

2016.09.26



Write the code. Change the world.

SWIFT GIRLS (IOS)

PART 6

Firestore x ChatRoom 最後大進擊！

by 雅方

2016.09.26

SWIFT GIRLS (IOS)



專為Swift新手展開的第一次”探勘“行動

Swift Girls aims to give tools for girls and women to build their ideas.

希望不斷推廣而有更多女生一同透過交流學習，一起了解ios開發並能互相切磋，歡迎對Swift語言有興趣的
一起加入！

雅方比女人更了解女人的雅方

也是雅方冰淇淋的雅方

雅方羊肉爐的雅方

by 雅方

SWIFT GIRLS (IOS)

學習資源 線上網上各式各樣影片文件無所不有！！！！

在這裡 可以有老師教練們給你來問問題

可以有夥伴跟你一起討論 (給你問白癡的問題!?XD)

重點是 能有一起學習的夥伴 一起成長 :)

網上線上 都比不上實際面對面操作實作與立即性的討論

看大大們debug 回家後你們也可以互相討論 自己debug :)

SWIFT GIRLS (IOS)-PART 6

發問解答工具

1. Github(程式碼儲存庫), stackOverFlow
2. google 大神
3. 社群發問

範例Github:

https://github.com/Avonee/swift-girls-meetig_teaching

2016.09.26

SWIFT GIRLS (IOS)-PART 6



by 雅方

2016.09.26

改變世界



Write the code. Change the world.

by 雅方

2016.09.26

START!!!

by 雅方

資料庫 (Databases)

- 資料庫是由**資料表 (Tables)** 所形成的集合
資料表則是一筆一筆的記錄 (**Records 或是 Rows**) 所成的集合，每筆記錄都包含不同的欄位 (Fields 或是 Columns) 。
- 資料的儲存架構又分為多種類型，較常見的有 4 種：
 - **階層式**(Hierarchical Database)
 - **網狀式**(Network Database)
 - **關聯式**(Relational Database)
 - **物件導向式**(Object-Oriented Database)

關聯式(Relational Database)

- 2 維的矩陣來儲存資料—(將資料儲存在表格的欄、列之中)
- 可從一個資料表中的欄位, 透過資料表的關聯, 而找到另一個資料表中的資料。

訂單序號	日期	客戶編號	是否付款
1	2005/7/1	6	1
2	2005/7/1	3	1
3	2005/7/3	2	0

訂單資料表

客戶編號	客戶名稱	聯絡人	性別	地址
1	十全書店	陳圓圓	女	台北市
2	大發書店	陳季暄	女	台北市
3	好看書店	趙飛燕	女	台中市

客戶資料表

經由客戶編號欄的關聯, 可知道
訂單序號 2 的客戶為好看書店

- 目前市場上是以關聯式資料庫使用最廣泛。

Microsoft SQL Server、MySQL、PostgreSQL、Access..., 都是屬於關聯式資料庫管理系統 (Relational DBMS, 縮寫為RDBMS (Relational Database Management System))。

Firestore Realtime Database

關聯式資料庫的通用查詢語言是SQL

Firestore Realtime Database 是一個雲端 NoSQL 資料庫

採用只有 2 欄，稱為雜湊表（HASH TABLE）的方式儲存。
1 欄是關鍵字（KEY），另1欄是值（VALUE），作為查詢的資料結構。
把KEY值通過一個函數的計算，映射到表中一個位置來查詢記錄，這加快了查找速度。

p.s. SQL 使得源源湧入的資料無法及時反應，可擴展性低。
造成傳統關聯式查詢查詢相對緩慢、效益相對低。

Firestore Database

「關聯式資料庫」 — 較適合一般機構的管理資訊系統。

欄位必須要定義「資料格式」，如字串、整數、實數（含小數）、貨幣、日期...等

NoSQL/Key-Value — 適用巨量網站

不必定義，還包括「資料格式」設定、系統訊息、資料位址、...
什麼都可以，而且，沒有順序問題，可以有空白格，也就是沒有
「資料結構」問題。

Firestore

Document — Realtime Database

The screenshot shows the Firebase documentation website. The top navigation bar includes links for Home, Features, Pricing, Docs (selected), Customers, and Support. A search bar and a 'Go to console' link are also present. Below the navigation bar, the 'Documentation' section is active, with sub-links for Overview, Guides (selected), Reference, Samples, and Libraries. A 'Send Feedback' link is on the right. The left sidebar lists various Firebase services, with 'Realtime Database' expanded to show 'iOS' guides, including 'Get Started'. The main content area is titled 'Installation & Setup on iOS' with a 5-star rating. It describes the Realtime Database as a cloud-hosted database that synchronizes data in real-time. Below this, a 'Prerequisites' section lists two steps: installing the Firebase SDK and adding the app to the Firebase project in the console. A note at the bottom mentions SDK updates. A right sidebar provides a 'Contents' table of contents for the document.

Documentation

OVERVIEW GUIDES REFERENCE SAMPLES LIBRARIES SEND FEEDBACK

DEVELOP

Cloud Messaging

Authentication

Realtime Database

Introduction

iOS

Get Started

Structure Data

Save Data

Retrieve Data

Enable Offline Capabilities

Android

Web

Installation & Setup on iOS

☆☆☆☆☆

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Android, iOS, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

Prerequisites

1. [Install the Firebase SDK.](#)
2. Add your app to your Firebase project in the [Firebase console](#).

