|  |  |  |  |
| --- | --- | --- | --- |
| **Subject :** | QPlay Backend Platform | **Latest Ver :** | 1.0 |
| **Latest Release Date :** | 2018-05-23 |

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

# 目錄

[**目錄**](#_30j0zll) **3**

[**1.0 系統設計 (System Design)**](#_1fob9te) **5**

[**2.0 資料庫設計 (Database Design)**](#_2et92p0) **6**

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

[**4.0 功能修改記錄**](#_3dy6vkm) **7**

[4.1 [ QPlay 後台新增功能 [ 公司管理 ]，以及 QPlay Login Page 使用動態公司選項，和 QPlay Login 流程修改，另外修正 QPlay Backend Login - Remember Me ]](#_1t3h5sf) [7](#_1t3h5sf)

[4.2 [ QPlay 後台新增功能 [ 消費券管理 ]，DB `qplay` 新增 QPay 相關 Table，QPlay API 新增 QPay 相關 API 以及 更新 QAccount Password，APP QPlay 整合 QPay 以及新增功能 [ 更新 QAccount Password ]，QPay Web 查詢頁面 ]](#_4d34og8) [19](#_4d34og8)

[4.3 [ QPlay 後台新增功能 [ Function Manage ]，DB `qplay` 新增 Function Manage 相關 Table，QPlay API 新增 Function Manage 相關 API ]](#_2s8eyo1) [26](#_2s8eyo1)

[4.4 [ Plugin QLogin 頁面 支援 QAccount，QPlay API login 支援 QAccount，APP QPlay 使用 Function Manage ]](#_17dp8vu) 42

# 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

## 4.1 [ QPlay 後台新增功能 [ 公司管理 ]，以及 QPlay Login Page 使用動態公司選項，和 QPlay Login 流程修改，另外修正 QPlay Backend Login - Remember Me ]

4.1.1 需求說明

* APP QPlay Login UI 上面的公司選項，目前為 Hard Code 寫死，考量到之後陸續會有新公司加入集團，公司選項需要變更為動態產生，由 QPlay Backend 提供資料。
* QPlay Backend Platform Login UI 上面的公司選項，目前為 Hard Code 寫死，考量到之後陸續會有新公司加入集團，公司選項需要變更為動態產生。
* QPlay Backend Platform 新增功能 [ 公司管理 ]，管理公司名稱，以及驗證 AD 的方式和 Server IP。
* QPlay Backend Login 的 Remember Me 無法正確運作，修正。

4.1.2 時程評估

* [ QPlay 後台新增功能 (公司管理) ]
  + 預估時程 : 5/29 ~ 6/1 (4天)
  + 實際時程 : 5/29 ~ 5/31 (3天)
* [ QPlay Login Page 使用動態公司選項 ]
  + 預估時程 : 6/4 ~ 6/5 (2天)
  + 實際時程 : 6/1 (3hr)
* [ QPlay Login 流程修改 ]
  + 預估時程 : 6/6 ~ 6/8 (3天)
  + 實際時程 :6/1 (3hr)
* [ QPlay Backend Login - Remember Me ]
  + 預估時程 : 6/6 ~ 6/8 (3天)
  + 實際時程 :6/1 (2hr)

