第七周 網路程式基礎 Part 2 & Firbase Part 登入

今天的目標

Alamofire
SwiftyJSON
Firebase Part 1

—

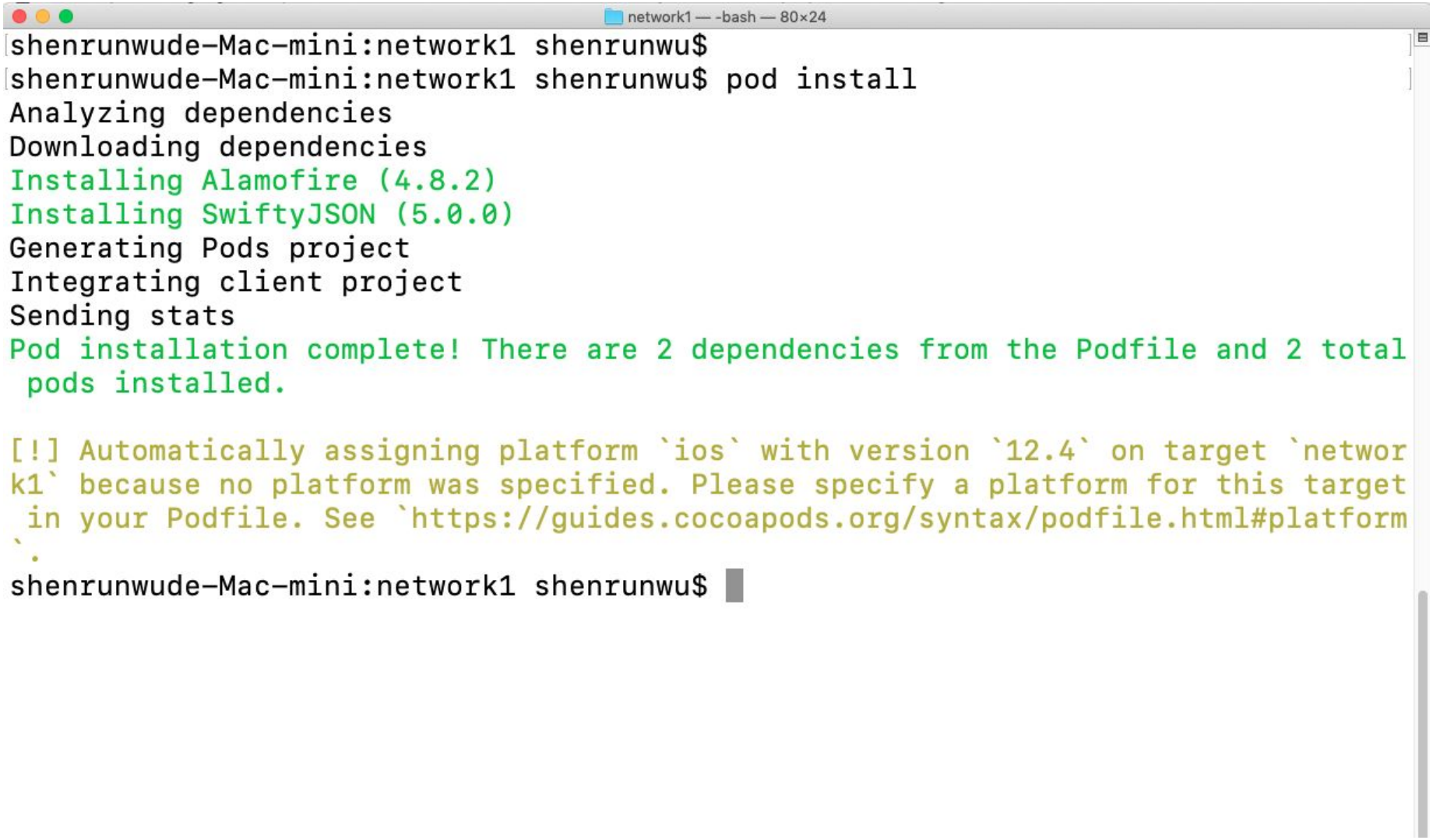
```
target 'network1' do

  pod 'SwiftyJSON'
  pod 'Alamofire'

  use_frameworks!

end
```

