|  |  |  |  |
| --- | --- | --- | --- |
| **Subject :** | QPlay Backend Platform | **Latest Ver :** | 1.6 |
| **Latest Release Date :** | 2018-08-24 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Change History** | | | | |
| Date | Author | Version | Type | Comment |
| 2018-05-23 | Darren.K.Ti | 1.0 | Function  DB | 功能修改記錄 : 4.1  1. QPlay 後台新增功能 [ 公司管理 ]  2. QPlay Login 流程修改  3. DB `qplay` 新增 Table  4. QPlay Backend Login - Remember Me |
| 2018-06-19 | Darren.K.Ti | 1.1 | Function  DB | 功能修改記錄 : 4.2  1. QPlay 後台新增功能 [ 消費券管理 ]  2. DB `qplay` 新增 QPay 相關 Table  3. QPlay API 新增   1. QPay 相關 API 2. 更新 QAccount Password   4. APP QPlay   1. 整合 QPay 2. 新增功能 [ 更新 QAccount Password ]   5. QPay Web 查詢頁面 |
| 2018-06-25 | Darren.K.Ti | 1.2 | Function DB  API | 功能修改記錄 : 4.3  1. QPlay 後台新增功能 [ Function Management ]  2. DB `qplay` 新增 Function Manage 相關 Table  3. QPlay API 新增 Function Manage 相關 API |
| 2018-07-06 | Darren.K.Ti | 1.3 | Function  API | 功能修改記錄 : 4.4  1. Plugin QLogin 頁面 支援 QAccount  2. QPlay API login 支援 QAccount  3. QPlay API register 支援 QAccount  3. APP QPlay 使用 Function Manage |
| 2018-07-16 | Darren.K.Ti | 1.3.1 | Function  DB  API | 功能修改記錄 : 4.3  1. Function Detail 的權限管理, 要處理 company\_label  2. 新增設定 QAccount 的權限等級 |
| 2018-07-20 | Darren.K.Ti | 1.4 | Function  DB  API | 功能修改記錄 : 4.2  1. 完成 2018-06-25 尚未完成的項目 |
| 2018-07-26 | Darren.K.Ti | 1.4.1 | Function  API | 功能修改記錄 : 4.2  1. QPay Web API : 修正執行流程  2. 依據功能，個別評估時程 |
| 2018-07-31 | Darren.K.Ti | 1.5 | Function  DB | 功能修改記錄 : 4.5  1. QPlay 後台新增功能 [預約推播]  2. DB `qplay` 新增 預約推播 相關 Table  3. QPlay API - getMessageList 要過濾預約推播 |
| 2018-08-02 | Darren.K.Ti | 1.5.1 | Function | 功能修改記錄 : 4.5  3. QPlay API - getMessageList 要過濾預約推播   * 修正處理邏輯 |
| 2018-08-07 | Darren.K.Ti | 1.5.2 |  | 功能修改記錄 : 4.5 - 修正文字錯誤  4.5.12 程式碼修改記錄  qplay/ app/ Http/ Controllers/ pushController.php   * pushBatchService () - return view “push/ **push\_batch\_service**” * pushBatchServiceData () - return view “push/ **push\_batch\_service\_data**” |
| 2018-08-16 | Darren.K.Ti | 1.4.2 | Function | 功能修改記錄 : 4.3  4.3.12 程式碼修改記錄 -  1. 3.d. 修改文字 'FUNCTION\_VARIABLE\_NAME'  2. 2.h.ii.4.  Function Name 修正為可以編輯修改  Variable Name 修正為不可編輯修改 |
| 2018-08-16 | Darren.K.Ti | 1.4.3 | API | 功能修改記錄 : 4.4  1. QPlay API login 支援 QAccount  2. QPlay API register 支援 QAccount  3. QPlay API logout 支援 QAccount  4.4.12 程式碼修改記錄 qplayApi/  g. app/ Repositories/ UserRepository.php -  funciton QAccountLogin ()  修改程式執行邏輯，QAccount 的  帳號需要比對 `qp\_user`.emp\_no，另外，  不需要驗證 `qp\_user`.ad\_flag = N  Verify.php / CommonUtil.php 新增 function，透過 emp\_no  取得 User 資料 |
| 2018-08-24 | Darren.K.Ti | 1.6 | API | 功能修改記錄 : 4.6  1. QPlay API register 紀錄 ad\_flag 在 qp\_session  2. QPlay API login 紀錄 ad\_flag 在 qp\_session  3. QPlay API sendPushToken 在 JPush  註冊 QAccount Tag |

