|  |  |  |  |
| --- | --- | --- | --- |
| **Subject :** | QPlay Backend Platform | **Latest Ver :** | 1.5.1 |
| **Latest Release Date :** | 2018-08-02 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
| 2018-010-08 | Candy.XC.Lee | 1.5.2 | Function  DB  API | 功能修改記錄 : 4.5  1. DB `qplay` 修改 預約推播 相關 Table  2. DataFlow  3. 預約推播免費因上限，改為推播 |

# 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

## 4.5 [ 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]

4.5.1 需求說明

* **QPlay 後台新增功能 [ 預約推播]**，在現有的 [推播服務] 底下 修改/新增 子選單 :
  + 推播服務
    - 訊息服務
      * 現有的即時推播功能
      * 需要過濾預約推播的訊息，不做顯示
    - 預約推播
      * 新增的預約推播功能
        + 上傳 Excel
        + Excel 檔案列表
    - 預約推播訊息紀錄
      * 預約推播的訊息記錄
        + 資料列表
        + 統計預約數量 / 已讀數 / 未讀數
* **DB `qplay` 新增 [ 預約推播] 相關 Table**
  + 紀錄
    - 上傳 Excel 的人員 / 檔案名稱 / 時間
    - 每則推播的內容 / 員工編號 / ~~公司~~ / 預約時間
    - JPush 回傳的 schedule\_id
* **QPlay API - getMessageList 要過濾預約推播**
  + 如果推播訊息屬於 [預約推播]，檢查當前時間是否大於預約時間，決定是否讓 User 看到預約推播的內容

4.5.2 時程評估

* QPlay 後台 > 推播服務 > 訊息服務
* QPlay 後台 > 推播服務 > 預約推播
  + - 實際時程 : (14 day)
* QPlay 後台 > 推播服務 > 預約推播訊息紀錄
* QPlayApi API - getMessageList 過濾預約推播

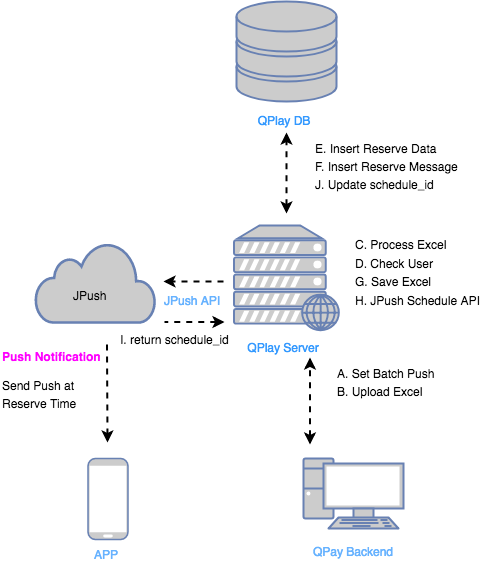
4.5.3 對應文件

* QPlay\_DB\_Schema\_3.9\_20180802.docx
* https://docs.jiguang.cn/jpush/server/push/rest\_api\_push\_schedule/
* https://github.com/jpush/jpush-api-php-client/blob/master/doc/api.md#schedule-api

4.5.4 修改項目

* DB `**qplay**`
  + 修改 Table
    - `**qp\_message**`
    - `**qp\_message\_send**`
  + 新增 Table
    - `**qp\_message\_batch**`
* qplayApi/ qplayApi/
  + 修改
    - app/ Http/ Controllers/ **qplayController.php**
* qplay/
  + 修改
    - app/ Http/ **routes.php**
    - app/ Http/ Controllers/ **pushController.php**
  + 新增
    - app/ Model/ **QP\_Message.php**
    - app/ Model/ **QP\_Message\_Send.php**
    - app/ Model/ **QP\_Message\_Batch.php**
    - app/ Services/ **pushService.php**
    - app/ Repositories/ **pushRepository.php**
    - resources/ views/ push/ **push\_batch\_service.blade.php**
    - resources/ views/ push/ **push\_batch\_service\_data.blade.php**
    - qplay/ resources/ lang/ en/ **messages.php**
    - qplay/ resources/ lang/ zh-cn/ **messages.php**
    - qplay/ resources/ lang/ zh-tw/ **messages.php**