★ **Note:** Our SDKs have changed. If you are upgrading from a 2.X version of the Firebase SDK, please read our [upgrade guide for iOS](#).

Contents

- Prerequisites
- Add Firebase Realtime Database to your app
- Configure Firebase Database Rules
- Set up Firebase Realtime Database
- Prepare for Launch
- Next Steps

參考<https://firebase.google.com/docs/database/ios/start>

by 雅方

2016.09.26

實作GO!

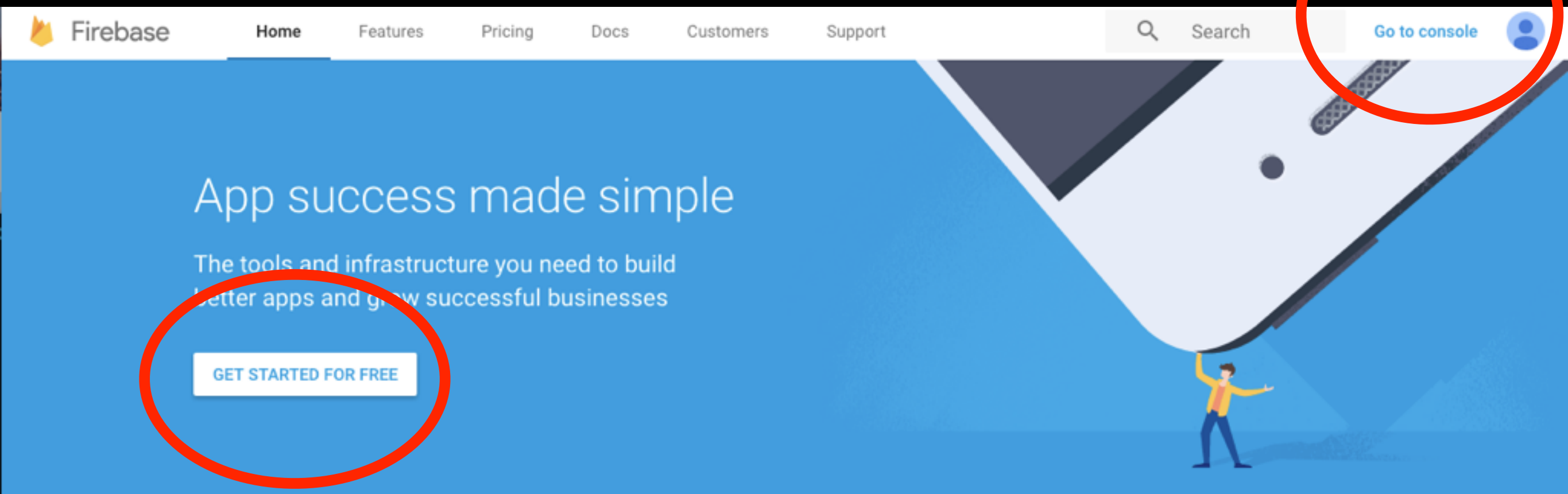
by 雅方

2016.09.26

Firebase

註冊登入 & Get Start

<https://firebase.google.com/>



Firebase

建立新專案



Firebase

建立新專案

×

建立專案

專案名稱

我的超酷炫專案

國家/地區 ?

台灣

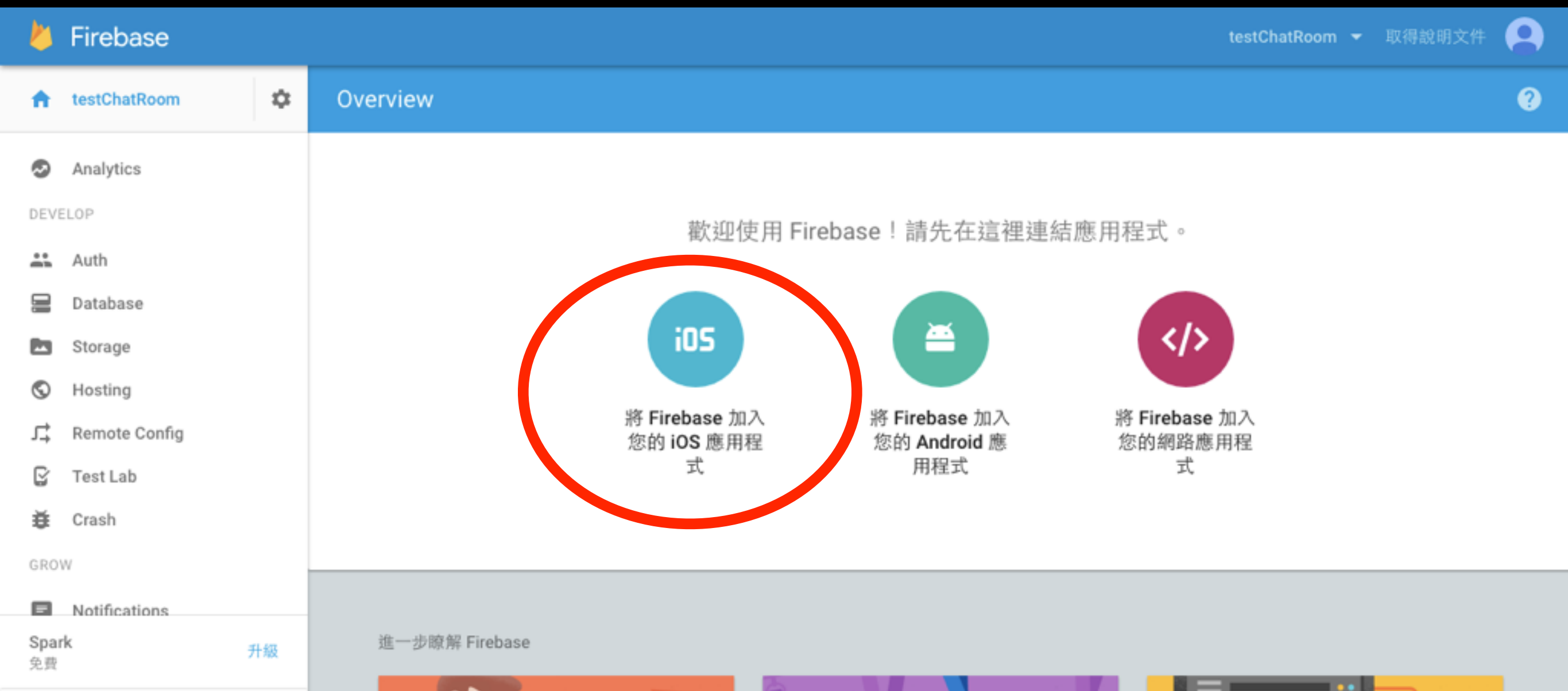
根據預設，Firebase Analytics 資料可用於改善其他 Firebase 功能和 Google 產品的成效。您隨時可以在設定中變更 Firebase Analytics 資料的共用方式。[瞭解詳情](#)

