

Firebase Cloud Messsing(雲端訊息)

Firebase 雲端訊息(FCM)是一個跨平台訊息傳送的解決方案，這方案讓開發者不用任何費用，可以很穩定的傳遞訊息。

使用FCM,你可以通知客戶端的app，新的email或其它的資料已經可以使用。你可以傳送通知的訊息給使用者，讓使用者有所反應。例如即時訊息。一個訊息的傳送最多到4KB大小。

主要的能力

傳送通知訊息或資料訊息	傳送通知訊息，讓使用者看到顯示的訊息內容。或傳送資料訊息，由app程式碼完全決定接收完成後要做出何種動作。
多種訊息目標	使用傳送訊息的3種方法中的任一種方法。單一個手機或一組手機或一組有訂閱相同主題的手機。
從使用者手機傳送訊息	由使用者手機傳送確定通知，聊天和其它訊息傳回至伺服器。

如何實作？

一個FCM的實作，包含2個主要的元件，送訊息和收訊息。

- 1.一個信任的環境，例如Firebase 的雲端函式或一個appServer。可以傳送訊息。
- 2.一個ios,Andorid或網頁(javascript)的使用者端，可以接收訊息。

開發者可以傳送訊息藉由Admin SDK 或者是HTTP和XMPP APIs。為了測試或傳送行銷或邀請訊息給目標者或分析資料，也可以使用Notification composer。



實作流程

設定 FCM SDK	根據不同的平台，設定Firebase 和 FCM
------------	--------------------------

開發使用者端的APP	增加訊息處理器，每個訂閱主題的邏輯或其它的功能到使用者端。在開發階段，可以使用Notifications composer，非常簡單的傳送測試資料。
開發app的Server端	判斷是否使用Admin SDK 或者只有Server 協定的其中一個來建立發送訊息的邏輯。驗證使用者邏輯，建立送出的需求邏輯，處理回應邏輯，建立在信任的環境上。註解:如果你想使用從app傳送upstream訊息，你必需使用XMPP。Cloud Functions沒有支援這種持續連結。

關於FCM 訊息

Firebase 雲端訊息提供多種類型的訊息選項和能力。我這篇文章主要是幫助開發者了解通知訊息和資料訊息。讓開發者了解，如何使用FCM的選項功能。

訊息類型

使用FCM,開發者可以傳送2種類型的訊息給使用者端的App:

- 通知訊息，想像就是顯示出來的訊息。
- 資料訊息，資料可以被使用者App處理。

一個通知訊息是一個更輕量的選項，被限制最多2KB和事前先定義一組使用者可以看到的keys。資料訊息讓開發者可以傳送最多4KB的key-value資料給app。通知訊息能包含一個選項資料，這個選項資料必需使用者按在通知上時，才會被傳遞給App。

	使用情境	如何送出
通知訊息	FCM自動顯示訊息到使用者端的手機，為了使用者的App。通知訊息有一個事前已經被定義可以讓使用者看到的Key和一個包含key-value的選項資料。	1.在一個信任的環境才可以送出通知。例如Cloud Functions或你的app伺服器,app伺服器必需使用Admin SDK或有設定notification key的HTTP和XMPP APIs。可以有一個傳送資料的選項。大部份是 collapsible
資料訊息	使用者端app會處理資料，資料訊息只有key-value的資料。	1.在一個信任的環境才可以送出通知。例如Cloud Functions或你的app伺服器,app伺服器必需使用Admin SDK或有設定data key的HTTP和XMPP APIs。可以是 collapsible或non-collapsible。

當你想要FCM處理在使用者App上顯示訊息，請使用通知訊息。當想要使用者的App處理資料，請使用資料訊息。

FCM可以送出通知訊息並可以包含選項資料。在這種情況下，FCM負責顯示通知工作而app負責處理選項資料。

註解:當app在背景模式時，如果僅想要送自訂的key-value訊息給ios手機。設定自訂key-value資料到data key和設定content_available到true;

通知訊息

當為了測試，行銷，邀約，你可以透過使用Notifications composer送出通知訊息。

使用Admin SDK或FCM協定的程式碼，送出通知訊息，必需設定notification key和事前定義的key, 自訂的value值，為了可以在使用者端顯示。如範例，這裏有JSON格式的通知訊息。可以預期使用者將在手機上看到下面訊息，title是”Portugal vs. Denmark” 和 “great match!”：

```
{
  "to" : "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
  "notification" : {
    "body" : "great match!",
    "title" : "Portugal vs. Denmark",
    "icon" : "myicon"
  }
}
```

當app是在背景時，通知訊息被傳遞到通知視窗。app在前景模式收到訊息時，處理訊息的function是:didReceiveRemoteNotification:..