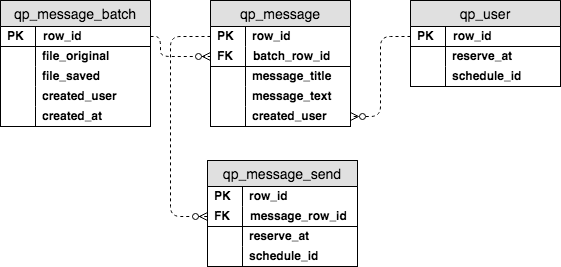
4.5.5 功能架構圖



4.5.6 資料庫物件清單

|  |  |  |  |
| --- | --- | --- | --- |
| **DB Type** | **Type** | **Object Name** | **Description** |
| MySQL | Table | qp\_message\_batch | 紀錄上傳 Excel 的人員 / 檔案名稱 / 時間。 |
| MySQL | Table | qp\_message | 新增欄位 “batch\_row\_id” : 紀錄對應哪一個 `qp\_message\_batch`。 |
| MySQL | Table | qp\_message\_send | 新增欄位 “reserve\_at” : 預約推播的時間。  新增欄位 “schedule\_id” : JPush 回傳的資料。 |

4.5.7 實體關聯圖



4.5.8 資料表結構

* 參考文件 : QPlay\_DB\_Schema\_3.9\_20180802.docx

4.5.9 公用元件

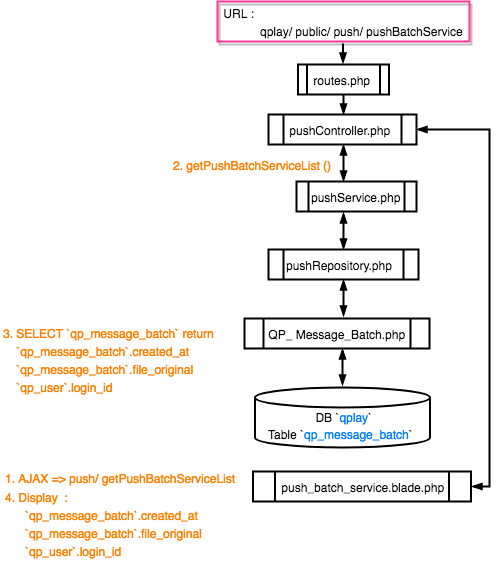
* 無

4.5.10 共用介面

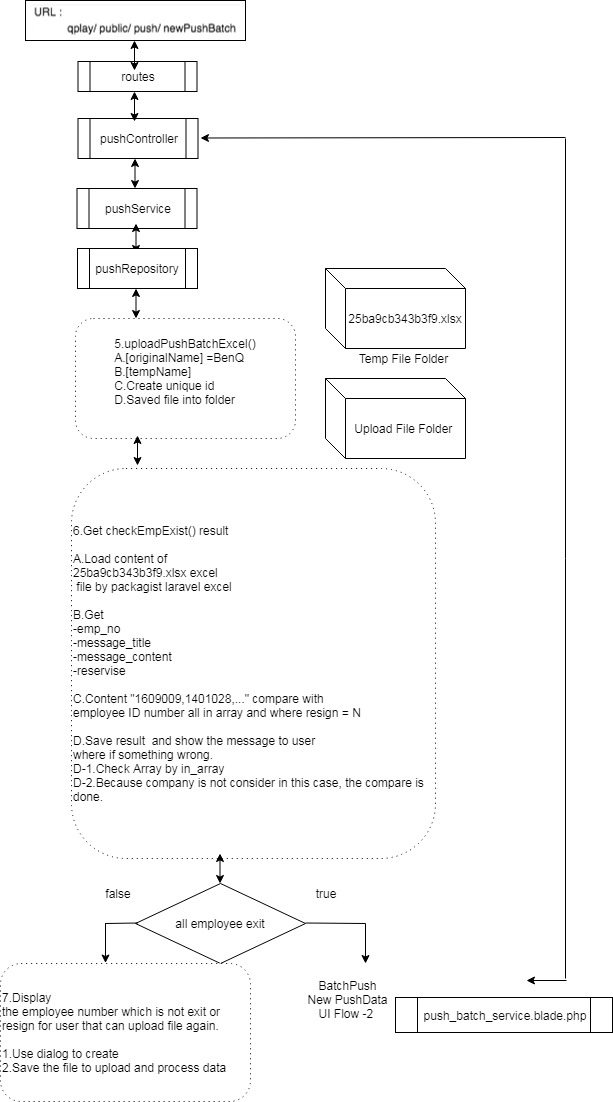
* 無

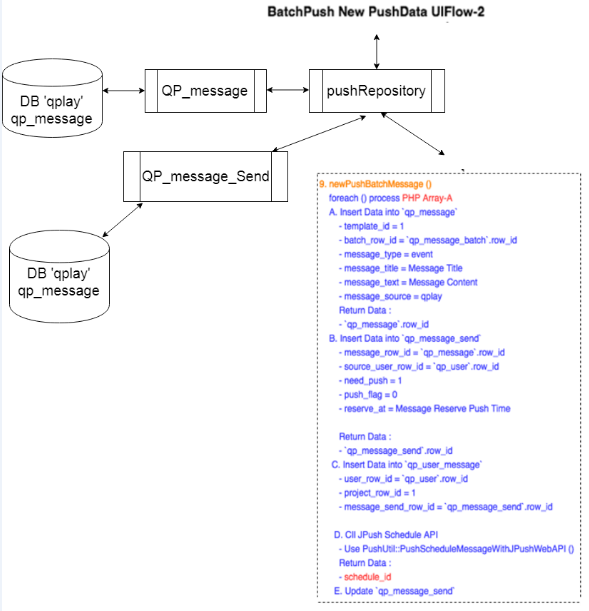
4.5.11 資料流程圖 Data Flow Diagram

* 預約推播 Excel 檔案資料列表

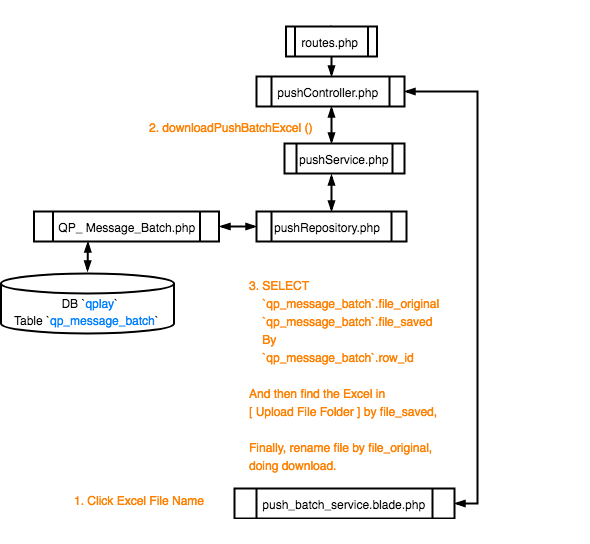


* 新增預約推播 Excel 檔案 - DataFlow 1

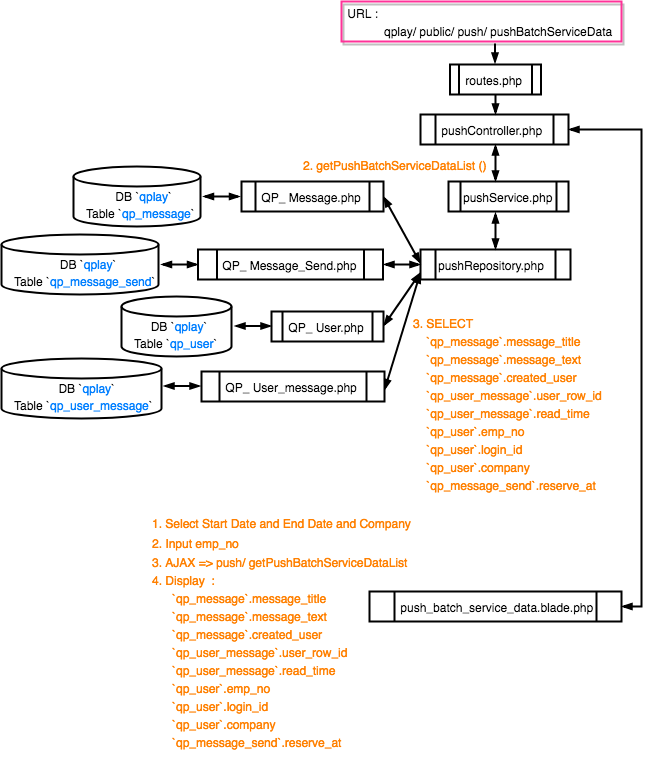




* 新增預約推播 Excel 檔案 - DataFlow2
* 下載 Excel 檔案



* 預約推播訊息紀錄 - 資料列表



4.5.12 程式碼修改記錄

* qplay/
  + app/ Http/ **routes.php**
    - URL Request Handle :
      * 新增 push/
        + pushService，推播服務。
        + pushBatchService，預約推播的 UI。
        + getPushBatchServiceList，預約推播的操作紀錄。
  + app/ Http/ Controllers/ **pushController.php**
    - 定義 function
      * pushBatchService ()