取消

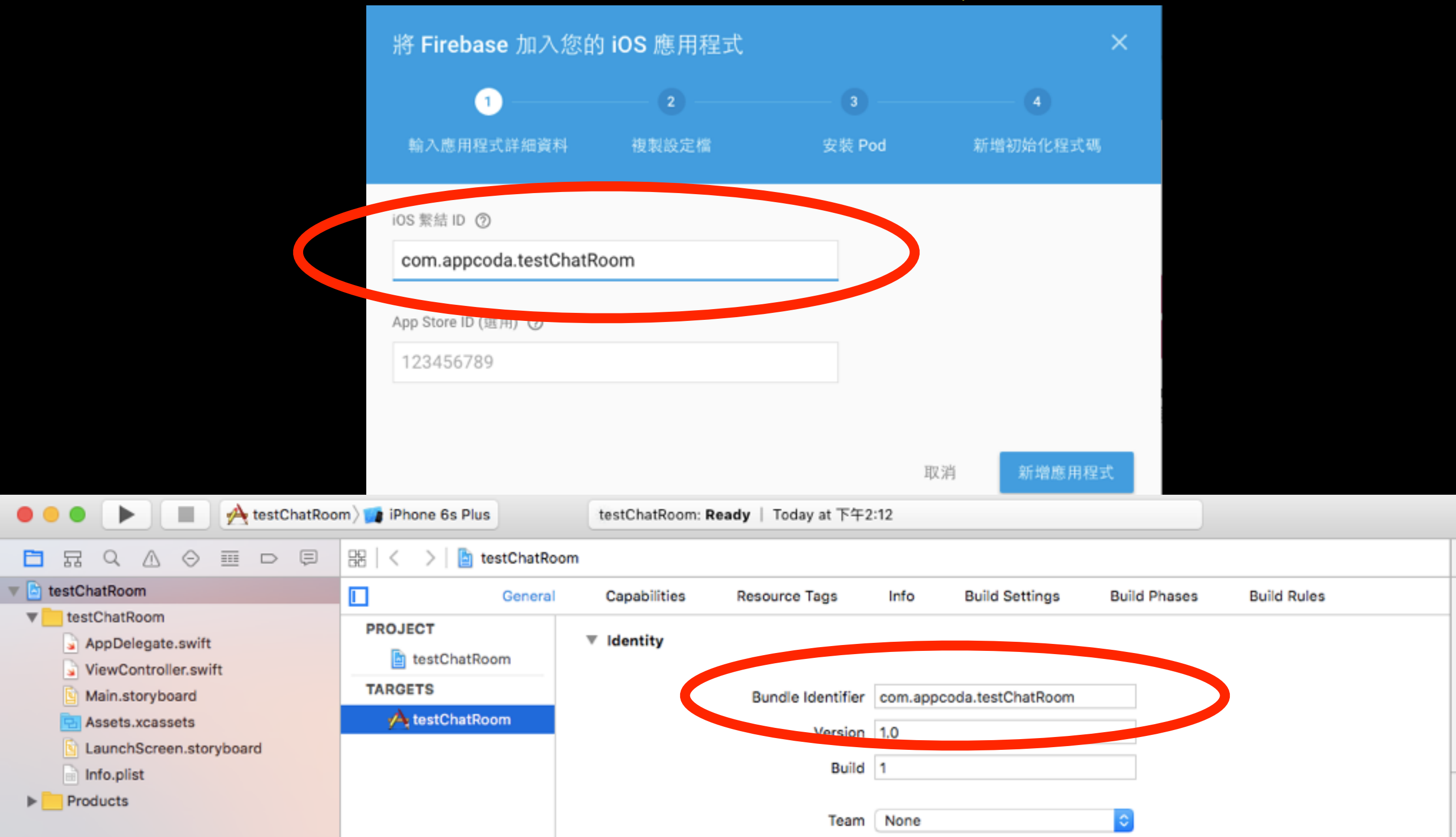
建立專案

- 印度
- 厄利垂亞
- 厄瓜多
- 古巴
- 台灣
- 史瓦濟蘭
- 吉布地
- 吉爾吉斯

Firestore 加入到 iOS 設定



Firestore 加入到iOS設定



Firebase 加入到iOS設定

將 Firebase 加入您的 iOS 應用程式

1

2

3

4


輸入應用程式詳細資料

複製設定檔

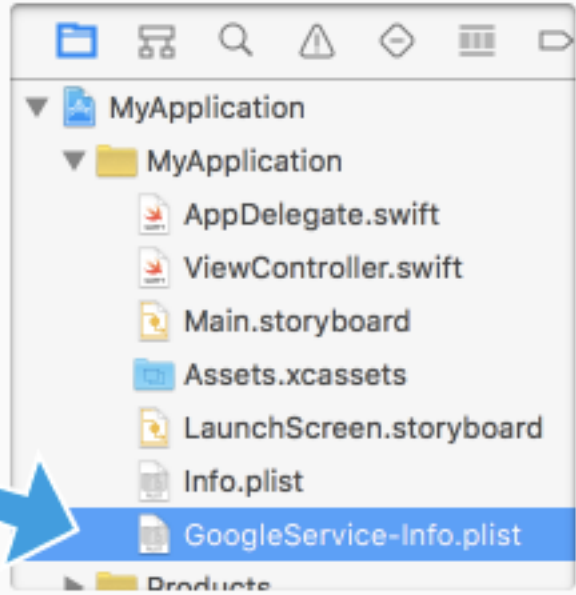
安裝 Pod

新增初始化程式碼

將您剛才下載的 **GoogleService-Info.plist** 檔案移到 Xcode 專案的根目錄中，並將該檔案新增到所有目標中。



GoogleService-Info.plist



The image shows an Xcode project file browser for a project named 'MyApplication'. The file list includes 'AppDelegate.swift', 'ViewController.swift', 