|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Subject | QPlay Custom API範例 | | Doc.: |  | | Rev. | V1.1.2 |
|  | | | Release Date: | 2017/02/16 | | Revision Status: | |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  | | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  |  | | | | |  |  |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| Approved By: | | Reviewed By: | | | Prepared By: | | |
| PM Head / | | SD / | | | PM /Steven.Yan | | |
|  | | QJ / | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |
|  | |  | | |  | | |

[1.0 Change History 3](#_Toc475020875)

[2.0 Introduction 4](#_Toc475020876)

[2.1 System Description 4](#_Toc475020877)

[2.2 Purpose & Scope 4](#_Toc475020878)

[3.0 Function Description: 5](#_Toc475020879)

[3.1 AddMyPhoneBook功能名稱與描述 8](#_Toc475020880)

[4.0 Signature 15](#_Toc475020881)

[5.0 Error Code List 16](#_Toc475020882)

[6.0 Others 19](#_Toc475020883)

# Change History

As to the revised document: the latest revised content is written in blue font, while the yellow background means the waiting confirm status. And the double strikethrough represents the deleting content.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Comment |
| 2016-07-07 | Steven Yan | 1.0.0 | Initial |
| 2016-11-07 | Steven Yan | 1.0.7 | 修改內容為yellowpage custom api範例 |
| 2016-11-24 | Steven Yan | 1.0.8 | 調整http response, 不需再提供d這個參數  直接回應整包json即可 |
| 2016-11-29 | Steven Yan | 1.0.9 | 隱藏加密演算法 |
| 2017-01-06 | Steven Yan | 1.0.10 | 提供Error Code章節 |
| 2017-01-23 | Steven Yan | 1.1.0 | 修改格式, 並補充signature章節 |
| 2017-01-24 | Steven Yan | 1.1.1 | 3.1.3, 3.1.4補充htmlencode, post body在送出前, 需要先htmlencode後再送出, 避免誤用xml保留字元, 造成程式失敗 |
| 2017-02-16 | Steven Yan | 1.1.2 | 5.0章節新增999015, 999016錯誤碼, 其中999016為呼叫custom API時會拋出的錯誤, 如果是URL填寫錯誤, 則會拋出999016, 請RD重新檢視URL填寫是否正確, 包含version,appkey,action |