        + return view “push/ pushBatch”。
      * getPushBatchServiceList ()。
  + app/ Model/
    - **QP\_Message.php**
      * 定義 DB `qplay` Table `qp\_message`。
    - **QP\_Message\_Send.php**
      * 定義 DB `qplay` Table `qp\_message\_send`。
    - **QP\_Message\_Batch.php**
      * 定義 DB `qplay` Table `qp\_message\_batch`。
  + app/ Http/ Controllers/ **pushController.php**
    - 定義 function
      * downloadPushBatchExcel ()
        + 使用定義的 downloadPushBatchExcel ()。
      * newPushBatch ()
        + 使用pushRepository.php 定義的 uploadPushBatchExcel ()。
        + 使用pushRepository.php 定義的 newPushBatchData ()。
        + 使用pushRepository.php 定義的 newPushBatchMessage ()。
      * getPushBatchServiceList ()
        + 使用pushRepository.php 定義的 getPushBatchServiceList ()。
      * getPushBatchServiceDataList ()
        + 使用pushRepository.php 定義的 getPushBatchServiceDataList ()。
  + app/ Repositories/ **pushRepository.php**
  + app/ Http/ Controllers/ **pushController.php**
    - * 定義 function uploadPushBatchExcel ()
        + 新增上傳 預約推播 Excel

將 Excel 檔案名稱儲存於[原始檔名]

另外透過 PHP uniqid () 產生 UNIQUE 的[暫存檔名]

存放至暫存資料夾

* + - * + 帶入

`qp\_user`.row\_id

* + - * + 回傳

處理成功後，執行 checkEmpExist ()