Podfile 内容



執行 pod install

```
import UIKit
import Alamofire

class APIModel {
    static var share = APIModel()
    private var apiURL = "https://randomuser.me/api/"
    private init(){}
    func queryRandomUserAlamofire(completion:@escaping (_ Data:Any?,_ respError:
Error?)->())->(){
        let url = apiURL
        let parameters:Parameters? = nil
        DispatchQueue.global().async {
            AF.request(url,
                       method: .get,
                       parameters: nil,
                       encoding: URLEncoding.default,
                       headers: nil).responseJSON(completionHandler: { (response) in
                        DispatchQueue.main.async(execute: {
                            switch response.result{
                            case .success(_):
                                return completion(response.data,nil)
                            case .failure(let error):
                                return completion(nil,error)
                            }
                        })
                       })
        }
    }
}
```

建立 讀取 API Model

Swifty JSON

著名的第三方 JSON 程式庫, 比起 apple 原生的 JSONSerialization 或 JSONDecoder 來說, 有使用方便, 效能良好等優點, 不必寫回傳Delegate, 也不用事先套好格式, 幾乎和 Dictionary 或 Array 一樣, 當成資料型別來用就可以了, 但會佔用較多的記憶體。

```
import UIKit
import SwiftyJSON
class ViewController: UIViewController {
    override func viewDidLoad() {
        super.viewDidLoad()
        APIModel.share.queryRandomUserAlamofire { (resData, error) in
            if error != nil{ print("error") }
            do{
                if let data = resData as? Data{
                    let json = try JSON(data: data)
                    //print(json)
                    print(json["results"][0]["login"]["username"].stringValue)
                    print(json["results"][0]["picture"]["thumbnail"].stringValue)
                }
            }catch{
                print("API Error")
            }
        }
    }
}
```

