處理, 需針對uuid個別處理

getMessageList, getMessageDetail這兩隻API, 獲得資訊時,

請調整為加入where條件**qp\_user\_message.uuid=XXXXXX and deleted\_at <>'0000-00-00 00:00:00'**

### 5.10 更新已讀訊息(updateMessage)

#### 5.10.1接口描述

更新message的狀態, 將狀態傳送到server

目前只開放一種狀態更新, 只有event的已讀狀態的才會傳

將來會逐步開放刪除或news類別更新

#### 5.10.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/updateMessage?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  token值需要使用sha256加密, 加密時需要添加salt值, salt值為request header內的Signature-Time參數  $token = base64\_encode (hash\_hmac(‘sha256’, $Signature-Time, $token’, true)); |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| message\_send\_row\_id | Required | integer | message row id |
| message\_type | Required | string | news  event |
| status | Required | string | read  delete  目前只開放read |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| message\_send\_row\_id | content | 0-1 | Integer | message row id |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000910 | 此訊息不存在 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_user\_message

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000907 | token過期 | 代表token已經超過使用期限, APP直接切換到登入頁面讓User重新登入  如果過期前有call renewToken API()就不會發生過期狀況 |
| 000908 | token已經失效 | 代表token不存在於系統中, APP直接切換到登入頁面讓User重新登入 |
| 000910 | 此訊息不存在 | 代表要更新的message不存在DB中, 因此會造成資訊不對稱, 須清除APP端該則訊息 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

Check Point:

1. message\_send\_row\_id=all表示刪除該帳號底下所有訊息, 必須卡message\_type
2. 當處理一筆以上的id時, message\_send\_row\_id=1,2,3,4,5,6,7用逗號,分隔
3. 會有一人多隻手機的問題

因此DB結構上針對read, delete需要再另外處理, 需針對uuid個別處理

3.1 新增欄位, qp\_user\_message新增uuid欄位

[ALTER](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) [TABLE](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) `qp\_user\_message` ADD `uuid` [VARCHAR](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-types.html)(500) [CHARACTER](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/charset.html) [SET](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/charset.html) utf8 [COLLATE](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/charset-collations.html) utf8\_general\_ci [NOT](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL AFTER `message\_send\_row\_id`;

3.2 updateMessage時, where條件, 新增uuid=xxxxxx

### 5.11 傳送推播token給Server (sendPushToken)

#### 5.11.1接口描述

採用Jpush極光推送機制

所以手機需要向Jpush Server註冊後, 發送RegisterID給Qplay server

~~推送Client端向APNS或GCM註冊後, 取得device\_token或是register\_id後, 需要向qplay server註冊這個token~~

~~iOS需特別注意, 每次開啟app後理應向iOS server重新取得token, 並再次向qplay註冊~~

~~這個動作一般接在register API之後, 認證設備後, 需要立刻送出PushToken給server~~

~~The device token is your key to sending push notifications to your app on a specific device. Device tokens can change, so your app needs to reregister every time it is launched and pass the received token back to your server. If you fail to update the device token, remote notifications might not make their way to the user’s device. Device tokens always change when the user restores backup data to a new device or computer or reinstalls the operating system. When migrating data to a new device or computer, the user must launch your app once before remote notifications can be delivered to that device.~~

~~上文出處~~

[~~https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/IPhoneOSClientImp.html#//apple\_ref/doc/uid/TP40008194-CH103-SW1~~](https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/Chapters/IPhoneOSClientImp.html#//apple_ref/doc/uid/TP40008194-CH103-SW1)

~~IOS7之前，apple對於一個設備上的多個APP，生成相同的DeviceToken。  
IOS7以及之後，apple對於一個設備上的多個APP，生成不同的DeviceToken~~

#### 5.11.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/sendPushToken?lang=en-us
3. header請求參數:

sendPushToken時, 需要將push-token欄位透過HTTP Header送出, 所以包含下面參數：

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| push-token | 必填 | ~~iOS為device\_token~~  ~~android為register\_id~~  傳送向極光SERVER獲取的RegisterID |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| app\_key | Required | string | 如果是qplay平台用的, 就傳qplay  假設是e04也許就會傳phonebook這個值 |
| device\_type | Required | string | iOS或是Android  提供server辨識該向APNS或是GCM發送推播 |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | NA | 0-1 | String | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c  成功的話, 會傳送此次交易的uuid供double check |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000909 | app key不存在 |
| 000911 | uuid不存在 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_push\_token

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000909 | app key不存在 | 一般狀況下不會出現此訊息, 若出現此訊息表示該專案可能已被下架或不提供服務, 因此需提示錯誤後並關閉APP  提示錯誤:  此系統服務已被關閉, 若需要進一步訊息, 請聯絡系統管理員 |
| 000911 | uuid不存在 | 一般在登入成功後會送token給server  若此時出現000911, 則表示uuid是不存在於Server端的, 也就是沒有認證過, 這時請提示錯誤訊息, “此移動裝置需經過認證才能使用本系統”, 點擊OK後轉至認證頁面 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

Check Poit:

1. RegisterID的取得請參考Jpush文檔
2. RegistrationID 定义:

集成了 JPush SDK 的应用程序在第一次成功注册到 JPush 服务器时，JPush 服务器会给客户端返回一个唯一的该设备的标识 - RegistrationID。JPush SDK 会以广播的形式发送 RegistrationID 到应用程序。

应用程序可以把此 RegistrationID 保存以自己的应用服务器上，然后就可以根据 RegistrationID 来向设备推送消息或者通知。

1. iOS API - getRegistrationID

调用此 API 来取得应用程序对应的 RegistrationID。 只有当应用程序成功注册到 JPush 的服务器时才返回对应的值，否则返回空字符串。

**支持的版本**

开始支持的版本：1.7.0。

**接口定义**

+(NSString \*)registrationID

温馨提示：

iOS 9系统，应用卸载重装，APNs返回的devicetoken会发生变化，开发者需要获取设备最新的Registration id。请在kJPFNetworkDidLoginNotification的实现方法里面调用"RegistrationID"这个接口来获取 RegistrationID。

1. Android API-getRegistrationID

调用此 API 来取得应用程序对应的 RegistrationID。只有当应用程序成功注册到 JPush 的服务器时才返回对应的值，否则返回空字符串。

**支持的版本**

开始支持的版本：1.6.0。

**接口定义**

//SDK 初次注册成功后，开发者通过在自定义的 Receiver 里监听 Action - cn.jpush.android.intent.REGISTRATION 来获取对应的 RegistrationID。注册成功后，也可以通过此函数获取 public static String getRegistrationID(Context context)

**附加说明**

通过 RegistrationID 进行点对点推送

可以通过 RegistrationID 来推送消息和通知， 参考文档 Push API v2， 当 receiver\_type = 5 并且设置 receiver\_value 为 RegistrationID 时候即可根据 RegistrationID 推送。

注：要使用此功能，客户端 App 一定要集成有 r1.6.0 及以上版本的 JPush Android SDK。

### 5.12 刷新Token有效時間搓記(renewToken)

#### 5.12.1接口描述

Token快過期時, 可以透過這個api重新取得token, 延長token效期,正常狀況下, 一個token可以用48小時, 必須在到期前就call renewToken重新刷新Token的有效期限

#### 5.12.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/renewToken?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  token值需要使用sha256加密, 加密時需要添加salt值, salt值為request header內的Signature-Time參數  $token = base64\_encode (hash\_hmac(‘sha256’, $Signature-Time, $token’, true)); |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | content | 0-1 | String | 設備唯一識別碼 |
| token | content | 0-1 | String | 由server配發的token, 用來識別是否已經登入qplay, 時效為48hour |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000914 | 帳號已被停權 |

**Related Tables:**

qp\_session

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000907 | token過期 | 代表token已經超過使用期限, APP直接切換到登入頁面讓User重新登入 |
| 000908 | token已經失效 | 代表token不存在於系統中, APP直接切換到登入頁面讓User重新登入 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |

### 5.13 傳送推播訊息給Server (sendPushMessage)

#### 5.13.1接口描述

提供接口給外部系統傳送推播訊息, iOS及android是統一接口, 透過此API將資訊送至qplay後, 再由qplay統一向JPUSH發送推播, Jpsuh會再將資訊轉到APNS及GCM

特別注意, 此接口為HTTP POST method

#### 5.13.2接口訊息定義

1. 請求方法：

POST /v101/qplay/sendPushMessage?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| Content-Type | 必填 | 訊息體類型  使用POST方法傳輸, 類型需為application/json |
| App-Key | 必填 | 專案名稱, 例如Qplay平台就是appqplay  此欄位也會用來當作app間識別的scheme  如果是for外部系統推播使用, 請跟Steven Yan討論app-key這個欄位 |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  qplay的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e  每個專案key是不一樣的, 請跟Steven Yan拿取所屬專案的AppSecretKey |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| lang | Required | String | 請填入 語系-國家  ex:  en-us  zh-tw  zh-cn |
| app\_key | Required | string | 如果是qplay平台用的, 就傳qplay  如果是for外部系統推播使用, 請跟Steven Yan討論app-key這個欄位 |
| need\_push | Required | string | Y:要push  N:不需push, 此為默認值  **如果need\_push欄位填上N, 則http request body內的destination\_user\_id或是destination\_role\_id可以為空** |
| push\_time\_utc | Optional | int | unix時間搓  (unix timestamp型式), 共10碼  填入預計推播的時間, 格林威治時間 |
| qplay\_message\_list | Optional | String | 預設qplay\_message\_list = Y  不填此參數表示為Y  也就是會寫入到qplay的message list  例如ENS的推播是屬於論壇性質  不需要顯示在qplay的message list  因此, 請填入qplay\_message\_list = N |

**URL新增的參數qplay\_message\_list, 預設=Y , 也就是會寫到qplay的message list**

qplay\_message\_list = Y，event\_type=news，need\_push=Y  會寫入 MySql: qp\_message,qp\_message\_send,

qplay\_message\_list = Y，event\_type=event，need\_push=Y 會寫入 MySql: qp\_message,qp\_message\_send,qp\_user\_message,qp\_role\_message

qplay\_message\_lsit = N，event\_type=news，need\_push=Y  會寫入 Mongo: qp\_message\_send\_pushonly,需紀錄appkey

qplay\_message\_list, = N，event\_type=event，need\_push=Y 會寫入 Mongo: qp\_message\_send\_pushonly,需紀錄appkey

* [PushUtil]調整

1.新增一個config pushkey.php來管理各app的key

2.以下三支API新增參數，$appKey，預設為未填，讀取qplay的key來推播；若有填寫則用app的key推播

PushMessageWithJPushWebAPI

PushScheduleMessageWithJPushWebAPI

AddTagsWithJPushWebAPI

1. **POST Body請求參數**

請按照以下格式送出資訊, 一定要是Json格式:

{

"template\_id": "1",

"message\_title": "U3lzdGVtIEFubm91bmNlbWVudA==",

"message\_type": "event",

"message\_text": "5oiR5piv5Y+N5pac57eaXA0K5oiR5piv6ZuZ5Y+N5pac57eaXFwNCuaPm+ihjA0KICAg56m655m9DQrmiJHmmK/llq7lvJXomZ8n6Lef6ZuZ5byV6JmfIg0K5oiR5piv5Lit5ous6JmfW10=",

"message\_html": "",

"message\_url": "",

"message\_source": "Oracle ERP",

"source\_user\_id": "qgroup\\Moses.Zhu",

"destination\_user\_id":["qgroup\\Moses.Zhu","qgroup\\Sammi.Yao","qgroup\\Moses.Zhu","qgroup\\Moses.Zhu","qgroup\\Sammi.Yao"],

"destination\_role\_id":["qisda/administrator","benq/ITS","qisda/staff","benq/staff","qisda/AIC0 RD"],

"extra":"123"

}

**注意事項:**

* 1. source\_user\_id的格式為: DomainName\\個人AD帳號

ex:qgroup\\Moses.Zhu

* 1. destination\_user\_id的格式, 如果是個人,格式為:DomainName\\個人AD帳號

ex:qgroup\\Moses.Zhu

* 1. destination\_role\_id的格式為:公司名稱/企業角色

ex:BenQ/ITS

* 1. **message\_html及message\_url這兩個tag暫時不開放給Client填寫, 請勿使用這兩個欄位**

1. **回應訊息：**
2. **目前處理方式, 送出的api中, 如果有任何一個接收者是有問題的, 則整個api reject**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| ~~uuid~~ | ~~NA~~ | ~~0-1~~ | ~~String~~ | ~~設備的unique id~~  ~~uuid: qplay平台的push\_token~~  ~~由client端決定~~  ~~成功的話, 會傳送此次交易的uuid供double check~~ |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000909 | app key不存在 |
| 000912 | 接受推播用戶不存在 |
| 000914 | 帳號已被停權 |
| 000916 | 標題欄位太長 |
| 000917 | 角色不存在 |
| **000918** | 数据不完整 |

**Related Tables:**

qp\_push\_token (拿取token)

qp\_user\_message

qp\_message

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000909 | app key不存在 | 一般狀況下不會出現此訊息, 若出現此訊息表示該專案可能被下架或不提供服務, 因此需提示錯誤後並關閉APP  提示錯誤:  此系統服務已被關閉, 若需要進一步訊息, 請聯絡系統管理員 |
| 000912 | 接受推播用戶不存在 | 送出推播時需要指定接收推播的用戶, 若提供的domain\account不存在於DB中, 則會出現000912訊息  此筆交易會失敗, 請退回到原處理頁面重新檢視destination清單  如果是Qplay推播畫面, 請回到準備送出推播畫面, 並提示錯誤訊息”請重新選擇推播用戶”  如果是custom系統直接推送訊息, 則請custom系統自行處理, 這筆資訊不會推播, 也送不到play server |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |
| 000916 | 標題欄位太長 | Server端提醒Client端標題欄位太長, 導致無法正常推播 |
| 000917 | 角色不存在 | Server端提醒Client端角色不存在, 導致無法正常推播 |

Check point:

1. End User手機端能收到推播訊息的大原則為:

* 已經安裝Qplay app, 並成功登入過Qplay app
* 手機端允許推播功能有開啟

1. Qplay Server收到訊息後, 如果外部系統傳入的網址列參數need\_push=Y, 則Qplay會協助推播到手機端

need\_push=N, 則只會將訊息送到Qplay後台, 不會推送到手機端

* 如果need\_push欄位填上N, 則http request body內的destination\_user\_id或是destination\_role\_id可以為空
* 不管送出的參數 need\_push是Y或是N, 都會同時往qp\_message, qp\_message\_send, qp\_role\_message/qp\_user\_message裡頭塞資料
* qp\_user\_mesage, qp\_role\_message這兩張表用意是用來管控手機的用戶是否有權限可以查看此則消息

qp\_message\_send則是用來控制消息推播狀況, 兩者在意義上是完全可以分割的

* qplay後台的推播管理,顯示歷史推播清單時, 只能顯示qp\_message\_send.need\_push=Y的資料

1. 各節點長度限制

|  |  |  |  |
| --- | --- | --- | --- |
| **欄位** | **型態** | **範例** | **描述** |
| template\_id | int | 1 | 樣板編號, 目前有16種編號, 呈現的樣式由app自行套版, API只提供套版編號  正常狀況下的推播, template\_id請用0  其他對應如下:  template\_id對應類別如下:  0 Event Notice  1 上線公告  2 System/Function Go-Live  3 維護公告  4 System/Service Maintenance  5 月結公告  6 Month End Notice  7 安全公告  8 Information Security  9 服務公告  10 Service Notice  11 重大公告!!  12 (Urgent!!) Critical Notice  13 緊急安全公告!!  14 (Urgent!!) Information Security  15 緊急服務公告!!  16 (Urgent!!) Service Notice  17 Event Notice-Urgent |
| message\_type | varchar(50) | news | 目前訊息有兩種  news  event |
| message\_title | varchar(264) |  | 訊息標題:  先以escape編碼, 再以base64編碼  ex:  var strText=$.base64.btoa(escape("這是推播標題"));  編碼後為:  JXU5MDE5JXU2NjJGJXU2M0E4JXU2NEFEJXU2QTE5JXU5ODRD |
| message\_text | varchar(5000) |  | 文字格式的訊息:  先以escape編碼, 再以base64編碼  支持html tag  ex:  var strText=$.base64.btoa(escape("<html>你们好,我很好</html>"));  編碼後為:  JTNDaHRtbCUzRSV1NEY2MCV1NEVFQyV1NTk3RCUyQyV1NjIxMSV1NUY4OCV1NTk3RCUzQy9odG1sJTNF |
| message\_html | text |  | html格式:  先以escape編碼, 再以base64編碼  ex:  var strText=$.base64.btoa(escape("<html>你们好,我很好</html>"));  編碼後為:  JTNDaHRtbCUzRSV1NEY2MCV1NEVFQyV1NTk3RCUyQyV1NjIxMSV1NUY4OCV1NTk3RCUzQy9odG1sJTNF |
| message\_url | varchar(500) |  | URL格式, 可以直接連到網頁查看:  先以escape編碼, 再以base64編碼  ex:  var strText=$.base64.btoa(escape("http://www.google.com"));  編碼後為:  aHR0cCUzQS8vd3d3Lmdvb2dsZS5jb20= |
| message\_source | varchar(50) | qplay | 預設是qplay  如果是for外部系統推播使用, 請跟Steven Yan討論message\_source這個欄位 |

1. "message\_title": 訊息標題, 基本上推送的時候, Qplay只會幫忙推送標題¸ 內文是不推送的

* 此欄位在call api傳送資料前必須編碼,為支持中文,先進行escape編碼, 然後再以base64編碼, charset=ntf-8
* message\_title這個欄位在通過escape及base64 編碼後,長度限制要在396Bytes內, 因為base64編碼後, 無論中英文都會被編成4bytes, 因此換算只允許中英文或編點符號或空白字組合共99個字, 長度太長會收到錯誤碼000916

( mysql定義為varchar(99), 不過APNs其實限制為2048Bytes, 極光定義為2000Bytes, 包含iOS:{….} 從iOS開頭到 } 結尾的所有字元 )

* 因此, 標題請限制在99個字以內
* 必須將message\_title欄位轉成base64 encode的字串, charset=utf-8

1. "message\_type": 目前定義兩種

Event:發給特定人, 或是特定群組, 但無法發給全公司

群組定義下面章節會介紹

News:發給全公司, 無法選擇特定人

Api只接受以上兩種, 其餘會reject

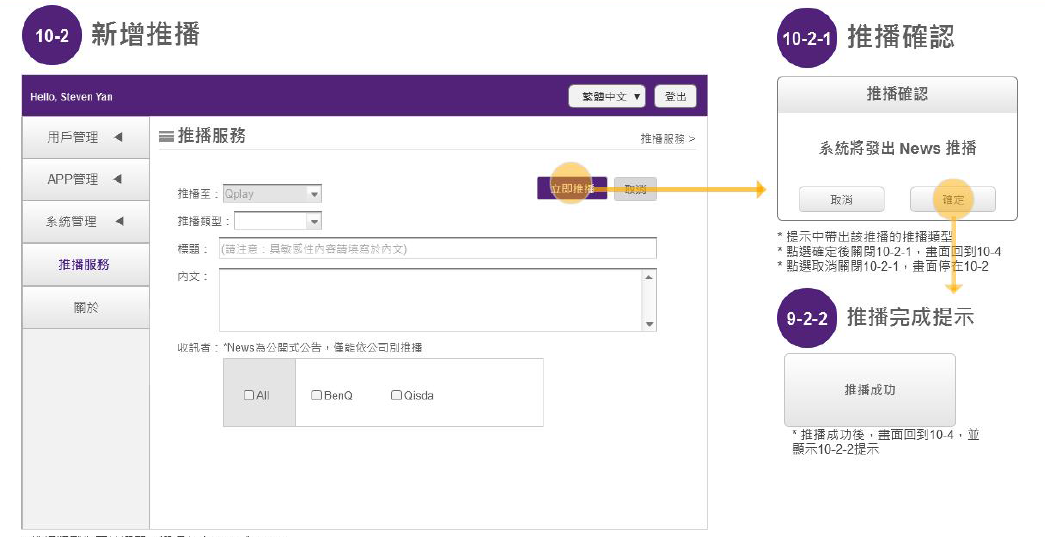
1. "message\_text": 這個節點放的是訊息內容, 這個欄位不會在推播的payload裡面, 推播主體只會有"message\_title"的內容

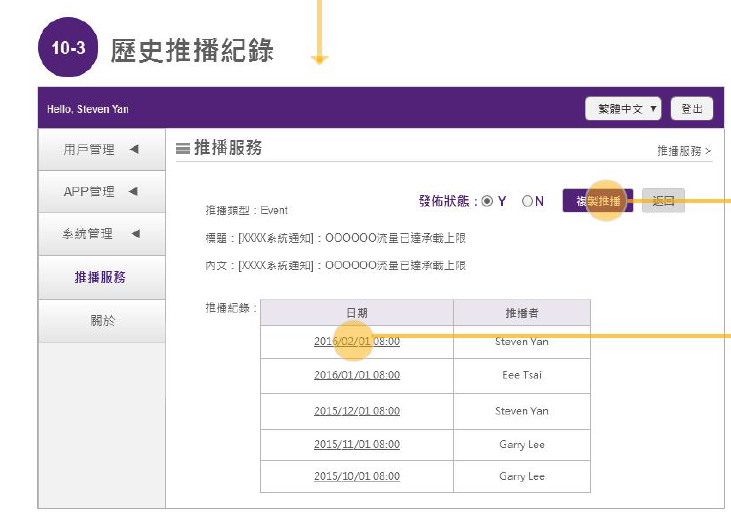
* call api傳送資料前必須編碼,為支持中文,先進行escape編碼, 然後再以base64編碼, charset=ntf-8, 如果字串中有換行, 支持html語法, 請直接在字串中填入<BR>, 但& “ ‘ < >**這五個特殊符號**是html保留字元, 所以要先轉換為