* + - * 定義 function downloadPushBatchExcel ()
        + 下載 預約推播 Excel

檔案位於上傳資料夾

下載的時候，使用 `qp\_message\_batch`.file\_original 取代

`qp\_message\_batch`.file\_saved

* + - * + 帶入

`qp\_message\_batch`.row\_id

* + - * 定義 function checkEmpExist ()
        + 檢查 Excel 內的所有 emp\_no 是否存在 DB Table `qp\_user`
        + 執行流程 :

透過 Laravel-Excel 讀取暫存資料夾內的 Excel :[暫存檔名]

將 Excel 資料轉換為陣列 A，儲存於變數

取得所有工號資料，以逗號區隔，組成字串

ex: 1609009, 11004100, 11111108

使用 SQL : SELECT emp\_no,

FROM `qp\_user` WHERE

emp\_no AND resign = N 取得回傳值，將資料轉換為陣列 B，key : emp\_no

比對陣列 A和陣列 B 的 emp\_no

透過 foreach () 處理 陣列 A

先使用 PHP in\_array (needle, array, true) 檢查是否存在

兩者一致，表示所有 Excel 內的員工名單，皆存在 `qp\_user`，回傳 OK

兩者有差異，表示 Excel 內的員工名單，有部分員工不存在於 `qp\_user`，回傳 NO，並且回傳有問題的員工 emp\_no

* + - * + 回傳

全部 emp\_no 皆存在

執行 newPushBatch ()

檢查到 emp\_no 不存在，將有問題的 emp\_no 回傳至前端，透過提示視窗顯示

* + - * 定義 newPushBatchData ()
        + 將剛剛上傳且通過檢查的預約推播 Excel 資料寫入 DB
        + 帶入 原始與暫存檔名
        + 回傳

Y : 新增成功，以及 `qp\_message\_batch`.row\_id

N : 新增失敗

處理流程 :

將剛剛位於暫存資料夾的 Excel 檔案，

使用[暫存檔名]，並且將檔案移轉到上傳資料夾。

刪除暫存資料夾的 Excel 檔案。

將資料新增至 DB Table `qp\_message\_batch`。

回傳新增的 row\_id。

* + - * 定義 newPushBatchMessage ()
        + 依據 `qp\_message\_batch`.row\_id 在 `qp\_message` 及