取得資料

練習

若己可成功取得頁面, 製作一個畫面上面要有

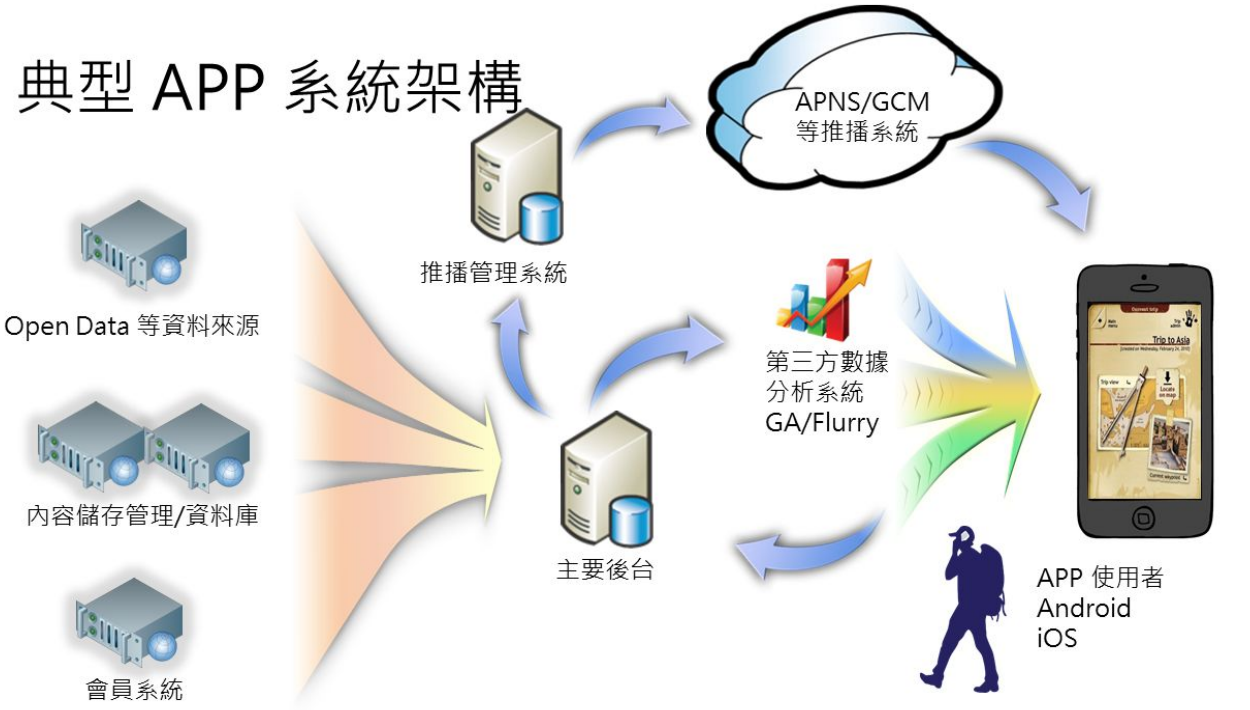
1. 大頭照, 圓形
2. 名字與電話
3. 更新按鈕

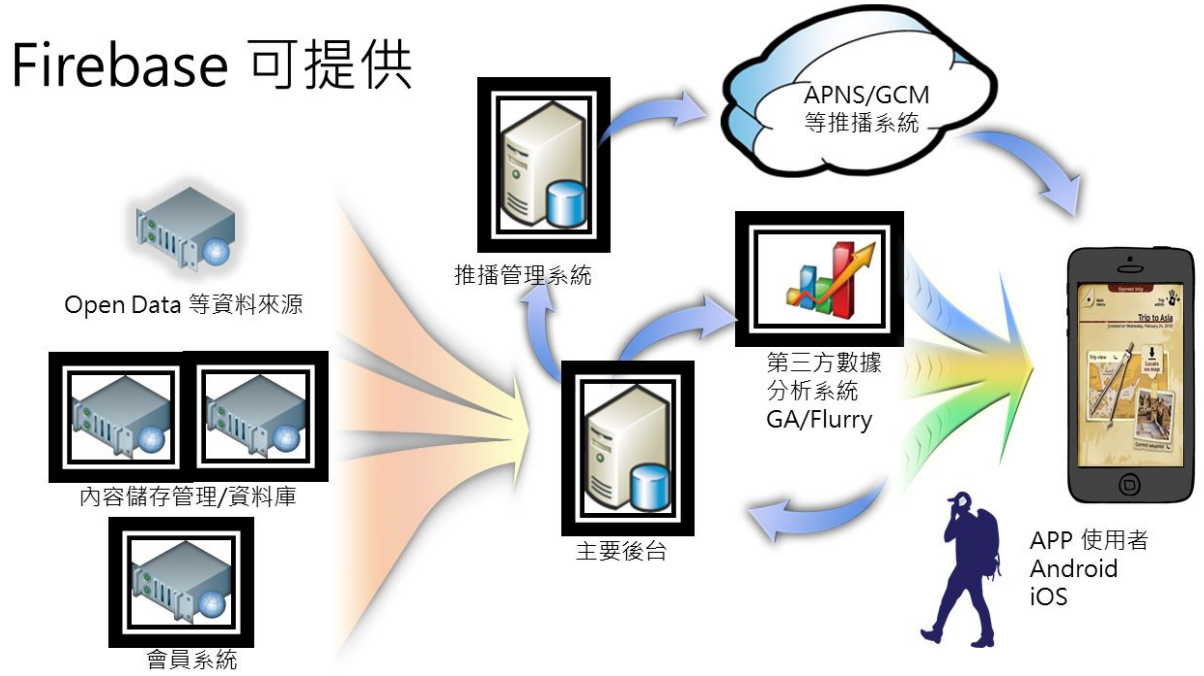
Firebase

Firebase

Firebase 是 Google 子公司, 主導一個為行動應用程式開發的 BaaS (Backend As A Service) 雲端系統, 其主要的目的是要解決在行動應用開發中, 希望取代部份或全部的後台功能, 讓開發者能有一個不需要考量效能需求與功能驗證的後台, 也不需要特別配合後台的開發時程, 或伺服器的採購行政流程等項目, 可以更有效率的開發應用。

為什麼需要 Firebase？





Firebase 有那些功能？

- 會員認證
  - 就是各種帳密認證與第三方整合
  - 我們會實作 GoogleID整合 Facebook 整合 電子郵件認證
- Reltime Database 即時資料庫
  - 由資料驅動的即時 NoSQL
  - 我們會實作一個類討論區的留言版
- Cloud Message 訊息推播
  - 基於 GCM/APNS 的推播系統
  - 我們實作測試 APNS 的推播
- 其他功能
  - 遠端設定，Cloud Hosting，Cloud Storage，Test Lab，機器學習，Chrashlytics，Gooogle Analytics，Performance Monitoring…..