# Introduction

## System Description

請填入系統描述

提供User查詢公司員工的連絡資訊

## Purpose & Scope

請填入專案目的及範疇

能夠過手機查詢員工電話並加入常用電話簿

# Function Description:

請在此章節填入APP及所有API的基本資料

*首先先填寫專案基本資訊*

此專案的appKey=**xxxxx**

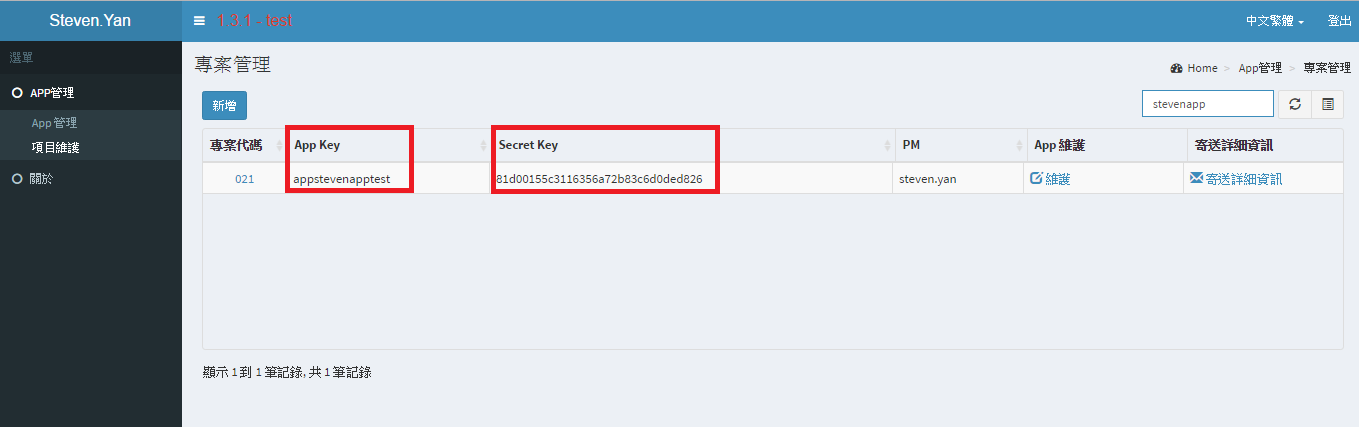
secretKey為**xxxxxxx**

**紅字處請放入透過IT Portal所申請的appKey及secret key**

Appkey在呼叫Custom API時, 需在網址列及http header中填入

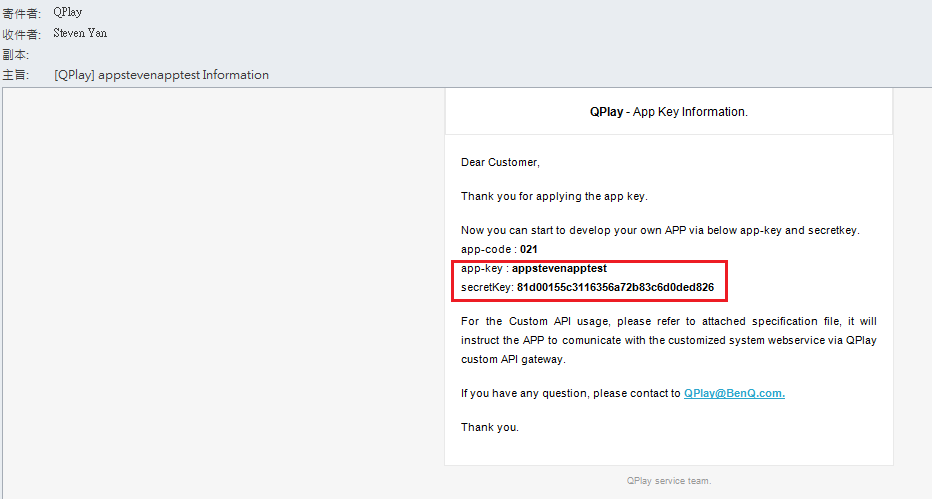
secretKey則是使用在4.0章節, Signature加密, 同樣需加密後放在http header

如下為ITPortal連到專案管理畫面中取得appkey的sample:



申請時, 會主動寄發Email給申請者及專案歸屬PM, 如果Email遺失,

也可以重新回到上圖頁面, 點擊寄送詳細資訊後, 從Email中獲取資訊



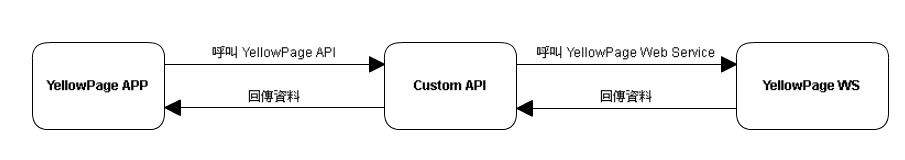
*填寫完專案基本資訊後, 接著要填寫API的清單, 以下以yellowpage為範例*

* Custom API名稱列表與描述

*請列出所有會用上的Custom API, 以下為yellowpage的範例, 共會開發6支web service供QPlay使用*

|  |  |
| --- | --- |
| **Custom API名稱** | **描述** |
| QueryEmployeeData | 在查詢頁面輸入查詢條件後，透過API回傳查詢結果。 |
| QueryEmployeeDataDetail | 在查詢結果頁面和我的電話簿頁面面點選詳細資料後，透過API回傳詳細資料。 |
| AddMyPhoneBook | 詳細資料頁面點擊空白星號後，透過API回傳新增成功訊息。 |
| DeleteMyPhoneBook | 詳細資料頁面點擊黃色星號後，透過API回傳刪除成功訊息。 |
| QueryMyPhoneBook | 在查詢頁面點擊我的電話簿後，透過API回傳我的電話簿資料。 |
| QueryCompanyData | 在查詢頁面公司的下拉選單資料，透過API回傳Company項目 |