&ltBR&gt, 再做escape, 再轉成base64

* Client端(外部系統) call api後, 成功送出給Qplay Platform後, message\_text這個欄位, 會被base64\_decode後存放在Qplay database內, 透過Qplay後台可以查看這個訊息內文, 也可以重新推送到手機端, 所以送出api時, 網址列的need\_push參數可以填N, 進入Qplay Platform後, 再從後台叫出來推播, 如下示意圖:







1. "message\_html": 同message\_text, 不會在推送的payload裡面, 但會存在於Qplay Database裡面, 跟text最大差別是可以處理html字串, 因此有html字串請放置在這個節點上

* call api傳送資料前必須編碼,為支持中文,先進行escape編碼, 然後再以base64編碼, charset=ntf-8,

ex:

var strText=$.base64.btoa(escape("<html>你们好,我很好</html>"));

編碼後為:

JTNDaHRtbCUzRSV1NEY2MCV1NEVFQyV1NTk3RCUyQyV1NjIxMSV1NUY4OCV1NTk3RCUzQy9odG1sJTNF

* **此階段不開放這個節點, 外部系統暫時請勿使用**

1. "message\_url": 同message\_text, 不會在推送的payload裡面, 但會存在於Qplay Database裡面, 如果這則訊息有網頁可以查看, 可以直接將URL放在這個欄位, Qplay端可以直接開啟網頁, 請注意, 因為是手機瀏覽, 請注意如果要傳送此欄位, 要符合手機layout(最好是RWD寫法), 且必須是對外服務的網址

* call api傳送資料前必須將message\_text欄位先以escape編碼, 再以base64編碼

ex:

var strText=$.base64.btoa(escape("http://www.google.com"));

編碼後為:

aHR0cCUzQS8vd3d3Lmdvb2dsZS5jb20=

* **此階段不開放這個節點, 外部系統暫時請勿使用**

1. "message\_source": 基本上用來記錄要求的外部系統名稱, 請填入適當且容易識別的外部系統名稱
2. "source\_user\_id": 放的是發送訊息的ad帳號資訊, 格式為domainName\\AD帳號, 且中間採用雙斜線

Ex: "benq\\steven.yan" 或是 "qgroup\\ike.huang"

1. "destination\_user\_id": 放的是接收訊息的ad帳號資訊, 如果是發送到個人, 格式為domainName\\AD帳號, 且中間採用雙斜線

如果多個請用[ ]中括號包著, 裡面用 , 逗號隔開

EX:

["benq\\steven.yan","qgroup\\ike.huang","qgroup\\Moses.Zhu","qgroup\\Sammi.Yao"]

* 如果沒有接收端, 也要放這個節點, 請用空白表達:

Ex: "destination\_user\_id":[]

* 如果是單個接收者, []中括號也要放, ["benq\\steven.yan"]

1. "destination\_role\_id": 如果要發送的是一個群組

可以直接發群組名稱, 群組名稱前面必須補上公司, 請注意不是domain, 且中間採用單斜線隔開

如果多個請用[ ]中括號包著, 裡面用 , 逗號隔開

Ex: ["qisda/administrator","benq/ITS","qisda/staff","benq/staff","qisda/AIC0 RD"]

外部系統其實沒有群組清單, 群組清單只存在於Qplay平台, 所以建議外部系統發訊息時, 直接採用destination\_user\_id這個節點即可, destination\_role\_id放空白, 或事先與Qplay平台負責人確認群組清單,

* 如果沒有接收端, 也要放這個節點, 請用空白表達:

Ex: "destination\_role\_id":[]

* 如果是單個接收者, []中括號也要放, ["benq/its"]

1. **如果是News類型, 表示要推送到整個公司**

所以destination\_user\_id的節點, 後面直接放公司名稱

Ex:

"destination\_user\_id":["benq","qisda"],

這邊定需要特別注意一下, 如果是要推送到全公司的話, 一定要用Tag的方式來做, 一個Tag底下最多一千個手機, 所以每一家公司我計畫按login id分成6個group

ABCD開頭的第一個Group, 命名為BENQ1, 也就是”公司名稱”加”流水號”

EFGH開頭的是第二個Group, 命名為BENQ2

IJKL開頭的是第三個Group, 命名為BENQ3

MNOP開頭的是第四個Group, 命名為BENQ4

QRST開頭的是第五個Group, 命名為BENQ5

UVWXYZ開頭的是第六個Group, 命名為BENQ6

Qisda同樣方式:

ABCD開頭的第一個Group, 命名為QISDA1, 也就是”公司名稱”加”流水號”

EFGH開頭的是第二個Group, 命名為QISDA2

IJKL開頭的是第三個Group, 命名為QISDA3

MNOP開頭的是第四個Group, 命名為QISDA4

QRST開頭的是第五個Group, 命名為QISDA5

UVWXYZ開頭的是第六個Group, 命名為QISDA6

例如Steven.Yan, 則歸屬於BENQ5這個Tag

只要是News推播, 就是全公司用TAG推六次

P.S1 要特別注意新進人員跟離職人員, 必須要去更新Tag內每個Group目前最新的清單

P.S2 Tag的行為是由Qplay Server來做

1. 推播時調用:

直接呼叫JPUSH的restful API

<https://docs.jiguang.cn/jpush/server/push/rest_api_v3_push/>

推送時的語法, 大部分維持不變

Audience的地方

本來放registration\_id, 改成放Tag

Ex:修改下面紅字部分到Tag即可

{

"platform":[

"ios",

"android"

],

**"audience":{**

**"tag": [**

**"BENQ1",**

**"Qisda1"**

**]**

**},**

"notification":{

"android":{

"alert":"RMC(1) QTY 13:58 ESH incident injury 0",

"extras":{

"Parameter":671

}

},

"ios":{

"alert":"RMC(1) QTY 13:58 ESH incident injury 0",

"sound":"default",

"badge":0,

"extras":{

"Parameter":671

}

}

},

"message":{

"msg\_content":"RMC(1) QTY 13:58 ESH incident injury 0",

"title":"RMC(1) QTY 13:58 ESH incident injury 0",

"content\_type":"text",

"extras":{

"Parameter":671

}

},

"options":{

"apns\_production":true

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **关键字** | **类型** | **含义** | **说明** | **备注** |
| tag | JSON Array | 标签 | 数组。多个标签之间是 OR 的关系，即取并集。 | 用标签来进行大规模的设备属性、用户属性分群。 一次推送最多 20 个。   * 有效的 tag 组成：字母（区分大小写）、数字、下划线、汉字、特殊字符@!#$&\*+=.|￥。 * 限制：每一个 tag 的长度限制为 40 字节。（判断长度需采用UTF-8编码） |
| tag\_and | JSON Array | 标签AND | 数组。多个标签之间是 AND 关系，即取交集。 | 注册与 tag 区分。一次推送最多 20 个。 |

Ex:

* 推送给多个标签（只要在任何一个标签范围内都满足）：在深圳、广州、或者北京

{

    "audience" : {

        "tag" : [ "深圳", "广州", "北京" ]

    }

}

* 推送给多个标签（需要同时在多个标签范围内）：在深圳并且是“女”

{

    "audience" : {

        "tag\_and" : [ "深圳", "女" ]

    }

}

1. 註冊Tag時調用:

直接呼叫JPUSH的restful API

<https://docs.jiguang.cn/jpush/server/push/rest_api_v3_device/>

**API 概述**

Device API 用于在服务器端查询、设置、更新、删除设备的 tag,alias 信息。

包含了device, tag, alias 三组API。其中：

* device 用于查询/设置设备的各种属性，包含tags, alias；
* tag 用于查询/设置/删除设备的标签；
* alias 用于查询/设置/删除设备的别名。

API URL: [https://device.jpush.cn](https://device.jpush.cn/)

**查询设备的别名与标签**

GET /v3/devices/{registration\_id}

获取当前设备的所有属性，包含tags, alias，手机号码mobile。

**Example Request**

**Request Header**

GET /v3/devices/{registration\_id}

  Authorization: Basic (base64 auth string)

  Accept: application/json

**Request Params**

* N/A

**Example Response**

**Response Header**

HTTP/1.1 200 OK

  Content-Type: application/json; charset=utf-8

**Response Data**

{

     "tags": ["tag1", "tag2"],

     "alias": "alias1",

     "mobile": "13012345678"

}

* 找不到统计项就是null,否则为统计项的值

**设置设备的别名与标签**

使用短信业务，请结合服务端[SMS\_MESSAGE](https://docs.jiguang.cn/jpush/server/push/rest_api_v3_push/#sms_message)字段

POST /v3/devices/{registration\_id}

更新当前设备的指定属性，当前支持tags, alias，手机号码mobile。

**Example Request**

**Request Header**

POST /v3/devices/{registration\_id}

  Authorization: Basic (base64 auth string)

  Accept: application/json

**Request Body**

{

        "tags":{

            "add": [

                "tag1",

                "tag2"

            ],

            "remove": [

                "tag3",

                "tag4"

            ]

        },

        "alias": "alias1",

        "mobile":"13012345678"

    }

**Request Params**

* tags: 支持add, remove 或者空字符串。当tags参数为空字符串的时候，表示清空所有的 tags；add/remove 下是增加或删除指定的 tag；
* alias: 更新设备的别名属性；当别名为空串时，删除指定设备的别名；
* mobile: 设备关联的手机号码

**Example Response**

**Response Header**

HTTP/1.1 200 OK

  Content-Type: application/json; charset=utf-8

**Response Data**

* N/A

**更新标签**

为一个标签添加或者删除设备。

POST /v3/tags/{tag\_value}

**Example Request**

**Request Header**

POST /v3/tags/{tag\_value}

  Authorization: Basic (base64 auth string)

  Accept: application/json

**Request Body**

{

        "registration\_ids":{

            "add": [

                "registration\_id1",

                "registration\_id2"

            ],

            "remove": [

                "registration\_id3",

                "registration\_id4"

            ]

        }

}

**Request Params**

* action操作类型，有两个可选："add"，"remove"，标识本次请求是"添加"还是"删除"。
* registration\_ids 需要添加/删除的设备registration\_id。
* add/remove最多各支持1000个；

**Example Response**

**Response Header**

HTTP/1.1 200 OK

 Content-Type: application/json; charset=utf-8

**Response Data**

* N/A

**删除标签**

删除一个标签，以及标签与设备之间的关联关系。

DELETE /v3/tags/{tag\_value}

**Example Request**

**Request Header**

DELETE /v3/tags/{tag\_value}?platform=android,ios

  Authorization: Basic (base64 auth string)

  Accept: application/json

**Request Params**

* platform 可选参数，不填则默认为所有平台。

**Example Response**

* N/A

1. 如果是推送到個人, 維持原架構, domain name\\ad id

Ex:

"destination\_user\_id": {

"user\_id": "benq\\steven.yan",

"user\_id": "qgroup\\eee.tsai"

},

1. 如果是推送到企業角色, 也就是用Jpush的Tag功能推送

則透過destination\_role\_id這個節點資料來送

須注意企業角色的Tag是**company\_role**

Ex:BenQ\_ITS, Qisda\_Supporting, BenQ\_CC…etc

P.S1 當然後台更新企業角色或是跟Jpush註冊企業角色就需要非常注意Tag的正確性及即時性

P.S2 Tag的行為是由Qplay Server來做, 更新企業角色時一定要同步註冊或取消註冊

1. 推送採用Jpush平台推送
2. 使用JPUSH REST API

直接呼叫JPUSH的restful API

<https://docs.jiguang.cn/jpush/server/push/rest_api_v3_push/>

其中幾個重點:

0.