`qp\_message\_send` 新增資料

* + - * + 帶入

`qp\_message\_batch`.row\_id

* + - * + 處理流程 :

使用 foreach () 處理陣列 A

將資料新增至 `qp\_message`

template\_id = 1

batch\_row\_id = `qp\_message\_batch`.row\_id

message\_type = event

message\_title = 訊息標題

message\_text = 訊息內容

message\_source = qplay

新增成功, 取得 `qp\_message`.row\_id

將資料新增至 `qp\_message\_send`

message\_row\_id = qp\_message`.row\_id

source\_user\_row\_id = `qp\_user`.row\_id

need\_push = 1

push\_flag = 0

reserve\_at = 預約推播的時間

created\_at = 預約推播的時間

新增成功, 取得 `qp\_message\_send`.row\_id

將資料新增至 `qp\_user\_message`

取得此 User 所有的 UUID

project\_row\_id = 1

user\_row\_id = 收到推播的員工

message\_send\_row\_id =

`qp\_message\_send`.row\_id

使用 ~~PushUtil::PushScheduleMessageWithJPushWebAPI ()~~

~~進行 JPush Schedule API 處理~~

$send\_by\_tag = false

~~其他參數依據 function 定義帶入~~

JPush 會回傳 schedule\_id / name，儲存於 PushScheduleMessageWithJPushWebAPI ()

的回傳值 $result["content"]

最後，將 JPush 回傳的 scheudle\_id

更新至 `qp\_message\_send`.schedule\_id

使用PushUtil::PushMessageWithJPushWebAPI進行 JPush Schedule API 處理

參數依據 function 定義帶入

JPush 會回傳 result，將 JPush 回傳的部分更新 [TODO]

{"result":false,"info":"APIRequest Exception occurred"}

更新至 `qp\_message\_send`

* + - * + Y : 處理成功
        + N : 處理失敗
      * 定義 function getPushBatchServiceList ()
        + 取得預約推播的 Excel 上傳資料
        + 回傳

`qp\_message\_batch`.row\_id

`qp\_message\_batch`.file\_original

`qp\_user`.login\_id

`qp\_message\_batch`.created\_at

* + - * 定義 function getPushBatchServiceDataList ()
        + 取得預約推播的訊息內容資料
        + 帶入

start\_date

end\_date

company

emp\_no (選填)

* + - * + 回傳

`qp\_message`.message\_title

`qp\_message`.message\_text

`qp\_user\_message`.user\_row\_id (接收推播的使用者)

`qp\_user`.emp\_no (接收推播使用者的工號)

`qp\_user`.login\_id (接收推播使用者的帳號)

`qp\_message\_send`.reserve\_at (預約推播的時間)

`qp\_message`.created\_user (推播者)

`qp\_user\_message`.read\_time (已讀 / 未讀)

預約推播總數

員工已讀總數

* + - * + 備註 :

取得全部資料之後，透過 PHP 統計剛剛取得的資料，計算

預約推播總數

員工已讀總數

* + resources/ views/ **push**/
    - **push\_batch\_service.blade.php**
      * 點擊 [新增] 開啟 dialog\_content
        + 點擊 [上傳預約推播 Excel]，選擇 Excel 進行上傳
        + 點擊 [儲存]，透過 pushService.php

定義的 newPushBatch () 進行上傳以及處理 Excel 資料。

* + - * 預約推播 Excel 檔案資料列表 :
        + 使用 bootstrap table

id = gridBatchPushList

data-url = push/ getPushBatchServiceList

data-field

`qp\_message\_batch`.row\_id

`qp\_message\_batch`.file\_original

`qp\_user`.login\_id

`qp\_message\_batch`.created\_at

* + - * 點擊 [檔案名稱] 進行下載
        + 觸發 push/ downloadPushBatchExcel
    - **push\_batch\_service\_data.blade.php**
      * 預約推播訊息資料列表 :
        + 條件式 :

開始日期 (必填) : 預設給予操作時月份的第一天

結束日期 (必填) : 預設給予操作時月份的最後一天

公司 (必填) : 預設選在第一個選項

工號 (選填) : 有填入資料時，才加入搜尋條件

* + - * + 使用 bootstrap table

id = gridBatchPushdataList

data-url = push/ getPushBatchServiceDataList

data-field

`qp\_message`.message\_title

`qp\_message`.message\_text

`qp\_user\_message`.user\_row\_id (接收推播的使用者)