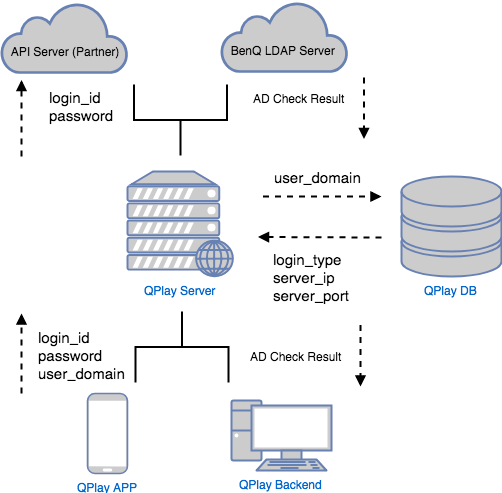
4.1.3 對應文件

* QPlay\_DB\_Schema\_3.0\_20180523.docx

4.1.4 修改項目

1. DB `qplay`
   1. 新增
      1. Table `**qp\_company**`
      2. Table `**qp\_company\_log**`
2. qplayApi/ qplayApi/
   1. 修改
      1. app/ Http/ **routes.php**
      2. app/ Http/ Controllers/ **qplayController.php**
      3. resources/ views/ **login.blade.php**
   2. 新增
      1. app/ Model/
         1. **QP\_User.php**
         2. **QP\_Company.php**
      2. app/ Repositoreies/ **CompanyRepository.php**
      3. app/ Services/ **CompanyService.php**
3. qplay/
   1. 修改
      1. app/ Http/ **routes.php**
      2. app/ Http/ Controllers/ **AuthController.php**
      3. app/ lib/ **ResultCode.php**
      4. resources/ views/ auth/ **login.blade.php**
      5. resources/ lang
         1. en/ **messages.php**
         2. zh-cn/ **messages.php**
         3. zh-tw/ **messages.php**
   2. 新增
      1. app/ Model/
         1. **QP\_Company.php**
         2. **QP\_Company\_Log.php**
      2. app/ Repositoreies/
         1. **CompanyRepository.php**
         2. **CompanyLogRepository.php**
      3. app/ Http/ Controllers/ **companyController.php**
      4. app/ Services/ **CompanyService.php**
      5. resources/ views/ company\_maintain/
         1. **company\_list.blade.php**
4. 系統操作
   1. [ 系統管理 ] > [ 選單維護 ] > [ 新增 ] 建立 ( **公司管理** )
   2. [ 系統管理 ] > [ 選單維護 ] > 點擊 ( **公司管理** ) > [ 新增 子選單 ] 建立 ( **公司管理** )
   3. [ 系統管理 ] > [ 群組維護 ] 設定可以使用 ( **公司管理** ) 的 QPlay Backend User

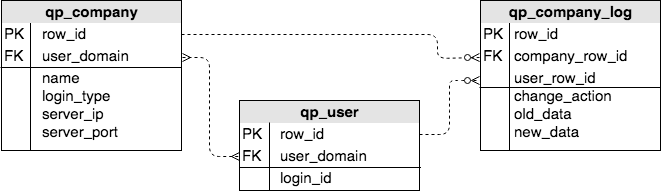
4.1.5 功能架構圖



4.1.6 資料庫物件清單

|  |  |  |  |
| --- | --- | --- | --- |
| **DB Type** | **Type** | **Object Name** | **Description** |
| MySQL | Table | qp\_user | 取得員工的基本資料，判定屬於哪家公司，決定使用哪種登入驗證方式。 |
| MySQL | Table | qp\_company | 維護公司資料，設定 LDAP 或 API 登入，以及 Server IP。 |
| MySQL | Table | qp\_company\_log | 維護公司資料，操作紀錄 Log。 |