寫Code時如果要測試的話

請使用Validate API就好

不用實際發訊息

**推送校验 API**

POST <https://api.jpush.cn/v3/push/validate>

**功能说明**

该 API 只用于验证推送调用是否能够成功，与推送 API 的区别在于：不向用户发送任何消息。 其他字段说明：同推送 API。

**调用返回**

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **描述** | **详细解释** | **HTTP Status Code** |
| 1000 | 系统内部错误 | 服务器端内部逻辑错误，请稍后重试。 | 500 |
| 1001 | 只支持 HTTP Post 方法 | 不支持 Get 方法。 | 405 |
| 1002 | 缺少了必须的参数 | 必须改正 | 400 |
| 1003 | 参数值不合法 | 必须改正，如Audience参数中tag，alias，registration\_id有空值，错误提示Empty tag/alias/registration\_id is not allowed! | 400 |
| 1004 | 验证失败 | 必须改正。详情请看：[调用验证](https://docs.jiguang.cn/jpush/server/push/rest_api_v3_push/#_5) | 401 |
| 1005 | 消息体太大 | 必须改正。 Android平台Notification+Message长度限制为4000字节； iOS Notification 中 “iOS”:{ } 及大括号内的总体长度不超过：2000个字节（包括自定义参数和符号），iOS 的 Message部分长度不超过 4000 字节； WinPhone平台Notification长度限制为1000字节 | 400 |
| 1008 | app\_key参数非法 | 必须改正 | 400 |
| 1011 | 没有满足条件的推送目标 | 请检查audience | 400 |
| 1020 | 只支持 HTTPS 请求 | 必须改正 | 404 |
| 1030 | 内部服务超时 | 稍后重试 | 503 |

**1.**

app\_key:33938c8b001b601c1e647cbd

secret\_key:aa15c71a5071f539240b5c83"

**2.**

新增DB欄位

qp\_message\_send新增JPUSH的error\_code

[ALTER](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) [TABLE](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) `qp\_message\_send` ADD `jpush\_error\_code` [VARCHAR](http://qplaytest.benq.com/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-types.html)(500) NULL AFTER `push\_flag`;

<https://docs.jiguang.cn/jpush/server/push/http_status_code/>

填入jpush\_error\_code時

請直接將整個json填入

Ex:{"error": {"message": "Authen failed","code": 1004}}

**3.**

推送格式:

**3.1**

URL:https://api.jpush.cn/v3/push

https post方式

**3.2**

Headers請填入

Authorization: Basic base64\_auth\_string

#Basic不能漏掉, 一定要寫

Base64計算如下:

**调用验证**

HTTP Header（头）里加一个字段（Key/Value对）：

Authorization: Basic base64\_auth\_string

其中 base64\_auth\_string 的生成算法为：base64(appKey:masterSecret)  
即，对 appKey 加上冒号，加上 masterSecret 拼装起来的字符串，再做 base64 转换。

Ex:

Base64(33938c8b001b601c1e647cbd:aa15c71a5071f539240b5c83)

轉出後為

MzM5MzhjOGIwMDFiNjAxYzFlNjQ3Y2JkOmFhMTVjNzFhNTA3MWY1MzkyNDBiNWM4Mw==

**3.3**

http post body規範如下:

{

"platform": "all",

"audience": {"registration\_id" : ["141fe1da9ea53377333", "1a0018970aa806ab91c"]},

"notification": {

"android": {

"alert": "Hi, the message from Steven to Test Jpush!",

"extras": {

"Parameter": "215"

}

},

"ios": {

"alert": "Hi, the message from Steven to Test Jpush!",

"sound": "default",

"badge": "0",

"extras": {

"Parameter": "215"

}

}

},

"options": {

"time\_to\_live": 60,

"apns\_production": true

}

}

**3.4**

Audience:直接推送到多個註冊ID

{

    "audience" : {

        "registration\_id" : [ "4312kjklfds2", "8914afd2", "45fdsa31" ]

    }

}

**3.5**

Alert格式:

直接分成ios及android

Alert內填入USER送出的標題, 也就是qp\_message.message\_title

extras這個節點

需要將message\_row\_id送給手機端

請參考Moses送入Message Center的格式

其餘格式按下面格式送

"notification": {

"android": {

"alert": "Hi, the message from Steven to Test Jpush!",

"extras": {

"Parameter": "215"

}

},

"ios": {

"alert": "Hi, the message from Steven to Test Jpush!",

"sound": "default",

"badge": "0",

"extras": {

"Parameter": "215"

}

}

},

**4.**

頻率:

請控制一分鐘內不要連續推600次

如果audience超過600人

請分批推送

**5. 請在config/app.php裡面新增4個參數**

**App\_id**

**Secret\_key**

**apns\_flag**:true

可能是true或是false

如果目标平台为 iOS 平台 需要在 options 中通过 apns\_production 字段来设定推送环境。True 表示推送生产环境，False 表示要推送开发环境； 如果不指定则为推送生产环境

**time\_to\_live:** 864000

推送当前用户不在线时，为该用户保留多长时间的离线消息，以便其上线时再次推送。默认 86400 （1 天），最长 10 天。设置为 0 表示不保留离线消息，只有推送当前在线的用户可以收到。

EX:

{

    "platform": "all",

    "audience": {

        "tag": [

            "深圳",

            "北京"

        ]

    },

    "notification": {

        "android": {

            "alert": "Hi, JPush!",

            "title": "Send to Android",

            "builder\_id": 1,

            "extras": {

                "newsid": 321

            }

        },

        "ios": {

            "alert": "Hi, JPush!",

            "sound": "default",

            "badge": "+1",

            "extras": {

                "newsid": 321

            }

        }

    },

    "message": {

        "msg\_content": "Hi,JPush",

        "content\_type": "text",

        "title": "msg",

        "extras": {

            "key": "value"

        }

    },

    "sms\_message":{

        "content":"sms msg content",

        "delay\_time":3600

    },

**"options": {**

**"time\_to\_live": 60,**

**"apns\_production": false**

**}**

}

1. 群推透過Tag功能完成

避免發送太多次的推送命令

**(企業角色維護的時候, 需要更新Jpush上的Tag清單)**

1. 推播服務直接參考Appedix章節, 但推播時的長度另外說明如下:

考量到以下iOS限制

"iOS":{ } 及大括号内的总体长度不超过：2000 个字节。

另外，JPush 在推送时使用 utf-8 编码，所以一个汉字占用 3 个字节长度。

* 因此, API需要特別針對標題長度限制

call api傳送資料前必須將message\_title欄位先以escape編碼後, 再轉成base64 encode的字串, charset=utf-8

Ex: 你们好\n我很好

Escape編碼, 再轉成base64後為: JXU0RjYwJXU0RUVDJXU1OTdEJTVDbiV1NjIxMSV1NUY4OCV1NTk3RA==

* message\_title這個欄位在通過escape及base64 編碼後,長度限制要在264個字元內(換算成中文字約33個字, 英文約99個字), 長度太長會收到錯誤碼000916

1. 會有一人多隻手機的問題

因此DB結構上需針對uuid個別處理

影響sendPushMessage, 推播後台

sendPushMessage, 推播後台這兩個功能

1.新增到qp\_user\_message表的時候, 多塞一個uuid欄位

1. 新增定時推播功能
2. sendPushMessage 的url參數增加一個Optional參數:push\_time\_utc:[unix時間, 10位數]
3. 後台一般推播, 秘書推播兩個畫面, 加入預計推播時間的選擇

schedule推播格式如下:

<https://docs.jiguang.cn/jpush/server/push/rest_api_push_schedule/>

**创建定时任务**

POST <https://api.jpush.cn/v3/schedules>

**Example Request**

**Request Header**

POST /v3/schedules

Authorization: Basic (base64 auth string)

Content-Type: application/json

Accept: application/json

**Request Body(紅字為比”一般推播”多出來的部分)**

{

"name": "send763",

 "enabled": true,

 "trigger": {

        "single":{

            "time": "2014-09-17 12:00:00"  //YYYY-MM-DD HH:MM:SS

         }

},

"push": {

      "platform": [

        "ios","android"

       ],

      "audience": {

        "registration\_id": [

        "1a0018970aa8fef3a1a","18171adc03039c4ceeb"

        ]

      },

      "notification": {

        "android": {

            "alert": "請確認android推播能否點進DETAIL",

            "extras": {

            "Parameter": 406

            }

        },

        "ios": {

            "alert": "請確認iOS推播能否點進DETAIL",

            "sound": "default",

            "badge": "0",

            "extras": {

            "Parameter": 406

            }

        }

      },

      "message": {

        "msg\_content": "請確認推播能否點進DETAIL",

        "extras": {

        "Parameter": 406

        }

      },

      "options": {

      "time\_to\_live": 60,

      "apns\_production": "true"

      }

}

}

**說明:**

每一个schedule任务，都由name、enabled、trigger、push这四个小节组成。

* **name**
  + 表示schedule任务的名字，由schedule-api在用户成功创建schedule任务后返回，不得超过255字节，由汉字、字母、数字、下划线组成。

**以下定義name參數, 使其為unique, 將來可以覆蓋預約推播的時間**

* 一般定時推播定義name為:send+qp\_message\_send.row\_id

Ex:

"name":"send763"

* 秘書定時推播則定義name為pushonly+qp\_message\_send\_pushonly.row\_id

Ex:

"name":"pushonly763"

* 這樣的好處是

如果往後真的要覆蓋同一則預約推播的話, 仍然可以拿到上次發出的unique key, 可以覆蓋預約推播的時間

* **enabled**
  + 表示任务当前状态，布尔值，必须为true或false，创建任务时必须为true。
* **trigger**
  + 表示schedule任务的触发条件，当前只支持定时任务（single）与定期任务（periodical）。
    - **single**
      * 表示定时任务
        + time为必选项且格式为"YYYY-mm-dd HH:MM:SS“，如"2014-02-15 13:16:59"，不能为"2014-2-15 13:16:59"或"2014-12-15 13:16"，即日期时间格式必须完整。