`qp\_user`.emp\_no (接收推播使用者的工號)

`qp\_user`.login\_id (接收推播使用者的帳號)

`qp\_user`.login\_id (接收推播使用者的公司)

`qp\_message\_send`.reserve\_at (預約推播的時間)

`qp\_message`.created\_user (推播者)

`qp\_user\_message`.read\_time (已讀)

* + - * + 預設只顯示訊息的標題，使用 Tooltip 的方式去顯示訊息內容

當滑鼠移到標題上方時，動態產生 DIV，顯示

詳細的日期時間 > YYYY/MM/DD HH:ii

標題

內容

* + - * + 統計資訊

預約推播總數

員工已讀總數

* qplayApi/ qplayApi/
  + app/ Http/ Controllers/ **qplayController.php**
    - 修改 function
      * getMessageList ()
        + 必須加上條件式 :

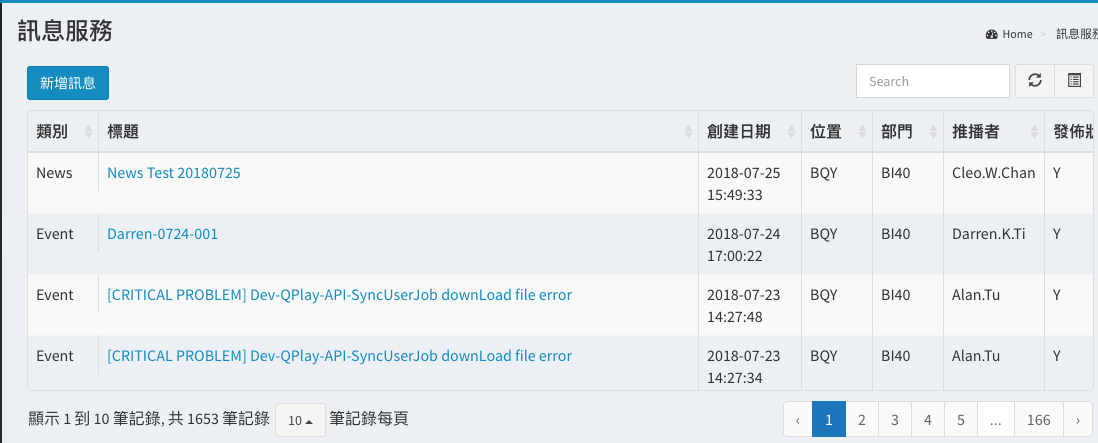
`qp\_message\_send`.reserve\_at > Now

* + - * + 撈取的資料中，如果 `qp\_message\_send`.reserve\_at != null，表示此資料為預約推播，必須檢查預約的時間，是否大於現在，決定是否要回傳此筆資料給 APP 端
        + [ 預約推播 ]在新增資料的時候，created\_at 就設定等於 reserve\_at，

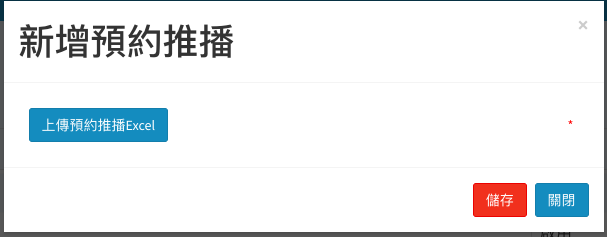
在原先的 SQL 條件式執行之下，就可以避免[ 預約推播 ]的訊息被使用者提前看到。