4.1.7 實體關聯圖



4.1.8 資料表結構

* 參考文件 : QPlay\_DB\_Schema\_3.0\_20180523.docx

4.1.9 公用元件

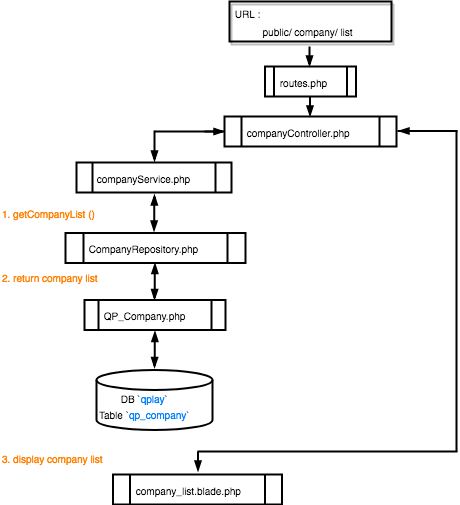
* 內部模組

4.1.10 共用介面

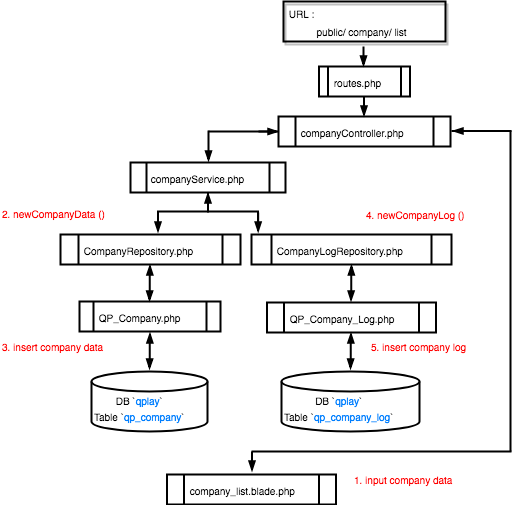
* 無

4.1.11 資料流程圖 Data Flow Diagram

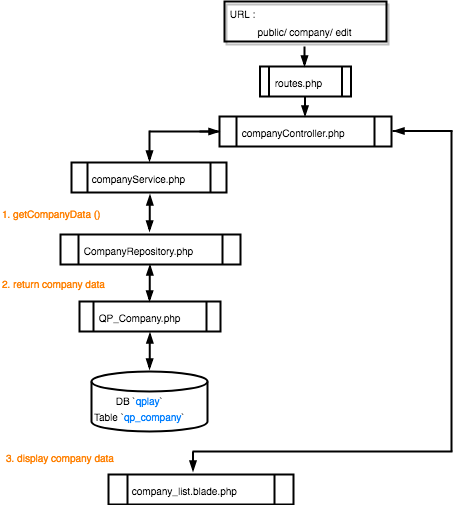
* QPlay 後台的新功能 [ 公司管理 ] - 資料列表



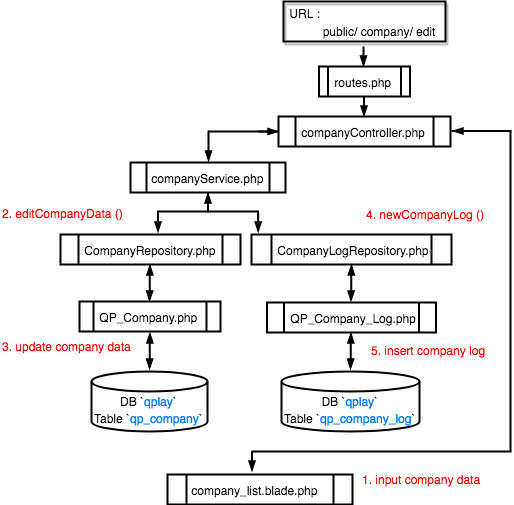
* QPlay 後台的新功能 [ 公司管理 ] - 新增資料



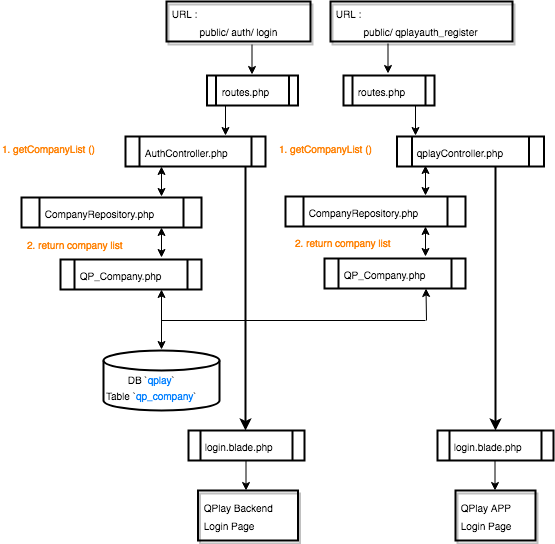
* QPlay 後台的新功能 [ 公司管理 ] - 資料編輯 - 顯示資料



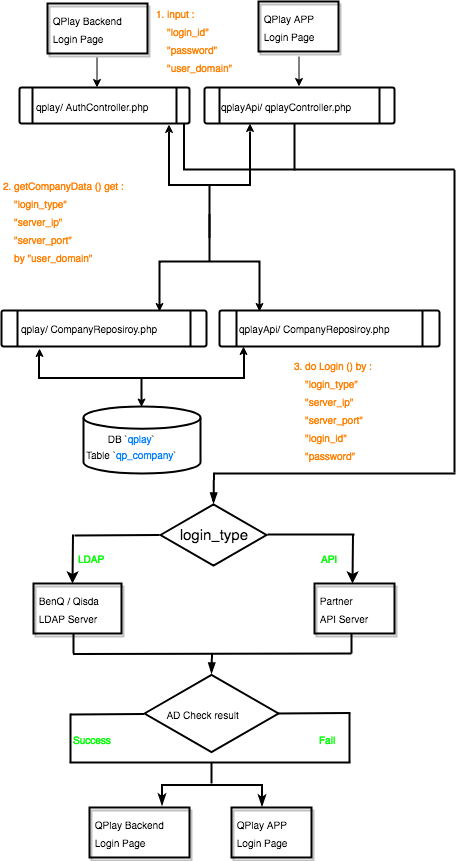
* QPlay 後台的新功能 [ 公司管理 ] - 資料編輯 - 更新資料



* QPlay Login Page 使用動態公司選項



* QPlay Login 流程修改



4.1.12 程式碼修改記錄

1. qplayApi/ qplayApi/
   1. app/ Http/ **routes.php**
      1. 由先 URL Request : /qplayauth\_register 會直接指派 login.blade.php 呈現 View。現在修改流程，先經過 qplayController.php 處理。
   2. app/ Http/ Controllers/ **qplayController.php**
      1. 動態公司選項 :
         1. 透過 ComponyRepository.php 取得 公司資料。
         2. 指派 login.blade.php 呈現 QPlay APP Login View。
      2. Login 流程修改 :
         1. 依據 User 輸入的 login\_id，透過 QP\_User.php 取得 user\_domain。
         2. 依據 user\_domain，透過 CompanyRepository.php 取得 login\_type 以及 server\_ip 和 server\_port。
         3. 使用 LDAP or API 驗證 User AD。
   3. resources/ views/ **login.blade.php**
      1. 下拉選單 id = ddlCompany 的 選項內容，原先 Hard Code 寫為 BENQ 跟 QGROUP，現在要改為由資料庫提供動態資料。
   4. app/ Model/ **QP\_User.php**
      1. 定義 DB `qplay` Table `qp\_user`。
   5. app/ Model/ **QP\_Company.php**
      1. 定義 DB `qplay` Table `qp\_company`。
   6. app/ Services/ **CompanyService.php**
      1. 透過 CompanyRepository.php 取得公司資料。