日期格式務必轉為GMT+8時區

* + - * + **Push**
        + 表示推送內容

**Example Response**

**Success Response**

HTTP/1.1 200 CREATED

 Content-Type: application/json; charset=utf-8

{

    "schedule\_id":"0eac1b80-c2ac-4b69-948b-c65b34b96512",

    "name":"&#23450;时推送示例"    //长度最大255字节，数字、字母、下划线、汉字。

}

**Failed Response**

HTTP/1.1 400 BAD REQUEST

Content-Type: application/json; charset=utf-8

{

  "error":

  {

    "code":8400,

    "message":"error message"

  }

}

1. 台北環境

http://qplay.benq.com/qplayApi/public/v101/qplay/sendPushMessage?lang=en-us&XXXXXXXXX

1. Extra參數, 如果有填入, 會拿來取代推播時放入的extra節點, 在不同系統, 代表的意義不同, 不過都是key的定義, 用來收到推播後, 除了可以打開指定APP外, 還能開啟到特定頁面或是case, ex:ENS就傳入event id

### 5.14 刷新獲取信息的時間搓記(updateLastMessageTime)

#### 5.14.1接口描述

提供API接口, 讓APP可以直接更新qp\_register底下的last\_message\_time這個欄位, 避免每次都取回相同的信息

#### 5.14.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/updateLastMessageTime?lang=en-us

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  token值需要使用sha256加密, 加密時需要添加salt值, salt值為request header內的Signature-Time參數  $token = base64\_encode (hash\_hmac(‘sha256’, $Signature-Time, $token’, true)); |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| last\_update\_time | Required | int | unix timestamp型式, 共10碼  下回這支手機再訪問信息時, 就會抓這個時間點之後的時間 |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| token\_valid | NA | 0-1 | Integer | 本次使用的token的有效時間  格式為Unix時間搓記  2016年3月25日 下午 01:49:56 換算為1458884996 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | content | 0-1 | String | 設備唯一識別碼 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000914 | 帳號已被停權 |
| 000911 | uuid不存在 |

**Related Tables:**

qp\_session

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000907 | token過期 | 代表token已經超過使用期限, APP直接切換到登入頁面讓User重新登入 |
| 000908 | token已經失效 | 代表token不存在於系統中, APP直接切換到登入頁面讓User重新登入 |
| 000914 | 帳號已被停權 | 停留在原畫面並提示錯誤訊息, 除原錯誤訊息外, 請加上請聯絡系統管理員確認帳號停權問題, 點擊OK後關閉APP |
| 000911 | UUID不存在 | 代表手機傳送的UUID在系統中查找不到, 需要重新走註冊流程 |

Check point:

1. APP使用這個API時, 直接以uuid先查找註冊表qp\_register, 如果不存在, 則直接拋出000911, UUID不存在, 這種訊息, 因為表示這個APP的UUID可能有問題(重新安裝過之類的), APP端收到000911, 則需要重新走register() API, 重新註冊UUID
2. 如果有找到UUID, 則接下來找qp\_session這張表

**uuid找不到或是token已經失效, 拋出錯誤訊息, 000908**

uuid有找到, 則以last\_message\_time更新qp\_register.last\_message\_time

1. 接下來說明何時APP會來用這支API:

建議APP在關閉以下視窗時, 除了使用getMessageList()來拿訊息外, 拿完訊息後, 需要呼叫updateLastMessageTime更新當前的時間搓到qp\_session表

User選擇Skip歷史訊息一樣要呼叫updateLastMessageTime更新時間



### 5.15設備移除 (Unregister)

#### 5.15.1接口描述

透過此API進行設備移除, 透過uuid從QPlay Server反註冊

#### 5.15.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/unregister?lang=en-us
3. header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 調用這支API的設備unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| target\_uuid | Required | string | 準備要移除的設備unique id |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | content | 0-1 | String | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000911 | 帳號不存在 |

**Related Tables:**

qp\_register

qp\_user

qp\_push\_token

qp\_project

qp\_error\_code

### 5.16 ITportal串接 (qplay web root directory)

#### 5.16.1接口描述

透過這個API, 讓User可以透過ITPortal連到App Manager畫面申請App Key

#### 5.16.2接口訊息定義

1. 請求方法：

ITPortal要開放一個link, 連到QPlay端申請app key

規格如下

**1.Method:**

http get

**2.Url:**

<https://qplay.benq.com/qplay/public>

**測試時, 網址可以先設定到**

[**https://qplaydev.benq.com/qplay/public**](https://qplaydev.benq.com/qplay/public)

**3.http headers:**

domain:BenQ

id:AD帳號,  ex:steven.yan

password: **md5(HMAC-SHA256(AD帳號, signature-time))**

signature-time就是10位數的unix時間搓

符合規範的話, 就自動幫忙登入, 切換到項目管理畫面

**signature-time需符合10min內的需求, 不能跟當時時間差距超過10min**

萬用密碼:

1416b460d0f2262770b59f5f00a83e4f

**初期開發, 為方便測試, 如果password輸入以上字串就可以直接通過**

不符合規範的話

系統會直接導到<https://qplay.benq.com/qplay/public/auth/login>

成功則會直接進入app key manager畫面

### 5.17新增下載次數 (addDownloadHit)

#### 5.17.1接口描述

進行APP下載時, 可以透過此API回傳下載時的資訊

#### 5.17.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/addDownloadHit?lang=en-us
3. header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| uuid | Required | string | 調用這支API的設備unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| login\_id | Required | string | 登入者的AD帳號 |
| package\_name | Required | string | 填入要下載的app package name |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| uuid | content | 0-1 | String | 設備的unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |
| login\_id | content | 0-1 | String | AD帳號 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000911 | 帳號不存在 |

**Related Tables:**

qp\_app\_download\_hit

### 5.18內網檢查APP版本號(checkAppVersionIntra)

#### 5.18.1接口描述

取得APP最新版號資訊, 並由Server主動確認是否需要更新

#### 5.18.2接口訊息定義

1. 請求方法：
2. GET /v101/qplay/checkAppVersionIntra?lang=en-us&package\_name=com.qplay.appqplay&device\_type=android&version\_code=1

header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱, 此專案名稱為 appqplay |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) )  此專案的AppSecretKey 為swexuc453refebraXecujeruBraqAc4e |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| package\_name | Required | string | app的package name |
| device\_type | Required | string | ios  android  其中一個 |
| vension\_code | Required | string | 填入目前版號 |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼  resul\_code=1, 表示需要更新  result\_code =000913, 表示不需要更新 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |
| version\_code | content | 0-1 | String | 新版本版號 |
| download\_url | content | 0-1 | String | 直接提供iOS或是Android的下載路徑 |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **回應碼：**

|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000913 | 不需更新 |

**Related Tables:**

qp\_app\_head

qp\_app\_version

qp\_error\_code

**錯誤碼處理:**

|  |  |  |
| --- | --- | --- |
| 錯誤碼 | 錯誤描述 | 處理 |
| 000913 | 不需更新 | 收到此訊息app不需另外處理, 代表APP是新版本, 直接進行下一個步驟即可 |

範例:

假設呼叫內部網址:

<https://qplayintra.benq.corp.com/OTAtest/public/v101/qplay/checkAppVersionIntra?lang=en-us&package_name=com.qplay.appyellowpagetest&device_type=android&version_code=1>

會收到以下的response

{

    "result\_code": 1,

    "message": "Need to update",

    "content": {

        "version\_code": 370,

        "download\_url": "<http://qplayintra.benq.corp.com/OTAtest/public/app/2/apk/android/370/appyellowpage.apk>"

    }

}

因為

詢問的是內網的staging環境

且使用checkAppVersionIntra這個API

所以download\_url會拿到<http://qplayintra.benq.corp.com/OTAtest/>  開頭的網址

其中

OTA

OTAtest

OTAdev

是跟著開發, 測試,正式環境跑的 (目前正式環境尚未開放, dev跟test已經開放使用)

假設APP使用的是外部網路,  仍舊使用原來的checkAppVersion API

詢問的網址會像以下這樣

<http://qplaytest.benq.com/qplayApi/public/v101/qplay/checkAppVersion?lang=en-us&package_name=com.qplay.appyellowpagetest&device_type=android&version_code=1>

得到的就是外部網址:

{

    "result\_code": 1,

    "message": "Need to update",

    "content": {

        "version\_code": 370,

        "download\_url": "<https://qplaytest.benq.com/qplay/public/app/2/apk/android/370/appyellowpage.apk>"

    }

}

### 5.19新增APP行為Log (addAppLog)

#### 5.19.1接口描述

透過此接口, 讓APP可以將log送到後台, 供未來大數據分析

#### 5.19.2接口訊息定義

1. 請求方法：
2. POST /v101/qplay/addAppLog?lang=en-us
3. header請求參數:

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，ex.使用POST方法傳輸, 類型需為application/json |
| app-key | 必填 | 專案名稱 |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | Base64( HMAC-SHA256( SignatureTime , YourAppSecretKey ) ) |

1. **網址列請求參數：**

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| lang | Required | string | 請填入 語系-國家  ex:  en-us  zh-tw  zh-cn |
| uuid | Optional | string | 調用這支API的設備unique id  uuid:  android:設備上的ANDROID\_ID  iOS:APNS所提供的DeviceToken  ex:  apple可能是6974ac11 870e09fa 00e2238e 8cfafc7d 2052e342 182f5b57 fabca445 42b72e1b  android可能是9774d56d682e549c |

POST Body結構

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **參數名稱** | **父節點** | **是否必填** | **資料類型** | **描述** |
| login\_id | NA | Required | string | 登入者的AD帳號 |
| package\_name | NA | Required | string | 填入APP的app package name |
| log\_list | NA | Required | Container | 底下包一組以上的陣列 |
| page\_name | log\_list | Optional | string | 請填入APP端自行定義的畫面名稱, 對Server端沒有意義  由APP自行定義每支APP對應每個頁面的naming rule及其背後的涵義 |
| page\_action | log\_list | Optional | string | 請填入APP端自行定義的畫面名稱, 對Server端沒有意義  由APP自行定義每支APP對應每個頁面裡的每個動作的naming rule及其背後的涵義 |
| start\_time | log\_list | Optional | int | 請提供unix時間搓記,  單位到秒  ex: 2016年3月25日 下午 01:49:56 換算為1458884996 |
| period | log\_list | Optional | float | 每個動作或是每個畫面停留的時間  如果說page\_action定義出來的欄位意義代表著次數, 那period欄位也可以不填 |
| device\_type | log\_list | Required | string | ios/android |
| latitude | log\_list | Optional | string | GPS的緯度座標 |
| longitude | log\_list | Optional | string | GPS的經度座標 |
| county | log\_list | Optional | string | 發送所在國家, 非必填 |
| city | log\_list | Optional | string | 發送所在城市, 非必填 |
| attribute1 | log\_list | Optional | string | 其他訊息  APP端自行定義 |
| attribute2 | log\_list | Optional | string | 其他訊息  APP端自行定義 |
| attribute3 | log\_list | Optional | string | 其他訊息  APP端自行定義 |
| attribute4 | log\_list | Optional | string | 其他訊息  APP端自行定義 |
| attribute5 | log\_list | Optional | string | 其他訊息  APP端自行定義 |