'Main.storyboard', 'Assets.xcassets', 'LaunchScreen.storyboard', 'Info.plist', and 'GoogleService-Info.plist'. A blue arrow points from the 'GoogleService-Info.plist' file icon in the previous step to this file in the Xcode project browser.

先前已新增過 Pod 和初始化程式碼了嗎？
[直接進入主控台](#)

繼續

Firebase 加入到iOS設定

[使用前安裝]—打開**終端機**

安裝套件cocoapods 要在自己的資料夾內
若第一次安裝cocoa pods 就要先打入

```
sudo gem install cocoapods
```

```
$ cd your-project-directory
```

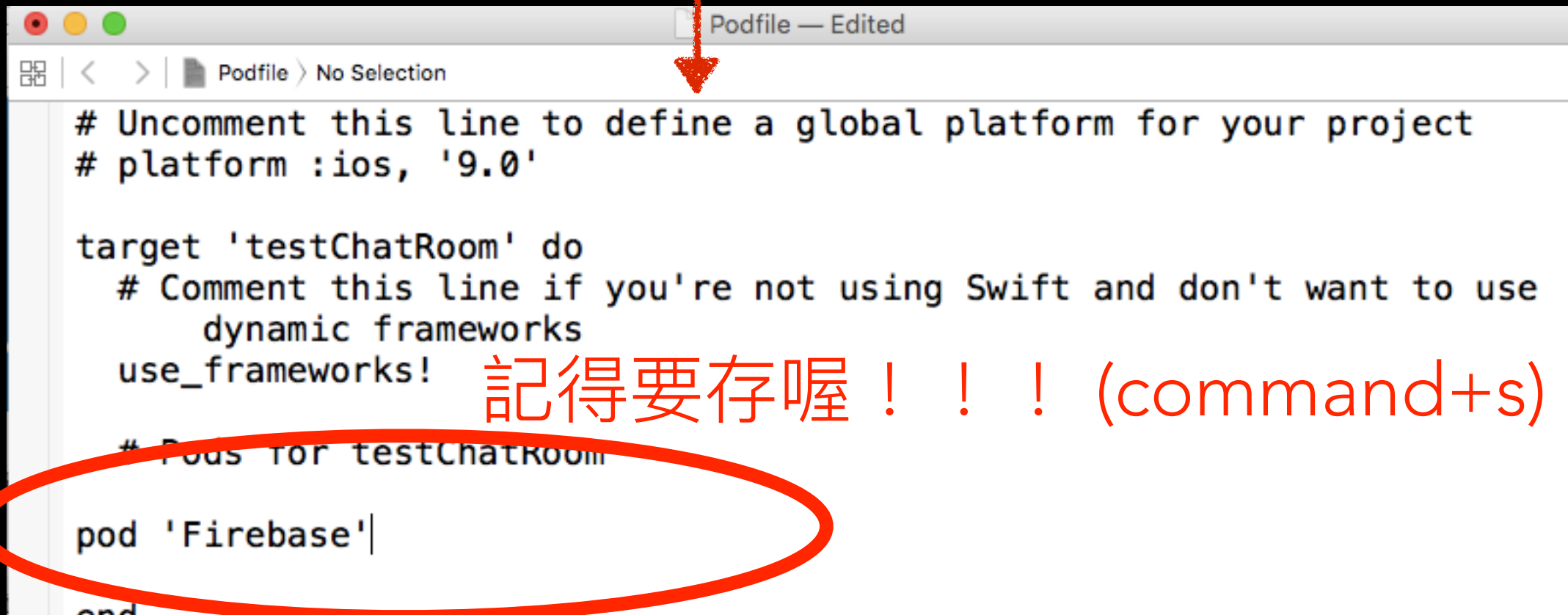
```
$ pod init
```

```
$ open -a Xcode Podfile
```