   7. app/ Repositoreies/ **CompanyRepository.php**
      1. 取得 company list。
      2. 取得欄位資料 “login\_type” “server\_ip” “server\_port”。
2. qplay/
   1. app/ Http/ **routes.php**
      1. 由先 URL Request : auth/ login 會直接指派 views/ auth/ login.blade.php 呈現 View。現在修改流程，先經過 authController.php 處理。
      2. 新增 URL Request Handle : companyMaintain，經由 CompanyController.php 處理 [ 公司管理 ]。
   2. app/ Http/ Controllers/ **AuthController.php**
      1. 動態公司選項 :
         1. 透過 ComponyRepository.php 取得 公司資料。
         2. 指派 views/ auth/ login.blade.php 呈現 QPlay Backend Login View。
      2. Login 流程修改 :
         1. 依據 User 輸入的 login\_id，透過 QP\_User.php 取得 user\_domain。
         2. 依據 user\_domain，透過 CompanyRepository.php 取得 login\_type 以及 server\_ip 和 server\_port。
         3. 使用 LDAP or API 驗證 User AD。
      3. Login Page - Remember Me
         1. 增加 Session 記憶 lang / loginid / domain / remember。
   3. app/ lib/ **ResultCode.php**
      1. 針對 [公司管理] 的資料維護結果，新增 Result code。
   4. app/ Http/ Controllers/ **CompanyController.php**
      1. 依據 URL Request，決定如何使用 CompanyService.php。
   5. app/ Services/ **CompanyService.php**
      1. 公司資料列表 :
         1. 透過 ComponyRepository.php 取得 公司資料 做列表呈現。
      2. 公司資料新增 :
         1. 透過 ComponyRepository.php 新增 公司資料。
      3. 公司資料編輯 :
         1. 透過 ComponyRepository.php 編輯 公司資料。
      4. 公司資料操作 Log :
         1. 只要新增 or 編輯公司資料，透過 ComponyLogRepository.php 寫入 Log。
   6. resources/ views/ auth/ **login.blade.php**
      1. 下拉選單 id = domain 的 選項內容，原先 Hard Code 寫為 BenQ 跟 Qgroup，現在要改為由資料庫提供動態資料。