# 目錄

[**目錄**](#_fvukxvw609q9) **3**

[**1.0 系統設計 (System Design)**](#_17qb5z9rsywe) **5**

[**2.0 資料庫設計 (Database Design)**](#_svpws9pqxbdm) **5**

[**3.0 Implementation**](#_l3yah7s6jx9k) **6**

[**4.0 功能修改記錄**](#_y9sxdk33wuqt) **7**

[4.5 [ QPlay 後台新增功能 [ 預約推播 ]，DB `qplay` 新增 [ 預約推播 ] 相關 Table ，QPlay API - getMessageList 要過濾預約推播]](#_f9crrxf6b8wb) 7

# 1.0 系統設計 (System Design)

**1.1 系統架構圖 (System Framework Diagram)**

* QPlay Backend Platform 整體的系統架構圖，待補上

**1.2 系統流程圖 (System Flow)**

* QChat API 基於 laravel 5.2 框架進行開發，並使用 Repository + Service 模式之程式架構。
* 使用 index.php 作為進入點，route 為路由分配，midware 作為中介層，進行 request 和 response 的驗證，以及寫入 API Log。



**1.3 類別圖 (Package / Class Diagram)**

* QPlay Backend Platform 整體的類別圖，待補上

# 2.0 資料庫設計 (Database Design)

請參考資料庫設計文件 : QPlay\_DB\_Schema.docx

2.1 資料庫物件清單 (Database Object List)

* 每個修改項目 4.X 各自描述

2.2 實體關聯圖 (Entity-Relation Chart)

* 每個修改項目 4.X 各自描述

2.3 資料表結構 (Table Layout)

* 每個修改項目 4.X 各自描述

# 3.0 Implementation

3.1 公用元件 (Public Component)

* 每個修改項目 4.X 各自描述

3.2 共用介面 (Public Interface)

* 每個修改項目 4.X 各自描述

# 4.0 功能修改記錄

## 4.6 [ QPlay API register / login 紀錄 ad\_flag 於 qp\_session，QPlay API sendPushToken 在 JPush 註冊 QAccount Tag]

4.6.1 需求說明

* **QPlay API register / login**
  + 修改處理流程，依據 APP 端傳入的 login\_type (AD or QAccount)，決定當前 User 的 ad\_flag (Y or N)
  + 將 ad\_flag 紀錄於 Table `qp\_session`
* **QPlay API sendPushToken**
  + 修改處理流程，依據 UUID，判定當前 User 的 ad\_flag，如果是 ad\_flag=N，表示需要在 JPush 註冊 QAccount Tag : qsida\_other / benq\_other

4.6.2 時程評估

* QPlay API register / login
  + 預估時程 : ? ~ ? (1 day)
  + 實際時程 :
* QPlay API sendPushToken
  + 預估時程 : ? ~ ? (1 day)
  + 實際時程 :

4.6.3 對應文件

* QPlay\_DB\_Schema\_3.12\_20180824.docx
* Qplay\_API\_1.3.8\_20180824.docx

4.6.4 修改項目

* DB `**qplay**`
  + 修改 Table
    - `**qp\_session**`
* qplayApi/ qplayApi/
  + 修改
    - app/ Http/ Controllers/ **qplayController.php**
    - app/ lib/ **CommonUtil.php**
    - app/ lib/ **PushUtil.php**
    - resources/ views/ **login.blade.php**