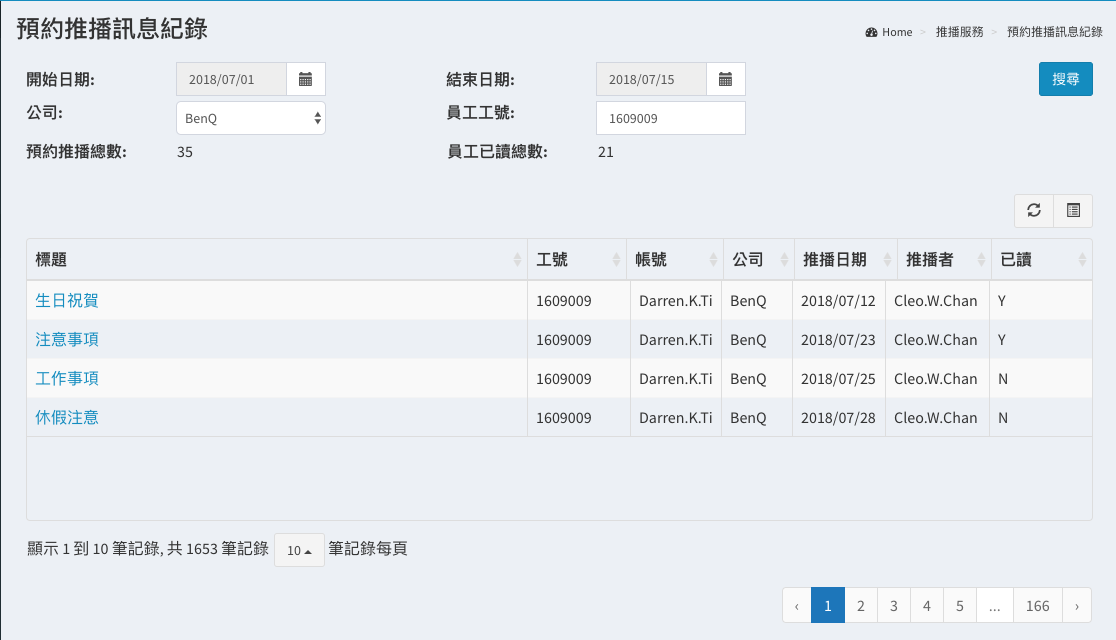
* QPlay Backend Platform
  + [ 系統管理 ] > [ 選單維護 ] > 點擊 ( **推播服務** ) > [ 新增 子選單 ]
    - 菜單名 : PUSH\_SERVICE
      * 鏈結 : pushService
      * 英文名 : Push Service
      * 簡體中文名 : 推播服务
      * 繁體中文名 : 推播服務
    - 菜單名 : PUSH\_BATCH\_SERVICE
      * 鏈結 : pushBatchService
      * 英文名 : Push Batch Service
      * 簡體中文名 : 预约推播
      * 繁體中文名 : 預約推播
    - 菜單名 : PUSH\_BATCH\_SERVICE\_DATA
      * 鏈結 : pushBatchServiceData
      * 英文名 : Push Batch Service Data
      * 簡體中文名 : 预约推播讯息纪录
      * 繁體中文名 : 預約推播訊息紀錄
  + [ 系統管理 ] > [ 群組維護 ] > 新增 / 編輯
    - 群組名稱 : PushManager
    - 系統群組 : 勾選 [ 推播服務 ] 全部功能
  + 上述動作完成之後，必須修改以下檔案，以 (菜單名) 為變數名稱，建立相對應的資料
    - qplay/ resources/ lang/ en/ **messages.php**
      * 'TITLE\_PUSH\_SERVICE' => 'Push Service'
      * 'TITLE\_PUSH\_BATCH\_SERVICE' => 'Batch Push Service'
      * 'TITLE\_PUSH\_BATCH\_SERVICE\_DATA' => 'Batch Push Service Data'
      * 'PUSH\_NEW\_BATCH\_SERVICE' => 'Add Batch Push Service'
      * 'PUSH\_UPLOAD\_BATCH\_SERVICE' => 'Upload Batch Push Service Excel'
      * 'START\_DATE' => 'Start Date'
      * 'END\_DATE' => 'End Date'
      * 'PUSH\_DATE' => 'Push Date'
      * 'PUSH\_READ' => 'Read'
      * 'PUSH\_COUNT\_BATCH\_PUSH' => 'Count of Push Message'
      * 'PUSH\_COUNT\_BATCH\_PUSH\_READ' => 'Count Read of Push Message'
    - qplay/ resources/ lang/ zh-cn/ **messages.php**
      * 'TITLE\_PUSH\_SERVICE' => '推播服务'
      * 'TITLE\_PUSH\_BATCH\_SERVICE' => '预约推播'
      * 'TITLE\_PUSH\_BATCH\_SERVICE\_DATA' => '预约推播讯息纪录'
      * 'PUSH\_NEW\_BATCH\_SERVICE' => '新增预约推播讯息'
      * 'PUSH\_UPLOAD\_BATCH\_SERVICE' => '上傳预约推播讯息 Excel'
      * 'START\_DATE' => '开始日期'
      * 'END\_DATE' => '结束日期'
      * 'PUSH\_DATE' => '推播日期'
      * 'PUSH\_READ' => '已读'
      * 'PUSH\_COUNT\_BATCH\_PUSH' => '预约推播总数'
      * 'PUSH\_COUNT\_BATCH\_PUSH\_READ' => '讯息已读总数'
    - qplay/ resources/ lang/ zh-tw/ **messages.php**
      * 'TITLE\_PUSH\_SERVICE' => '推播服務'
      * 'TITLE\_PUSH\_BATCH\_SERVICE' => '預約推播'
      * 'TITLE\_PUSH\_BATCH\_SERVICE\_DATA' => '預約推播訊息紀錄'
      * 'PUSH\_NEW\_BATCH\_SERVICE' => '新增預約推播訊息'
      * 'PUSH\_UPLOAD\_BATCH\_SERVICE' => '上傳預約推播訊息 Excel'
      * 'START\_DATE' => '開始日期'
      * 'END\_DATE' => '結束日期'
      * 'PUSH\_DATE' => '推播日期'
      * 'PUSH\_READ' => '已讀'
      * 'PUSH\_COUNT\_BATCH\_PUSH' => '預約推播總數'
      * 'PUSH\_COUNT\_BATCH\_PUSH\_READ' => '訊息已讀總數'