      2. 使用 Session 來處理 Remember Me 功能。
   7. app/ Model/ **QP\_Company.php**
      1. 定義 DB `qplay` Table `qp\_company`。
   8. app/ Model/ **QP\_Company\_log.php**
      1. 定義 DB `qplay` Table `qp\_company\_log`。
   9. app/ Repositoreies/ **CompanyRepository.php**
      1. 取得 所有公司資料。
      2. 取得欄位資料 “login\_type” “server\_ip” “server\_port”。
      3. 新增 公司資料。
      4. 編輯 公司資料。
   10. app/ Repositoreies/ **CompanyLogRepository.php**
       1. 新增 company 資料異動時的 Log。
   11. resources/ views/ company\_maintain/ **company\_list.blade.php**
       1. 呈現 所有公司的列表資料。
       2. 使用 Javascript 動態產生視窗，新增公司資料，送到 Server 端做處理。
       3. 必須設定 $menu\_name，才能在選單看到 [公司管理]。
   12. resources/ views/ company\_maintain/ **company\_detail.blade.php**
       1. 呈現 單一公司的詳細資料。
       2. 編輯 公司資料，送到 Server 端做處理。
3. QPlay Backend Platform
   1. [ 系統管理 ] > [ 選單維護 ] > [ 新增 ]
      1. 菜單名 : COMPANY\_MAINTAIN
      2. 英文名 : Company Maintain
      3. 簡體中文名 : 公司管理
      4. 繁體中文名 : 公司管理
   2. [ 系統管理 ] > [ 選單維護 ] > 點擊 ( **公司管理** ) > [ 新增 子選單 ]
      1. 菜單名 : COMPANY\_MAINTAIN
      2. 鏈結 : companyMaintain
      3. 英文名 : Company Maintain
      4. 簡體中文名 : 公司管理
      5. 繁體中文名 : 公司管理
   3. [ 系統管理 ] > [ 群組維護 ]
      1. Administrator : 預設有最高權限，可以使用全部群組。
   4. 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
      1. qplay/ resources/ lang/ en/ **messages.php**
      2. qplay/ resources/ lang/ zh-cn/ **messages.php**
      3. qplay/ resources/ lang/ zh-tw/ **messages.php**
   5. 最後，要看 User 的操作權限設定選項中的 [公司管理] 是否被勾起。
      1. 備註 : 就算是 Administrator 群組，個人的操作權限不一定跟群組一樣。