4.6.5 功能架構圖

* 無

4.6.6 資料庫物件清單

|  |  |  |  |
| --- | --- | --- | --- |
| **DB Type** | **Type** | **Object Name** | **Description** |
| MySQL | Table | qp\_session | 依據 User 的 UUID，紀錄 ad\_flag。 |

4.6.7 實體關聯圖

* 無

4.6.8 資料表結構

* 參考文件 : QPlay\_DB\_Schema\_3.12\_20180824.docx

4.6.9 公用元件

* 無

4.6.10 共用介面

* Qplay\_API\_1.3.8\_20180824.docx

4.6.11 資料流程圖 Data Flow Diagram

* 無

4.6.12 程式碼修改記錄

* qplayApi/ qplayApi/
  + app/ Http/ Controllers/ **qplayController.php**
    - 修改 function register ()
      * 新增 / 更新 `qp\_session`的時候，需要額外處理資料表欄位 ad\_flag
        + 依據 APP 端帶入的參數

login\_type = AD => ad\_flag = Y

login\_type = QAccount => ad\_flag = N

* + - * 回傳給 APP 端的資料
        + 移除 ad\_flag
    - 修改 function login ()
      * 新增 / 更新 `qp\_session`的時候，需要額外處理資料表欄位 ad\_flag
        + 依據 APP 端帶入的參數

login\_type = AD => ad\_flag = Y

login\_type = QAccount => ad\_flag = N

* + - * 回傳給 APP 端的資料
        + 移除 ad\_flag
    - 修改 function sendPushToken ()
      * 使用 PushUtil::GetTagByUserInfo ()，要加入新的參數，

$adFlag = CommonUtil::ADFlag()

* + - * 依據 PushUtil::GetTagByUserInfo () 的結果
        + 如果 $adFlag = Y，行為不變
        + 如果 $adFlag = N || NULL

如果是 Qisda 員工，要使用 Tag [ qisda\_other ] 在 JPsuh進行註冊

如果是 BenQ員工，要使用 Tag [ benq\_other ] 在 JPsuh進行註冊

* + app/ lib/ **CommonUtil.php**
    - 修改 function getUserInfoByUUID ()
      * 回傳的資料，移除 qp\_user.ad\_flag
  + app/ lib/ **PushUtil.php**
    - 修改 function GetTagByUserInfo ()
      * 新增參數
        + $adFlag
      * 依據 CommonUtil::ADFlag() 決定 company flag 是否要在後方加上 [ \_other ]
        + 如果 ad\_flag = N or NULL

如果是 Qisda 員工，回傳 Tag [ qisda\_other ]

如果是 BenQ員工，回傳 Tag [ benq\_other ]

* + resources/ views/ **login.blade.php**
    - 修改 function login ()
      * Header 參數需要帶入 login\_type
    - 修改 function register ()
      * Header 參數需要帶入 login\_type

4.6.13 Test Case

|  |  |  |  |
| --- | --- | --- | --- |
|  | 測試項目 | 結果 | 驗證 |
| 1. | 使用 QAccount 登入 QPlay，成功登入之後，查看 DB Table `qp\_session`.ad\_flag 是否為 N。 | `qp\_session`.ad\_flag = N。 | Y |
| 2. | 登入極光後台，選擇 comqplayappqplaydev，進入 [推送] > [發送通知]。  先選擇剛剛登入的裝置 (iOS / Android)，接著在 [目標人群] 的項目，點擊 [設備標籤(Tag)]，輸入公司，要出現特定選項  ex:   1. 如果是 BenQ 員工，應該要出現 [BENQ\_other] 的標籤。 2. 如果是 Qisda 員工，應該要出現 [QISDA\_other] 的標籤。 | 有出現 BENQ\_other 標籤。 | Y |
| 3. | 承上，輸入訊息內容，測試發送推播，查看手機是否收到推播。 | 有成功收到推播。 | Y |
| 備註 :   1. 如果是 Staging，要在極光後台選擇 comqplayappqplaytest 2. 標籤 [BENQ\_other] [QISDA\_other] 後方的數字 (1~6)，依據 User AD Account 有所不同 | | |  |