1. **回應訊息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| result\_code | NA | 1 | String | 回應代碼 |
| message | NA | 1 | String | 回應訊息描述 |
| content | NA | 0-1 | Container | 回應訊息內容Container |

P.S如果result\_code為1, 則會帶content, 反之, 不帶content

1. **POST Body請求參數**

請按照以下格式送出資訊, 一定要是Json格式:

{

"login\_id":"Steven.Yan",

"package\_name":"com.qplay.appyellowpage",

"log\_list":[

{

"page\_name":"main",

"page\_action":"query",

"period":"1.003",

"device\_type":"ios",

"latitude":"",

"longitude":"",

"county":"TW",

"city":"Taipei",

"attribute1":"",

"attribute2":"",

"attribute3" :"",

"attribute4" :"",

"attribute5" :""

},

{

"page\_name":"main",

"page\_action":"query",

"period":"1.003",

"device\_type":"ios",

"latitude":"",

"longitude":"",

"county":"TW",

"city":"Taipei",

"attribute1":"",

"attribute2":"",

"attribute3" :"",

"attribute4" :"",

"attribute5" :""

}]

}

**Check Point:**

1. 採用http post方式
2. http header裡面的app-key, 送出時請以各APP送出, ex:appqplay, appyellowpage

對應的signature, 也需以各自的secretkey去加密

**請勿全部以appqplay來送**

1. 所有欄位皆由APP端送出並加以記錄, 除了IP欄位由Server端透過http header自行處理
2. 其餘欄位直接接收APP端傳來的資料, 保存進DB, Server端並無任何邏輯確認
3. 如同GA, API中的page\_name, page\_action則由APP端PM自行定義, 甚麼APP該有哪些頁面, 頁面上該有哪些動作, 命名規則皆是由App端定義
4. 必填非必填欄位請參考上面的表格送出
5. **回應碼：**

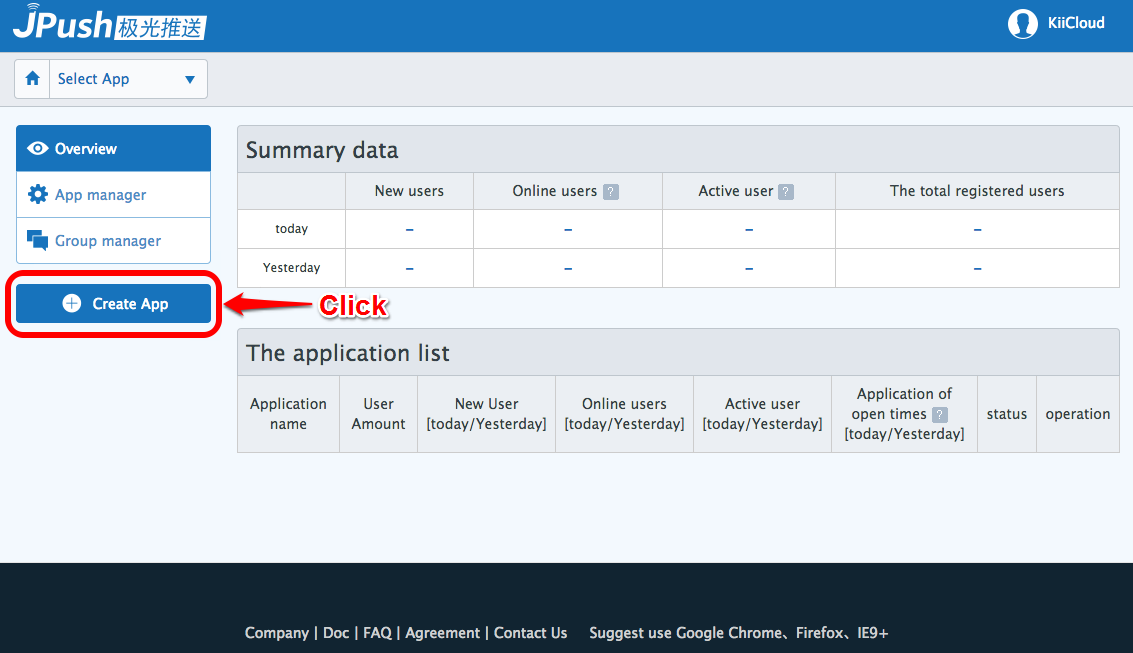
|  |  |
| --- | --- |
| **Result Code** | **Description** |
| 1 | 詳第3章節之通用回應代碼之定義 |
| 999xxx | 詳第3章節之通用回應代碼之定義 |
| 000911 | 帳號不存在 |
| 000907 | token過期 |
| 000908 | token已經失效 |
| 000909 | App Key不存在 |
| 000914 | 帳號已被停權 |
| 000915 | package name不存在 |