* APP、Custom API和WebService關係，如下圖



**API Gateway只擔任轉送資訊的角色,**

**因此, app端將所有參數組成合適的json格式, 透過http request body送出到QPlay的API gateway,**

**QPlay加上特定格式後轉送到custom system的web service內**

**QPlay收到web service的response後, 解析json, 並加入message及token\_valid兩個節點後,**

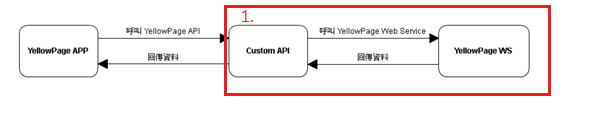
**返回給app**

*從3.1章節開始就需要填寫詳細的SPEC內容, 供API開發用, 以下僅以yellowpage六支API中的AddMyPhoneBook當範例說明:*

*從上圖看, SPEC撰寫需分為兩大區塊:*

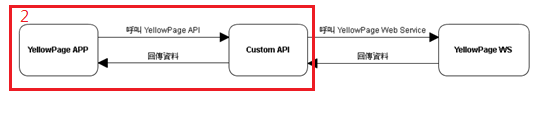
1. *外部系統(yellowpage)的web service規格:*

*Yellowpage WebService處理Business Logic部分, 需明確定義輸出入參數, 針對輸入參數回應正確的回應碼及輸出結果, 此部分的規格書應撰寫於3.X.3章節*

**

1. *QPlay Custom API gateway的規格:*

*Custom API用來接收App端發送的訊息, 並修改格式提交到外部系統(yellowpage), 收到回應後加入message及valid\_token回傳App Client, 此部分的規格書應撰寫於3.X.4章節*



## AddMyPhoneBook功能名稱與描述

### Description

請填入功能描述

透過API新增常用聯絡人至PhoneBook上

### UI

如果有必要, 請放入此功能的相關畫面及控制項



### Webservice

*外部系統的Web Service內的企業邏輯請填寫在這個區塊, 這個章節的規格書請提供給外部系統Web Service開發的RD*

Yellowpage webservice的Input/Output參數定義及傳輸範例

1. Input Parameter

請填入企業邏輯的Input參數

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| User\_EmpID | Required | string | 使用者的員工工號 |
| Add\_EmpID | Required | string | 被加入的員工工號 |
| Add\_Company | Required | string | 被加入的公司名稱 |

1. Output Parameter

請填入企業邏輯的output參數

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| ResultCode | NA | 1 | Integer | 回應碼 |
| Content | NA | 0-1 | Container | 回應訊息內容Container  裡面放web service想要提供的各種資訊  由各外部系統自行定義 |

1. 傳輸規格範例：

*請填入傳輸時所需的http協定資訊*

Name: YellowpageForQplayAPI

Function: AddMyPhoneBook

URL:http://xxxx.xxxx.xxxxx/xxxxxx/YellowpageForQplayAPI.asmx/AddMyPhoneBook*請填入webservice URL*

Method: HTTP POST *一定要是http post / https post*

Http request headers:

Content-Type=application/json

Http request body:

{"strXml":"<LayoutHeader><User\_EmpID>0407731</User\_EmpID><Add\_EmpID>1207952</Add\_EmpID><Add\_Company>BenQ</Add\_Company></LayoutHeader>"}

Http response headers:

**Cache-Control →**private, max-age=0

**Content-Length →**35

**Content-Type →**application/json; charset=utf-8

**Date →**Mon, 11 Jul 2016 07:14:04 GMT

**Server →**Microsoft-IIS/7.5

**X-AspNet-Version →**4.0.30319

**X-Powered-By →**ASP.NET

Http response body:

{

"d": "{\"resultcode\":\"001902\"}"

}

1. 範例說明:

*請說明第三點所提及的http協定資訊*

* http request header:

*必要參數: content-type必定為application/json, QPlay custom API呼叫webservice時只會傳json格式*

Content-Type=application/json

* http request body:

*Request Body內容一定是json格式, 最外層一定要包strXml這個變數, 名稱大小寫皆不可更改, 也就是QPlay Custom API呼叫webservice, 為了要讓webservice識別, 以strXml為變數, 後面的value接的即是webservice企業所需的input參數,且參數是以XML型態組成*

{"strXml":"<LayoutHeader><User\_EmpID>0407731</User\_EmpID><Add\_EmpID>1207952</Add\_EmpID><Add\_Company>BenQ</Add\_Company></LayoutHeader>"}

因為內文實質上是XML格式, 因此XML保留字元請先做HTMLENCODE

以下5個符號在http post之前需要先轉換為右邊格式

|  |  |
| --- | --- |
| 原始字元 | 轉換字元 |
| < | &lt; |
| > | &gt; |
| & | &amp; |
| ‘ | &apos; |
| “ | &quot; |

Ex: FlowER & IP Servers這樣的內文, 需轉換為FlowER &amp; IP Servers後再送出

* http response body:ResultCode

*外部系統的web service回傳值一定為json格式, 帶了整個web service處理完後的結果*

*一定要帶回ResultCode這個節點, 如下:*

*且外接系統的web service只會回傳ResultCode簡碼, 並不會回傳具體錯誤訊息的描述*

*具體錯誤訊息在QPlay API Gateway會再包上一層message後回傳給APP, 並透過lang語系參數取得不同語系的message, 詳見下一章節- --Custom API接口訊息定義*

{\"ResultCode\":\"001902\"}

* http response body-content:

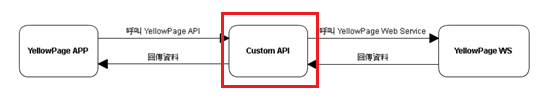
*若是查詢類的API, 除了ResultCode外, content節點內要放查詢結果, 且只能用content這個節點名稱, 將所有需要APP呈現的內容放置於content內, 由QPlay Custom API回傳給APP*

以上的例子是新增類的範例, 如果是查詢類的範例, 回傳值可能如下, 其中Content這個container底下就會包含查詢結果, 如下範例為yellowpage中, QueryEmployeeData的web service回傳結果

{\"ResultCode\":\"001906\",\"Content\":[{\"Company\":\"Qisda\",\"Name\_CH\":\"周芳立\",\"Name\_EN\":\"Steven Chou\",\"Ext\_No\":\"8800-2329\"}]}

### Custom API

*接著說明QPlay Custom API的參數, APP呼叫時需參考此份文件, 才能透過QPlay的API gateway, 取得yellowpage webservice所提供的資訊, 這個章節的內容, 請提供給APP端的RD開發使用*

****

1. Custom API網址

*Custom API的網址為固定格式, 依據3.1.3的function name變化, 如下說明*

https://qplay.benq.com/qplayApi/public/v101/custom/appyellowpage/AddMyPhoneBook?lang=en-us&uuid=xxxxx

*其中appyellowpage處填入透過IT Portal所申請的app-key*

*AddMyPhoneBook填入3.1.3章節的web service function name*

*後面則填上定義的參數, 其中lang及uuid為必填, QPlay要求輸入之必填項目*

1. 傳遞參數方法：

HTTPS POST

*Custom API只接受https post方式*

1. 網址列傳遞參數：

*任何Custom API, 網址列參數lang及uuid為必填*

|  |  |  |  |
| --- | --- | --- | --- |
| **參數名稱** | **是否必填** | **資料類型** | **描述** |
| lang | Y | string | 語系, ex:en-us |
| uuid | Y | string | 手機的uuid識別碼 |