終端機範例

```
testChatRoom — -bash — 80x24
18:56:29 on ttys000
rubl333$ cd Desktop/
Desktop rubl333$ cd testChatRoom/
testChatRoom rubl333$ pod init
testChatRoom rubl333$ open -a Xcode Podfile
testChatRoom rubl333$ pod install
```

4.



```
Podfile — Edited
Podfile > No Selection
# Uncomment this line to define a global platform for your project
# platform :ios, '9.0'

target 'testChatRoom' do
  # Comment this line if you're not using Swift and don't want to use
  # dynamic frameworks
  use_frameworks!
  # Pods for testChatRoom
  pod 'Firebase'
end
```

記得要存喔！！！！(command+s)

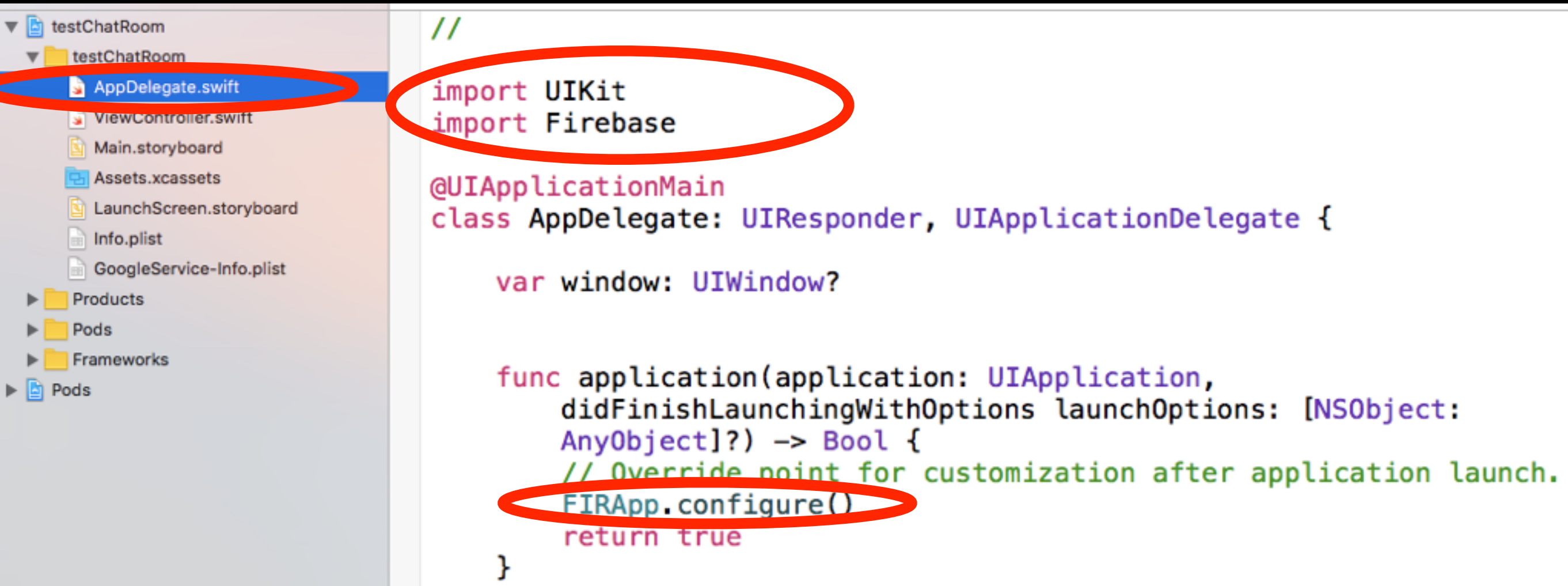
5.

Firebase 加入到iOS設定

依最後設定指示...