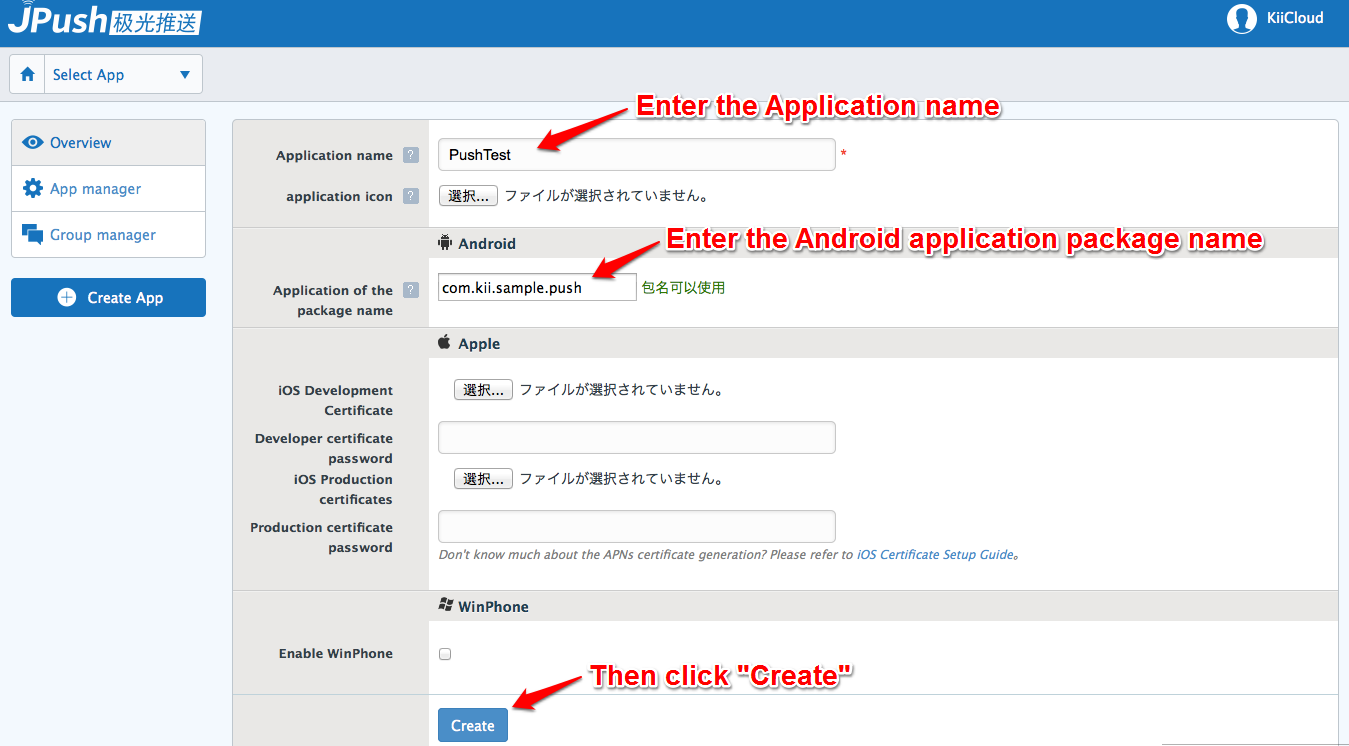
**Related Tables:**

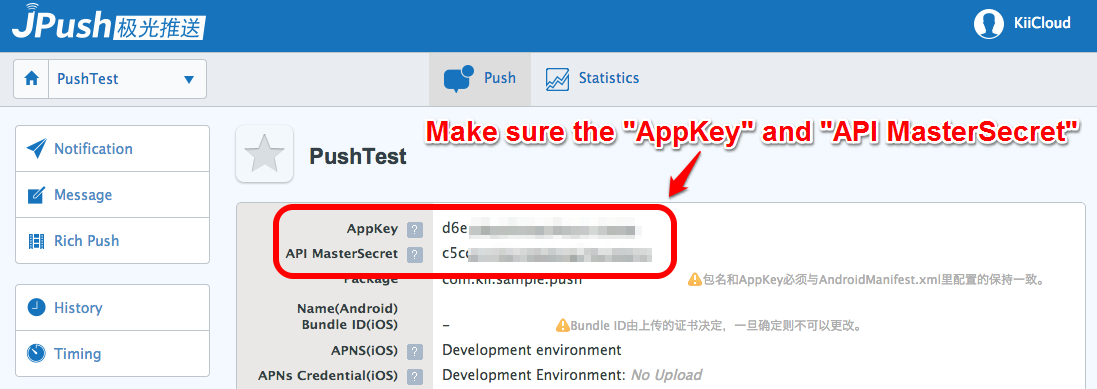
qp\_app\_log

# Appendix

Android: APP Key & Master Secret

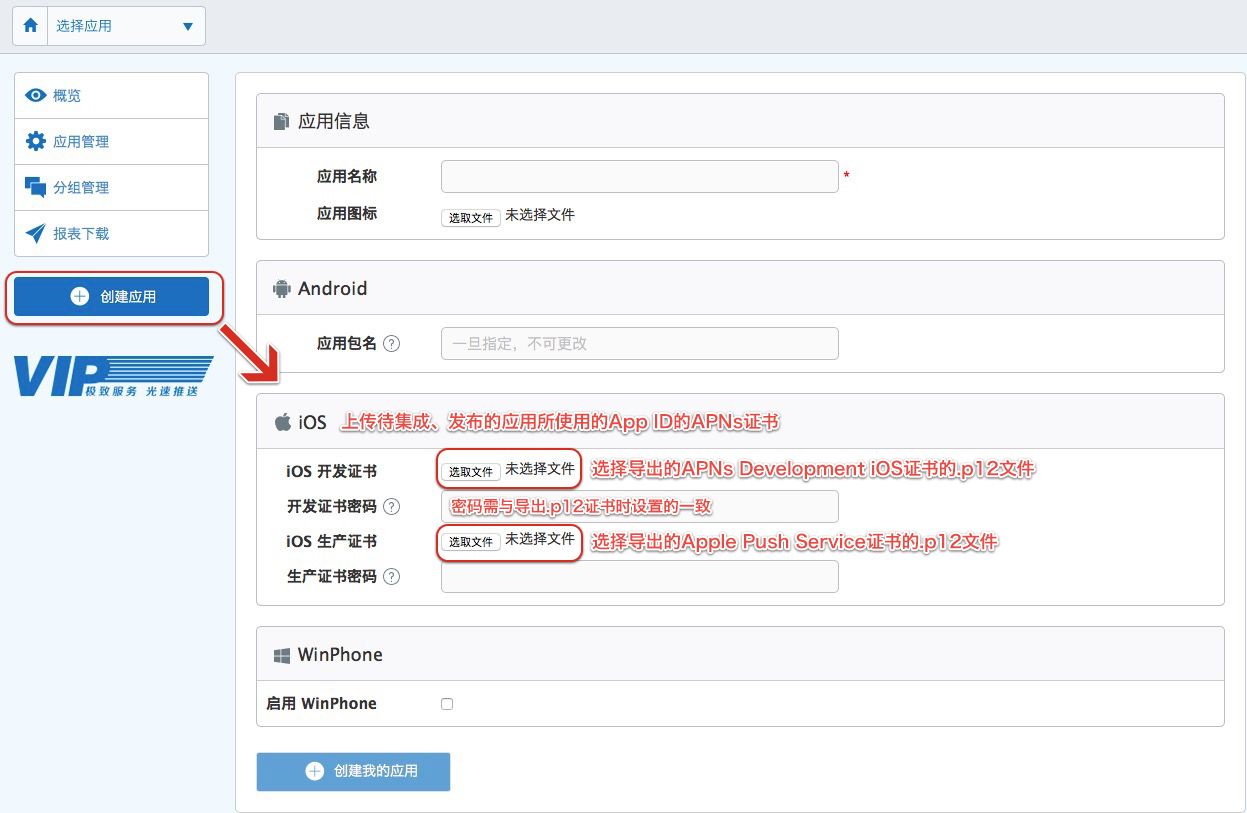
[](http://docs.kii.com/cn/samples/push-notifications/push-notifications-jpush/create-project/01.png)

[](http://docs.kii.com/cn/samples/push-notifications/push-notifications-jpush/create-project/02.png)

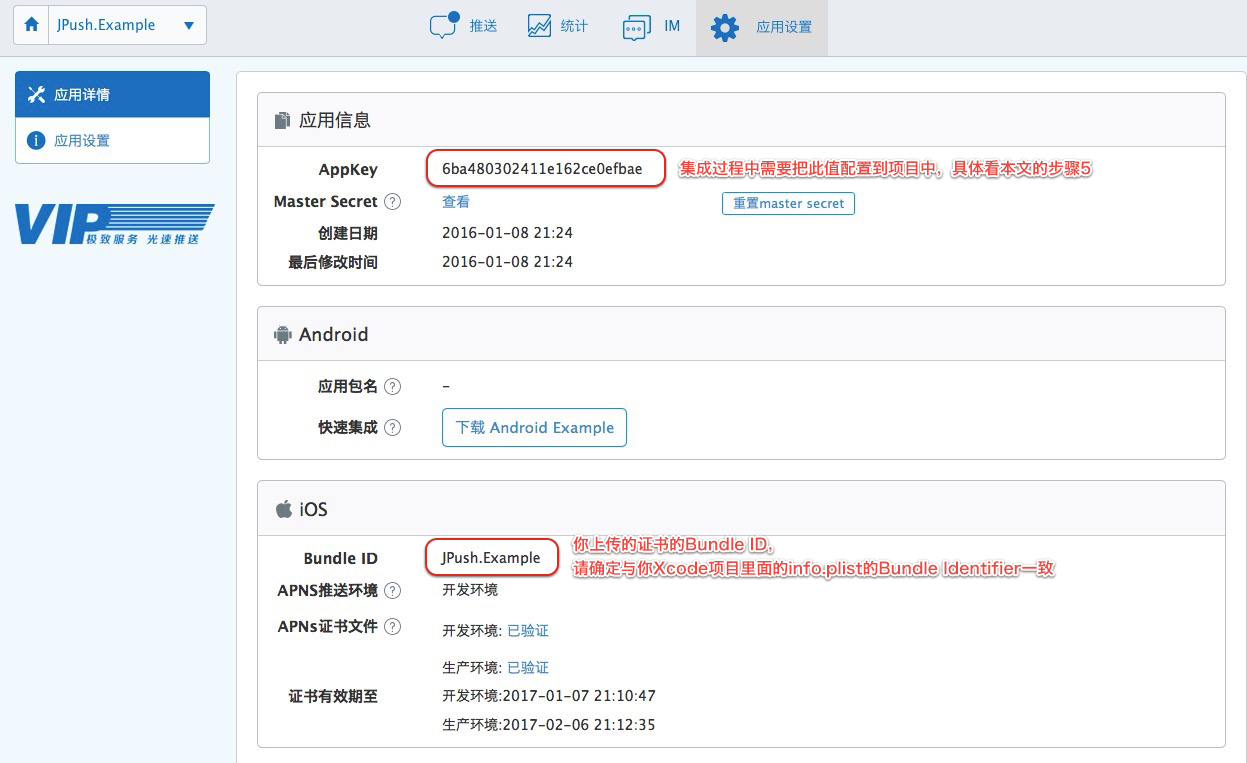
[](http://docs.kii.com/cn/samples/push-notifications/push-notifications-jpush/create-project/03.png)

iOS在: APP Key & Master Secret

JPush的管理Portal 上创建应用并上传APNs证书。如果对APNs证书不太了解 请参考： [iOS 证书设置指南](http://docs.jpush.io/client/ios_tutorials/#ios_1)



* 创建成功后自动生成 AppKey 用以标识该应用。



# A1.Push API v3

### http://docs.jiguang.cn/server/rest\_api\_v3\_push/

这是 Push API 最近的版本。

相比于 API v2 版本，v3 版本的改进为：

* 完全基于 https，不再提供 http 访问；
* 使用 HTTP Basic Authentication 的方式做访问授权。这样整个 API 请求可以使用常见的 HTTP 工具来完成，比如：curl，浏览器插件等；
* 推送内容完全使用 JSON 的格式；
* 支持的功能有所改进：支持多 tag 的与或操作；可单独发送通知或者自定义消息，也可同时推送通知与自定义消息；windows phone 目前只有通知。

### 推送概述

#### 功能说明

向某单个设备或者某设备列表推送一条通知、或者消息。

推送的内容只能是 JSON 表示的一个推送对象。

调用地址：  
POST <https://api.jpush.cn/v3/push>

#### 请求示例

curl --insecure -X POST -v https://api.jpush.cn/v3/push -H "Content-Type: application/json" -u "7d431e42dfa6a6d693ac2d04:5e987ac6d2e04d95a9d8f0d1" -d '{"platform":"all","audience":"all","notification":{"alert":"Hi,JPush !","android":{"extras":{"android-key1":"android-value1"}},"ios":{"sound":"sound.caf","badge":"+1","extras":{"ios-key1":"ios-value1"}}}}'

> POST /v3/push HTTP/1.1

> Authorization: Basic N2Q0MzFlNDJkZmE2YTZkNjkzYWMyZDA0OjVlOTg3YWM2ZDJlMDRkOTVhOWQ4ZjBkMQ==

#### 返回示例

< HTTP/1.1 200 OK

< Content-Type: application/json

{"sendno":"18","msg\_id":"1828256757"}

#### 调用验证

HTTP Header（头）里加一个字段（Key/Value对）：

Authorization: Basic base64\_auth\_string

其中 base64\_auth\_string 的生成算法为：base64(appKey:masterSecret)

即，对 appKey 加上冒号，加上 masterSecret 拼装起来的字符串，再做 base64 转换

### 推送校验 API

POST <https://api.jpush.cn/v3/push/validate>

#### 功能说明

该 API 只用于验证推送调用是否能够成功，与推送 API 的区别在于：不向用户发送任何消息。 其他字段说明：同推送 API。

### 推送对象

一个推送对象，以 JSON 格式表达，表示一条推送相关的所有信息。

##### 示例与说明

{

"platform": "all",

"audience": {

"tag": [

"深圳",

"北京"

]

},

"notification": {

"android": {

"alert": "Hi, JPush!",

"title": "Send to Android",

"builder\_id": 1,

"extras": {

"newsid": 321

}

},

"ios": {

"alert": "Hi, JPush!",

"sound": "default",

"badge": "+1",

"extras": {

"newsid": 321

}

}

},

"message": {

"msg\_content": "Hi,JPush",

"content\_type": "text",

"title": "msg",

"extras": {

"key": "value"

}

},

"sms\_message":{

"content":"sms msg content",

"delay\_time":3600

},

"options": {

"time\_to\_live": 60,

"apns\_production": false

}

}

|  |  |  |
| --- | --- | --- |
| **关键字** | **选项** | **含义** |
| platform | 必填 | 推送平台设置 |
| audience | 必填 | 推送设备指定 |
| notification | 可选 | 通知内容体。是被推送到客户端的内容。与 message 一起二者必须有其一，可以二者并存 |
| message | 可选 | 消息内容体。是被推送到客户端的内容。与 notification 一起二者必须有其一，可以二者并存 |
| sms\_message | 可选 | 短信业务功能 |
| options | 可选 | 推送参数 |

#### android

Android 平台上的通知。

被 JPush SDK 按照一定的通知栏样式展示。

支持的字段有：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **关键字** | **类型** | **选项** | **含义** | **说明** |
| alert | string | 必填 | 通知内容 | 这里指定了，则会覆盖上级统一指定的 alert 信息；内容可以为空字符串，则表示不展示到通知栏。 |
| title | string | 可选 | 通知标题 | 如果指定了，则通知里原来展示 App名称的地方，将展示成这个字段。 |
| builder\_id | int | 可选 | 通知栏样式ID | Android SDK 可设置通知栏样式，这里根据样式 ID 来指定该使用哪套样式。 |
| extras | JSON Object | 可选 | 扩展字段 | 这里自定义 JSON 格式的 Key/Value 信息，以供业务使用。 |

{

"notification" : {

"android" : {

"alert" : "hello, JPush!",

"title" : "JPush test",

"builder\_id" : 3,

"extras" : {

"news\_id" : 134,

"my\_key" : "a value"