1. Http request headers: *(以下5個參數必填, 欄位名稱大小寫須跟以下範例一模一樣)*

|  |  |  |
| --- | --- | --- |
| 欄位名稱 | 是否必填 | 描述 |
| content-type | 必填 | 訊息體類型，類型需為application/json |
| app-key | 必填 | 專案名稱 *請輸入透過IT Portal所申請的app-key* |
| signature-time | 必填 | 產生Signature當下的時間(unix timestamp型式), 共10碼 |
| signature | 必填 | *請參考4.1章節說明, 將獲得的signature填到http headers內的signature欄位* |
| token | 必填 | 由server配發的token, 用來識別是否已經登入qplay  *token由QPlay app的Qlogin plugin取得* |

1. Http request body:

*內容為html格式, 依照3.1.3章節, 跟web service 端RD所定義好的參數填入*

<LayoutHeader><User\_EmpID>0407731</User\_EmpID><Add\_EmpID>1207952</Add\_EmpID><Add\_Company>BenQ</Add\_Company></LayoutHeader>

因為內文實質上是XML格式, 因此XML保留字元請先做HTMLENCODE

以下5個符號在http post之前需要先轉換為右邊格式

|  |  |
| --- | --- |
| 原始字元 | 轉換字元 |
| < | &lt; |
| > | &gt; |
| & | &amp; |
| ‘ | &apos; |
| “ | &quot; |

Ex: FlowER & IP Servers這樣的內文, 需轉換為FlowER &amp; IP Servers後再送出

1. Response body:

*QPlay收到來自App的request後, 直接將http request body也透過post方式轉送到外部系統的web service*

*收到回應資料後, 按以下格式提供回應訊息給app*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **節點標識** | **父節點標識** | **出現次數** | **資料類型** | **描述** |
| ResultCode | 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 |

1. ResultCode說明:

*custom API收到ResultCode後, 會依據多語系參數, 傳送正確的翻譯內容給app端, 規格書中, 請定義可能發生的result code*

*其中前三碼為project code, 請填入申請app-key時所產生的三碼project code*

*第四碼如果為9, 表示可以直接顯示在APP GUI介面上, 供user識別*

*若非9, 則表示為APP需自行處理的邏輯*

*例如:001013表示參數格式錯誤, 此error需APP端自行處理, 顯示給user也無事無補*

|  |  |  |  |
| --- | --- | --- | --- |
| **Result Code** | **Description - TW** | **Description - EN** | **Description - CN** |
| 001902 | 新增成功 | Added Successfully | 新增成功 |
| 001903 | 新增失敗 | Add Failed | 新增失败 |

1. YellowPage Custom API傳輸規格範例：

URL: https://qplay.benq.com/qplayApi/public/v101/custom/appyellowpage/AddMyPhoneBook?lang=en-us&uuid=d00a793496e110db

Method: HTTP POST

Http request headers:

content-type: application/json

app-key:appyellowpage

signature-time: 1479101172

signature: 48MUfpc192JbYTxqs28Iu8jEXRxMEdUGnPX7lXx77O4=

token: 5808248171a93

Http request body:

<LayoutHeader><User\_EmpID>0407731</User\_EmpID><Add\_EmpID>0604234</Add\_EmpID><Add\_Company>Qisda</Add\_Company></LayoutHeader>

Http response headers:

**Cache-Control →**no-cache

**Connection →**Keep-Alive

**Content-Length →**74

**Content-Type →**application/json

**Date →**Mon, 14 Nov 2016 05:26:26 GMT

**Keep-Alive →**timeout=5, max=100

**Server →**Apache/2.4.6 (CentOS) PHP/7.0.6 OpenSSL/1.0.1e-fips

**X-Powered-By →**PHP/7.0.6

Http response body:

其中ResultCode及Content兩個節點取自上一章節web service

Message, token\_valid則是由QPlay custom API補上  
{\"ResultCode\":\"001902\",\"Message\":\"Added Successfully\",\"token\_valid\":\"1458884996\",\"Content\":\"\"}

1. 範例說明:

9.1) 因為範例是新增類的API, 所以Content處沒有回傳任何消息, 但也會回傳空值

9.2) 如果是查詢類的API, 會將查詢結果放在Content節點

如下是Yellowpage的QueryEmployeeData顯示結果, 會在Content節點看到查詢結果:

{"ResultCode":"1","token\_valid":1487101441,"Message":"成功","Content":[{"Company":"Qisda","Name\_CH":"\u5433\u5947\u5bef","Name\_EN":"Steven CC Wu","Ext\_No":"8800-2121"},{"Company":"Qisda","Name\_CH":"\u5468\u82b3\u7acb","Name\_EN":"Steven Chou","Ext\_No":"8800-2329"}]}

9.3) QPlay Custom API的機制只是將上一章節外部系統web service的回傳值原封不動的回傳, 不會加上任何邏輯, 所有企業邏輯仍是由外部系統控制

9.4) 唯一的差異是QPlay Custom API會補上Message及token\_valid兩個節點, 用來做error code多語系控制及token有限期限控制

# Signature

以下說明Signature獲取方式

QPlay APP的加密演算法, 會附在QSecurity plugin, 符合cordova pluging的做法, 採取javascript的方式

使用方法如下

var signatureParameter = {

    key: appsecretkey,

    time:signature-time

};

window.plugins.qsecurity.getSignature(signatureParameter, callback);

成功會從callback, 取得

callback = function(data) {

    var signature = data[‘result\_code'];

};

# Error Code List

以下說明Error Code的定義

QPlay類的錯誤碼, 編號為999開頭, QPlay回應999開頭的六位數Error Code表示為程式面錯誤, 不需要錯誤消息呈現在APP上, 開發過程中看到999類的訊息, 應直接fix bug

*請先列出以下所有999類的訊息給RD, 以下訊息都需要fix bug*

|  |  |
| --- | --- |
| **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 | APP未上傳 |
| 999016 | 調用custom API參數錯誤 |
| 999999 | 其他未知錯誤 |

APP類的Error Code, 請自行定義, 並提供json file

原則如下:

其中前三碼為project code, 請填入申請app-key時所產生的三碼project code

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**, 其餘未定義, APP端可自行定義, APP收到後可以自行處理

*寫規格時請列出以下非999類別的錯誤訊息供RD判斷該如何處理業務邏輯上的錯誤*

Ex:

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

*Error code的部分需要提供給QPlay處理多語系, 因此, 請PM撰寫以下json格式, 並提供給QPlay負責人:*[qplay@benq.com](mailto:qplay@benq.com)

Ex:

yellowpage的部分json file如下, 請另存為appkey.json, 並提供檔案給QPlay負責人:[qplay@benq.com](mailto:qplay@benq.com)

其中appkey為透過ITportal所申請的appkey

{

"error\_list": {

"appkey": "appyellowpage",

"code\_list": [

{

"error\_code": "001901",

"language\_list": [

{

"language": "en-us",

"error\_description": "No Results Found"

},

{

"language": "zh-tw",

"error\_description": "查無資料"

},

{

"language": "zh-cn",

"error\_description": "查无资料"

}

]

},

{

"error\_code": "001902",

"language\_list": [

{

"language": "en-us",

"error\_description": "Added Successfully"

},

{

"language": "zh-tw",

"error\_description": "新增成功"

},

{

"language": "zh-cn",

"error\_description": "新增成功"

}

]

},

{

"error\_code": "1",

"language\_list": [

{

"language": "en-us",

"error\_description": "OK"

},

{

"language": "zh-tw",

"error\_description": "成功"

},

{

"language": "zh-cn",

"error\_description": "成功"

}

]

}

]

}

}

# Others