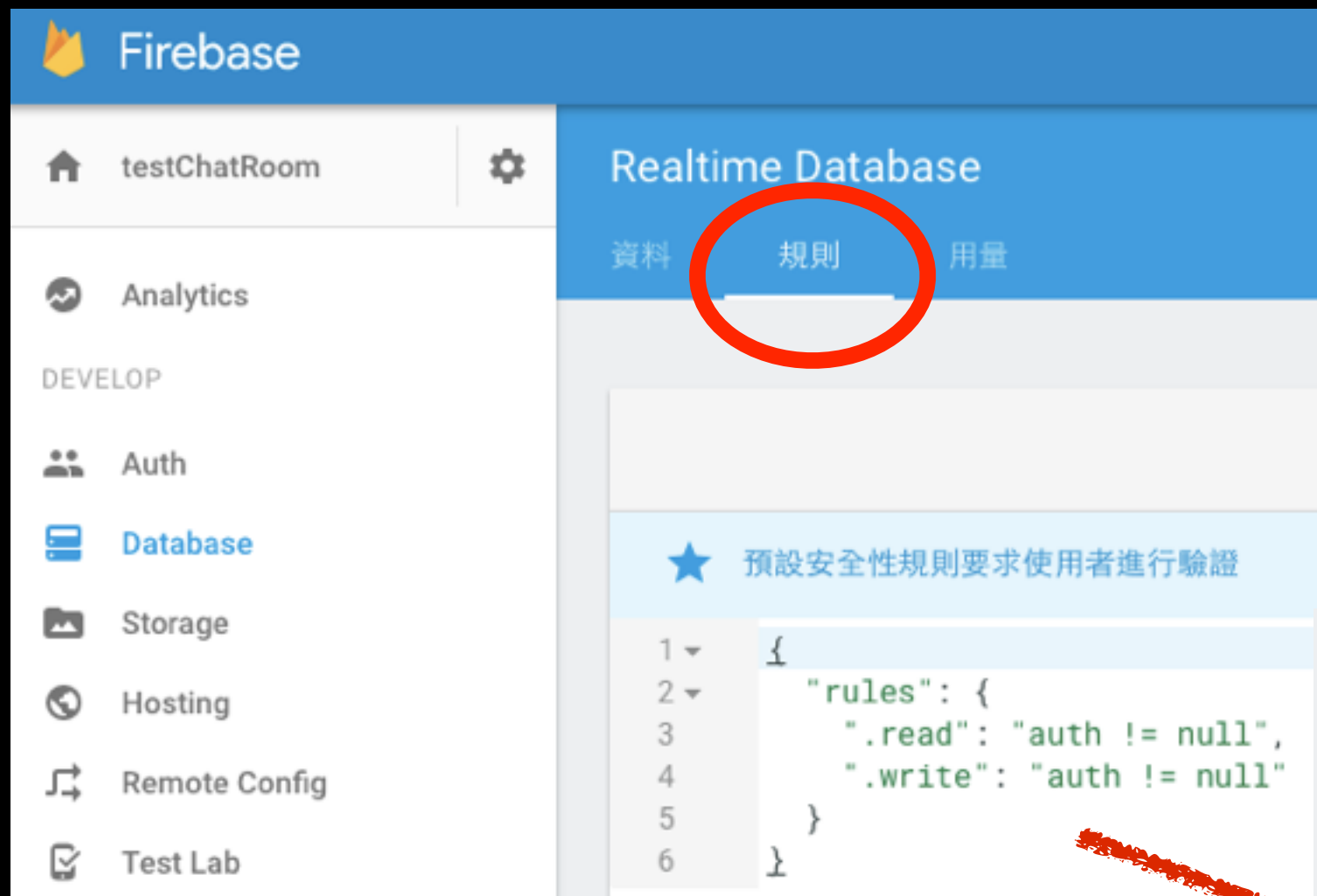


Firestore 加入到iOS設定



恭喜設定完成啦！！！！

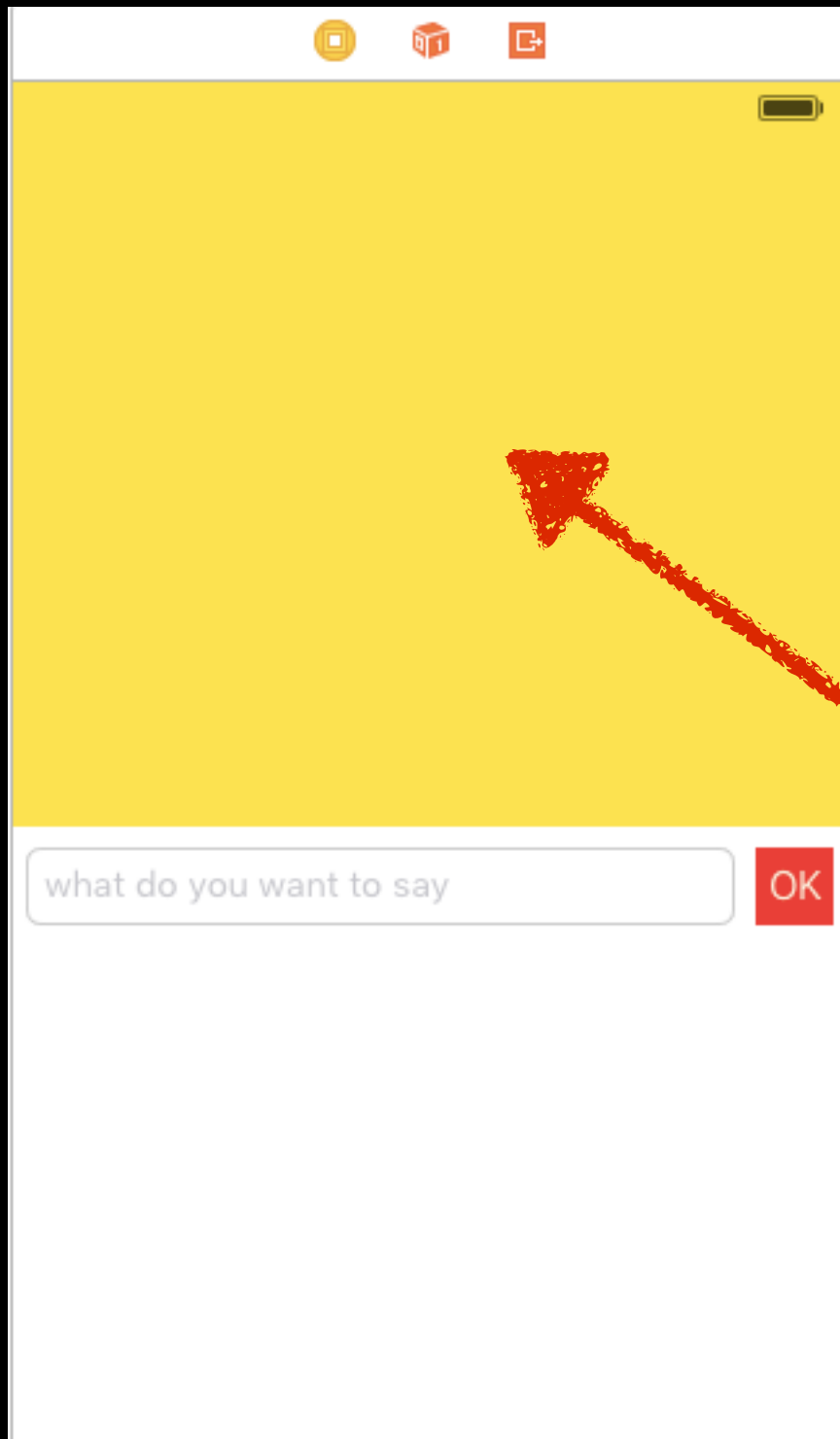
Firestore 允許權限



true! true!

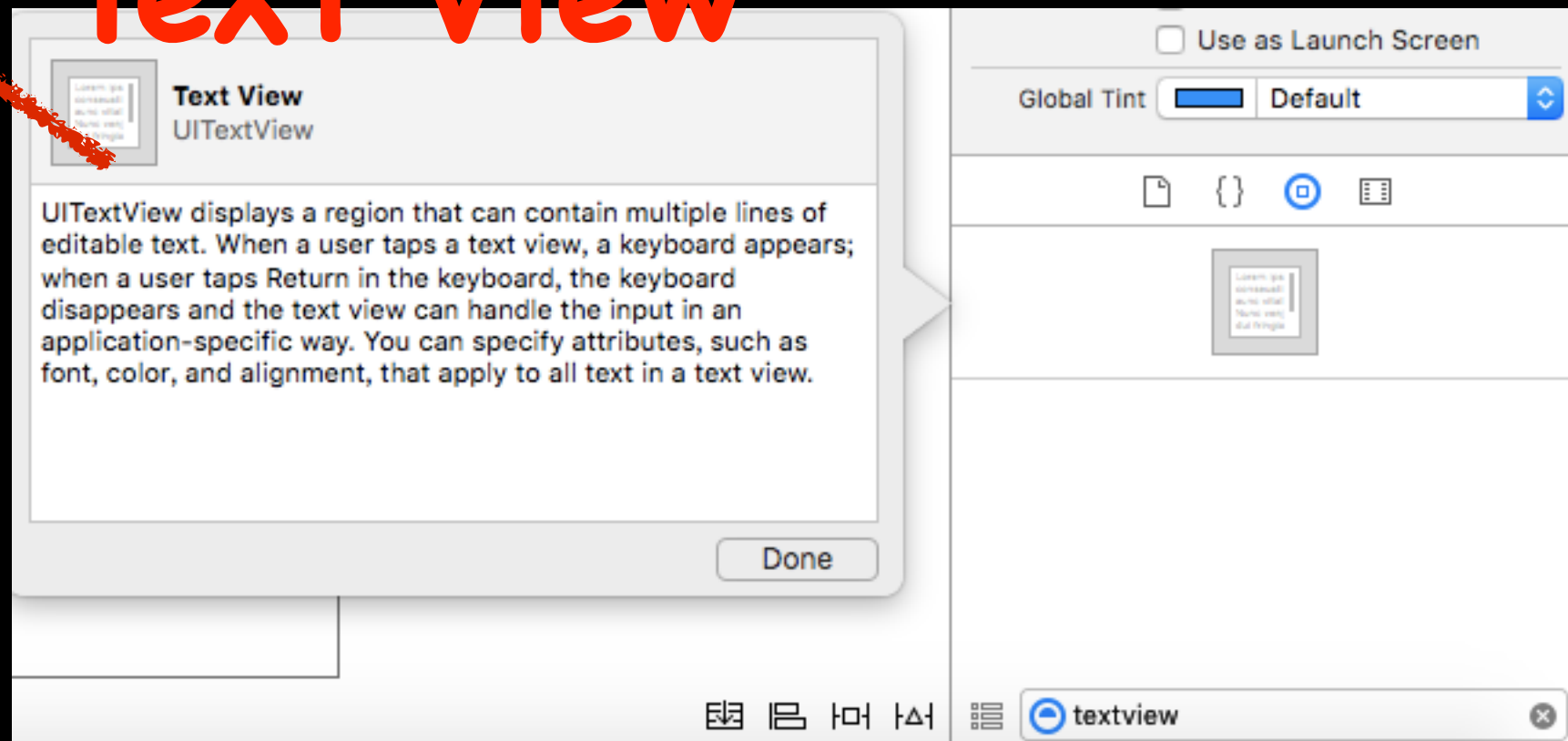


Storyboard Layout



```
@IBOutlet weak var showTalk: UITextView!//顯示對話框  
@IBOutlet weak var textInput: UITextField!//對話輸入筐  
@IBOutlet weak var okButton: UIButton!//新增輸出按鈕
```