Firebase 需要費用嗎？

基本上，這是一個商業系統，所以一定是要錢的，但很大程度上，為了推廣程學的使用，初學或測試期間，是免費的。在一些特定功能上，它也是完全免費的。當然，到底免費的範圍有多少，是 Google 依當時的政策來決定的。

https://firebase.google.com/pricing

Firebase 準備工作

- 你要有一個 Google ID (G-mail)帳號, 和一個合法的手機號碼, 可以收簡訊的
- 在 Firebase 網站, 啟用你的帳號 (可能需要簡訊認證或信用卡認證)
- 整合 Facebook 你還需要一個 Facebook 帳號
- 因為眾所周知的原因, Firebase 在中國的運作不是很順暢, 必需科學上網才能使用
- Firebase Facebook Google 各家的 SDK 變動的速度都算快, 實作的文件, 建議看線上的, 最好多找幾個參考點, 看看那一份是最新的

建立 Firebase 整合專案

1. 建立 Xcode iOS 專案
2. 於 Firebase Consol 建立你的專案
3. 於 Firebase 專案加入你的iOS 專案
4. 下戴 GoogleInfo.plist 並拉進Xcode 專案中
5. 使用 Cocoa Pods 或 SPM 拉 Firebase SDK 進入你的 iOS 專案中
6. 修改 Appdelegate.swift 加入 Firebase Config
7. 開始撰寫你的 Firebase APP

Firebase 登入的方式

- 匿名登入
  - 確定為 APP 使用者，不是路人駭客
- 社群整合登入
  - Firebase 支援 Google 帳號(Gmail) FaceBook, Microsoft, Twitter, GitHub…..等
- 帳號密碼登入
  - 使用電子郵件/密碼登入
  - 自行管理新建帳號，忘記密碼等
- 電信登入
  - 手機簡訊驗證登入
- 電子郵件登入
  - 發出電子郵件邀請的方式登入

整合社群登入

都用了社群整合, 為什麼還需要 Firebase?

1. 統一帳號的管理
2. 個人信息的進階安全性

OAuth 認證概念

開放授權(OAuth)是一個開放標準, 允許使用者讓第三方應用存取該使用者在某一網站上儲存的私密的資源(如相片, 影片, 聯絡人列表), 而無需將使用者名稱和密碼提供給第三方應用。OAuth 允許使用者提供一個權杖, 而不是使用者名稱和密碼來存取他們存放在特定服務提供者的資料。

實作社群登入

Google 帳號登入

由於是同一個集團下的產品, Firebase 與 Google 帳號有許多自動化設定, 所以先由這個產品作起

Facebook 帳號登入

Facebook 是目前台灣地區最多人用的社群媒體, 但它並非 Google 同一個公司, 實作起來就不如 Google 帳號方便。必需要有一個 FB 的開發者帳號與許多設定, 若能完成, 其他的第三方就應該不成大問題了

使用線上文件完成實作

電子郵件帳號密碼登入

先要有帳號才能登入, 所以先製作【建立帳號】頁面

建帳號 Auth.auth().createUser(withEmail: email, password: password)

登入 Auth.auth().signIn(withEmail: email, password: password) {
}

除了登入以外

完整的帳號管理, 除了登入之外, 還有許多事要做, 還好很多事 Firebase 已經幫我們處理了, 但我們仍需做一些事來整合

驗證電子郵件

發送認證郵件, 以確認電子郵件是真的

忘記密碼

使用者不記得密碼時, 透過電子郵件重設密碼