}

}

}

}

#### iOS

iOS 平台上 APNs 通知。

该通知内容会由 JPush 代理发往 Apple APNs 服务器，并在 iOS 设备上在系统通知的方式呈现。

该通知内容满足 APNs 的规范，支持的字段如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **关键字** | **类型** | **选项** | **含义** | **说明** |
| alert | string | 必填 | 通知内容 | 这里指定了，将会覆盖上级统一指定的 alert 信息；内容为空则不展示到通知栏。支持 emoji 表情。 |
| sound | string | 可选 | 通知提示声音 | 如果无此字段，则此消息无声音提示；有此字段，如果找到了指定的声音就播放该声音，否则播放默认声音,如果此字段为空字符串，iOS 7 为默认声音，iOS 8 为无声音。(消息) 说明：JPush 官方 API Library (SDK) 会默认填充声音字段。提供另外的方法关闭声音。 |
| badge | int | 可选 | 应用角标 | 如果不填，表示不改变角标数字；否则把角标数字改为指定的数字；为 0 表示清除。JPush 官方 API Library(SDK) 会默认填充badge值为"+1",详情参考：[badge +1](http://blog.jpush.cn/ios_apns_badge_plus/) |
| content-available | boolean | 可选 | 推送唤醒 | 推送的时候携带"content-available":true 说明是 Background Remote Notification，如果不携带此字段则是普通的Remote Notification。详情参考：[Background Remote Notification](http://docs.jiguang.cn/client/ios_tutorials/#ios-7-background-remote-notification) |
| category | string | 可选 |  | IOS8才支持。设置APNs payload中的"category"字段值 |
| extras | JSON Object | 可选 | 扩展字段 | 这里自定义 Key/value 信息，以供业务使用。 |

iOS 通知 JPush 要转发给 APNs 服务器。APNs 协议定义通知长度为 2048 字节。

JPush 因为需要重新组包，并且考虑一点安全冗余，要求"iOS":{ } 及大括号内的总体长度不超过：2000 个字节。

另外，JPush 在推送时使用 utf-8 编码，所以一个汉字占用 3 个字节长度。

**服务端发送消息串**

{

"notification" : {

     "ios" : {

             "alert" : "hello, JPush!",

             "sound" : "sound.caf",

             "badge" : 1,

             "extras" : {

                  "news\_id" : 134,

                  "my\_key" : "a value"

             }

        }

}

}

### 调用返回

##### HTTP 状态码

参考文档：[HTTP-Status-Code](http://docs.jiguang.cn/server/http_status_code/)

##### 业务返回码

|  |  |  |  |
| --- | --- | --- | --- |
| **Code** | **描述** | **详细解释** | **HTTP Status Code** |
| 1000 | 系统内部错误 | 服务器端内部逻辑错误，请稍后重试。 | 500 |
| 1001 | 只支持 HTTP Post 方法 | 不支持 Get 方法。 | 405 |
| 1002 | 缺少了必须的参数 | 必须改正 | 400 |
| 1003 | 参数值不合法 | 必须改正 | 400 |
| 1004 | 验证失败 | 必须改正。详情请看：[调用验证](http://docs.jiguang.cn/server/rest_api_v3_push/#_5) | 401 |
| 1005 | 消息体太大 | 必须改正。 Android平台Notification+Message长度限制为4000字节； iOS Notification 中 “iOS”:{ } 及大括号内的总体长度不超过：2000个字节（包括自定义参数和符号），iOS 的 Message部分长度不超过 4000 字节； WinPhone平台Notification长度限制为4000字节 | 400 |
| 1008 | app\_key参数非法 | 必须改正 | 400 |
| 1011 | 没有满足条件的推送目标 | 请检查audience | 400 |
| 1020 | 只支持 HTTPS 请求 | 必须改正 | 404 |
| 1030 | 内部服务超时 | 稍后重试 | 503 |

### HTTP状态码定义

JPush Push API v3 新版本 API 满足此规范。JPush Report API 也满足此规范。

|  |  |  |
| --- | --- | --- |
| **Code** | **描述** | **详细解释** |
| 200 | OK | Success! |
| 400 | 错误的请求 | 该请求是无效的。相应的描述信息会说明原因。 |
| 401 | 未验证 | 没有验证信息或者验证失败 |
| 403 | 被拒绝 | 理解该请求，但不被接受。相应的描述信息会说明原因。 |
| 404 | 无法找到 | 资源不存在，请求的用户的不存在，请求的格式不被支持。 |
| 405 | 请求方法不合适 | 该接口不支持该方法的请求。 |
| 410 | 已下线 | 请求的资源已下线。请参考相关公告。 |
| 429 | 过多的请求 | 请求超出了频率限制。相应的描述信息会解释具体的原因。 |
| 500 | 内部服务错误 | 服务器内部出错了。请联系我们尽快解决问题。 |
| 502 | 无效代理 | 业务服务器下线了或者正在升级。请稍后重试。 |
| 503 | 服务暂时失效 | 服务器无法响应请求。请稍后重试。 |
| 504 | 代理超时 | 服务器在运行，但是无法响应请求。请稍后重试。 |

**遵守的规范**

* 200 一定是正确。所有异常都不使用 200 返回码
* 业务逻辑上的错误，有特别的错误码尽量使用 4xx，否则使用 400。
* 服务器端内部错误，无特别错误码使用 500。
* 业务异常时，返回内容使用 JSON 格式定义 error 信息。

频率控制基于 AppKey 来定义，每个 AppKey 有一个基础的调用频率限制数量。免费版本如下表：

|  |  |
| --- | --- |
| **API 类型** | **频率（次/分钟）** |
| [Push API v3](http://docs.jiguang.cn/server/rest_api_v3_push) | 600 |
| [Report-API](http://docs.jiguang.cn/server/rest_api_v3_report) | 2400 |
| [Device-API](http://docs.jiguang.cn/server/rest_api_v3_device) | 600 |

#### HTTP 请求与响应

##### 响应头

所有的 HTTP API Response Header 里都加了三项频率控制信息：

* X-Rate-Limit-Quota：当前 AppKey 一个时间窗口内可调用次数
* X-Rate-Limit-Remaining：当前时间窗口剩余的可用次数
* X-Rate-Limit-Reset：距离时间窗口重置剩余的秒数

##### 超出频率限制

当一个请求遇到频率限制时，JPush API 返回的 HTTP 返回码为 429，其含义是：太多的请求。

此时返回内容里，是如下的信息：

{

  "error": {

       "code": 2002,

       "message": "Rate limit exceeded"

   }

}

##### 避免遭遇频率限制的建议

* 均匀地分布请求到各时间窗口
* 根据 alias 大量请求时，避免无效的 alias。
* 如果大量针对 alias 的请求内容一致，可以每次 API + 调用填写多个 alias 。具体请参考推送 API 说明。

##### 频率控制 - 推送场景举例

A, B 两应用都有千万的用户量。

A 应用推送方式，主要是群发

一次推送（即一次 API 调用）面向大量用户。这种使用广播，或者基于 Tag 做推送。

这种情况下，由于群发几乎不可能短期内推送很多次，从而几乎不可能触发 API 频率控制。

进一步地说，API 频率控制与推送送达的终端用户数无关。一次 API 广播推送，可以送达所有用户。

B 应用推送方式，主要是单点

一次推送（即一次 API 调用）只面向一个用户。这种基于 alias 或者 registrationI 做推送。

由于用户量大，向每个用户推送都需要调用一次 API，则相对比较容易触发到频率控制。

有一个技巧可以减少 API 调用次数：推送 API 支持向 alias, registrationId 推送时，最多一次可以向 1000 个目标对象推送。也就是把推送请求批量化。

但这个作法不好的是：1 批量一起推送到很多对象，需要统一时间；2 批量一起的推送，其内容体只能一样，不能根据每个对象不同。

# A2.JPush API client library for PHP

<http://docs.jiguang.cn/server/php_sdk/>

github source code:

<https://github.com/jpush/jpush-api-php-client/blob/master/examples/push_example.php>

|  |
| --- |
| <?php |
| /\*\* |
| \* 该示例主要为JPush Push API的调用示例 |
| \* HTTP API文档:http://docs.jpush.io/server/rest\_api\_v3\_push/ |
| \* PHP API文档:https://github.com/jpush/jpush-api-php-client/blob/master/doc/api.md#push-api--构建推送pushpayload |
| \*/ |
| ini\_set("display\_errors", "On"); |
| error\_reporting(E\_ALL | E\_STRICT); |
| require\_once("../src/JPush/JPush.php"); |
|  |
| $br = '<br/>'; |
| $app\_key = 'dd1066407b044738b6479275'; |
| $master\_secret = 'e8cc9a76d5b7a580859bcfa7'; |
|  |
| // 初始化 |
| $client = new JPush($app\_key, $master\_secret); |
|  |
| // 简单推送示例 |
| $result = $client->push() |
| ->setPlatform('all') |
| ->addAllAudience() |
| ->setNotificationAlert('Hi, JPush') |
| ->send(); |
|  |
| echo 'Result=' . json\_encode($result) . $br; |
|  |
| // 完整的推送示例,包含指定Platform,指定Alias,Tag,指定iOS,Android notification,指定Message等 |
| $result = $client->push() |
| ->setPlatform(array('ios', 'android')) |
| ->addAlias('alias1') |
| ->addTag(array('tag1', 'tag2')) |
| ->setNotificationAlert('Hi, JPush') |
| ->addAndroidNotification('Hi, android notification', 'notification title', 1, array("key1"=>"value1", "key2"=>"value2")) |
| ->addIosNotification("Hi, iOS notification", 'iOS sound', JPush::DISABLE\_BADGE, true, 'iOS category', array("key1"=>"value1", "key2"=>"value2")) |
| ->setMessage("msg content", 'msg title', 'type', array("key1"=>"value1", "key2"=>"value2")) |
| ->setOptions(100000, 3600, null, false) |
| ->send(); |
|  |
| echo 'Result=' . json\_encode($result) . $br; |
|  |
|  |
| // 指定推送短信示例(推送未送达的情况下进行短信送达, 该功能需预付短信费用, 并调用Device API绑定设备与手机号) |
| $result = $client->push() |
| ->setPlatform('all') |
| ->addTag('tag1') |
| ->setNotificationAlert("Hi, JPush SMS") |
| ->setSmsMessage('Hi, JPush SMS', 60) |
| ->send(); |
|  |
| echo 'Result=' . json\_encode($result) . $br; |

# A3. iOS SDK 集成指南

<http://docs.jiguang.cn/guideline/ios_guide/>

# A4. Android SDK 集成指南

<http://docs.jiguang.cn/guideline/android_guide/>

# A5.Android API

<http://docs.jiguang.cn/client/android_api/>

包含

## 停止与恢复推送服务 API

## 接收推送消息Receiver

## 别名与标签 API

## 获取 RegistrationID API

# A6. iOS API

<http://docs.jiguang.cn/client/ios_api/>

包含:

## 标签与别名 API (iOS)

## 获取 APNs（通知） 推送内容

## 获取自定义消息推送内容

## 获取 RegistrationID

## 设置Badge

## 地理位置统计