Text View



Install Firebase Database

★ 打開終端機

跟前面一樣步驟 在podFile新增一行,再install

```
pod 'Firebase/Database'
```

● import

```
import FirebaseDatabase
```

● class 內宣告databaseReference

```
var ref: FIRDatabaseReference!
```

Database .Write (寫入)

```
@IBAction func okButtonClick(sender: AnyObject) {  
    ref = FIRDatabase.database().reference()  
  
    //      ref.child("chat").setValue(["name": "Avon", "say": textInput.text!])  
  
    //寫入名字和對話  
    let key = ref.child("chat").childByAutoId().key  
    let post = ["name": "\(self.nameGet!)",  
                "say": textInput.text!]  
  
    //另一種顯示更新項目的寫法  
    //      let childUpdates = ["/chat/\(key)": post,  
    //                          "/user-chat/\("Avon")/\(key)"/": post]  
  
    //      let childUpdates = ["/chat/\(key)": post]  
    let childUpdates = ["\(key)": post]  
  
    FIRDatabase.database().reference().updateChildValues(childUpdates)  
  
}
```

Database .Read (讀取)

Retrieve Data

★在ViewDidLoad

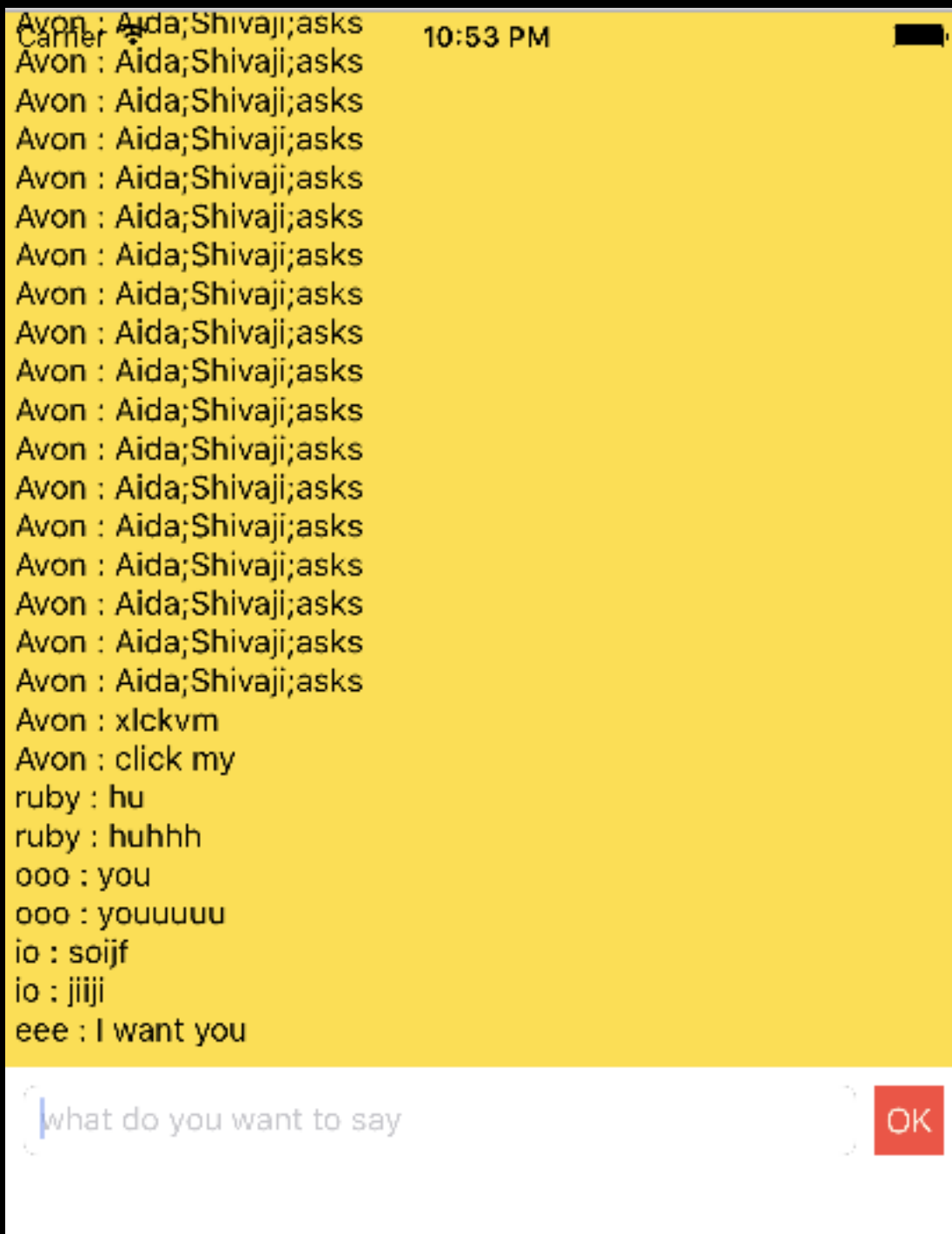
```
//.Read 讀取
_ = FIRDatabase.database().reference().observeEventType(.ChildAdded, withBlock: { (snapshot) in

    if snapshot.value != nil{
        //如果是有資料的狀況

        //讀取名字
        let getName = snapshot.value!.objectForKey("name") as! String
        //讀取對話
        let getSay = snapshot.value!.objectForKey("say") as! String
        //顯示名字+對話 全部內容
        self.showTalk.text = self.showTalk.text + "\\(getName)" + " : " + "\\(getSay)" + "\\n"
        //當完成以上動作之後，就清空剛填寫的對話欄，這樣即可立即在空的對話欄內再次輸入想說的話
        self.textInput.text = ""
    }

})
```

完成了!!!



★ Home Work問題：
如何不用一直往下滑，
就能讓每次新的一行都能跳到最下面呢？
Try ~~~ :D

Q & A

Swift girls (iOS) 講義與範例全：

https://github.com/Avonee/swift-girls-meeting_teaching