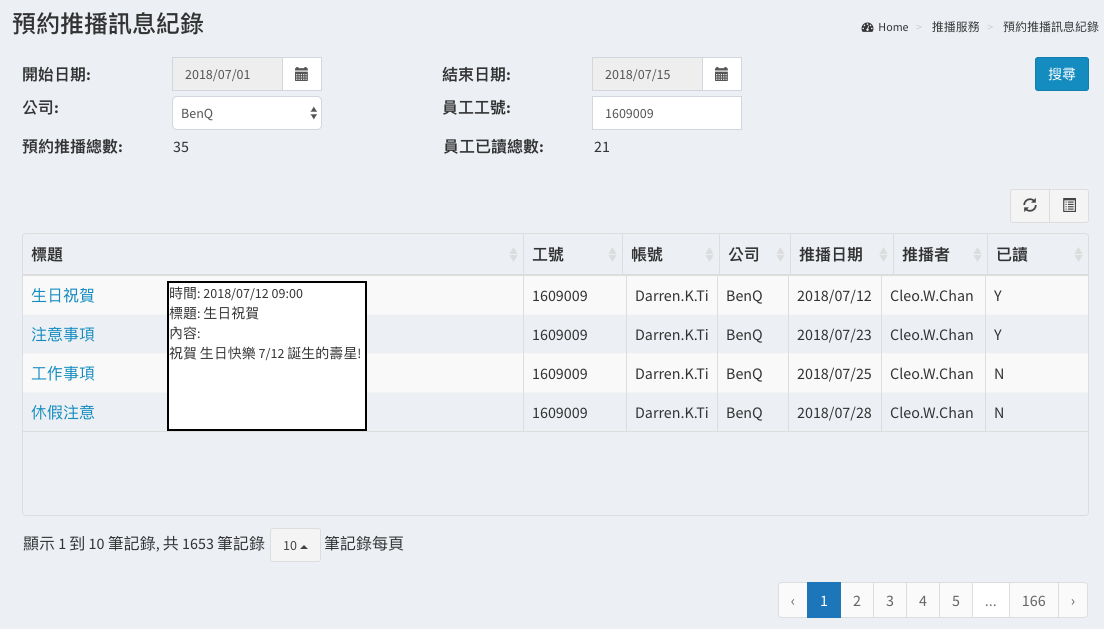
4.5.13 UI 示意圖

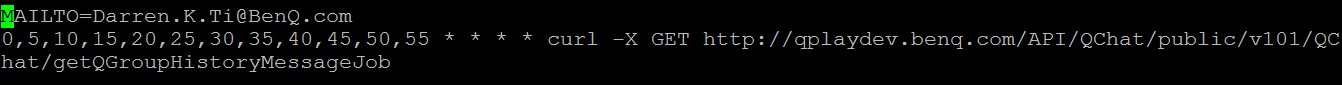












Linux 檢查