查看更多為了建立通知訊息而事先定義的key，請查詢notification payload support。

資料訊息

設定使用自訂的key-value資料加入至data key，然後送至使用者App。資料訊息最大可以到4KB。

如範例，有一個JSON格式的訊息，這些資料被包在data key內。使用者的app可以預想會收到這些資料。

```
{
  "to" : "bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
  "data" : {
    "Nick" : "Mario",
    "body" : "great match!",
    "Room" : "PortugalVSDenmark"
  },
}
```

在iOS上，app不是在前景時，FCM會暫時儲存資料，當到前景時，才會傳遞資料，並且會建立一個FCM的connection。在Android上,app會在onMessageReceived()內接收資料和根據key處理資料。

ios和android不同的特性：

- 在android上，資料可以被接收，並且啟動要開啟的頁面。
- 在ios上，這資料在didReceiveRemoteNotification:內

通知訊息包含自訂的資料

使用程式或Firebase console都可以送出通知訊息並且包含有自訂的key-value的資料。在Notifications composer的Advanced options內，使用自訂的Custom data就可以送出資料。

當App接收到訊息，這訊息包含通知和資料。app所做的反應基本上是依據目前app是在背景或是前景的狀態。在接受到訊息的當下，背景和前景所做的動作如下：

- 當在背景時-手機在通知視窗接收通知，只有使用者點下通知視窗，app才會開始處理訊息帶過來的資料。
- 當在前景時-app同時接收到訊息和訊息帶過來的資料。

這裏有JSON格式包含 2 個key,notification key和這data key:

```
{
  "to" : "APA91bHun4MxP5egoKMwt2KZFBaFUH-1RYqx...",
  "notification" : {
    "body" : "great match!",
    "title" : "Portugal vs. Denmark",
    "icon" : "myicon"
  },
  "data" : {
    "Nick" : "Mario",
    "Room" : "PortugalVSDenmark"
  }
}
```

不重疊和重疊的訊息

一個不重疊的訊息代表每個訊息是被個別的傳遞給手機。一個不重疊訊息傳遞一些有用的內容。而不是傳一個訊息，讓app再去聯絡伺服器並且取得資料。通知訊息預設是不重疊的訊息，其它的是重疊訊息。

FCM不保證傳遞的順序

一些平常使用不重疊訊息的案例，就是聊天訊息和重要訊息。如範例，在一個app內想要傳送每一個訊息，因為每一個訊息的內容一定會不一樣。

ios App內設定Firebase Cloud 訊息服務

ios 使用者App可以使用2種互補的方式來實作Firebase Cloud Messaging:

- 接收基本的推播訊息，訊息是透過Firebase Cloud Messaging APNs介面推播出來，最多到2KB。
- 送出向上串流或接收向下串流的資料，最多到4KB。

使用swift建立使用者端的程式碼，建議開發者使用FIRMessaging的API。

在Firebase Cloud Messagging內的方法轉換

FCM API 執行方法轉換在2個主要區域:對應你的APN token到the FCM註冊的token和在下載訊息獲取分析資料。

開發者不喜歡使用方法轉換，可以藉由在info.plist內，增加一個FirebaseAppDelegateProxyEnabled設定為NO來關閉這個功能。將會提供有開啟和關閉此功能的範例。

增加Fiebase到IOS Project

如果已經在app加入Firebase，這個章節你可能已經完成。為了一些FMC的特定功能，您需要上傳您的APN的憑證和註冊遠端的通知。

事前的準備

在開始前，你必需現設定一些環境：

Xcode 7.0以上

Xcode專案，目標是IO7以上

app bundle名稱

CocoaPods1.0.0以上

為了雲端訊息：

- 一個實體手機
- 一個APNs憑證和推播通知打開
- 在Xcode內的APP->Capabilities內打開Push Noifications

如果您沒有XCode專案，而且只是想要測試，你可以下載quickstart sample。如果是使用一個squickstart，記得在專案設定中取行bundle識別名稱，在下個章節將會使用到。

註解：如果你是使用Firebase SDK 2.0更新，請參考 [upgrade gide for ios](#)到開始。

增加Firebase 到App

現在增加Firebase到你的app，實作這個動作時，將需要在專案內加入Firebase的專案和Firebase的設定環境檔。

1. 如果還沒有Firebase 專案,則在Firebase控制台建立Firebase 專案。如果已經有Google專案連結到手機的app，請使用import Google Project。不然請使用Create New Project。
2. 使用Add Firebase to your IOS app和跟隨它的步驟。如果輸入的是一個已經出現過的Google專案，這些步驟將會自動執行。
如要下載 iOS 應用程式的設定檔，請按照以下步驟進行：
 1. 登入 Firebase，然後開啟專案。
 2. 點選 ，然後選取 [專案設定]。
 3. 在「您的應用程式」卡片中，從清單中選取您要下載設定檔的應用程式的繫結 ID。
 4. 點選 [ GoogleService-Info.plist]。
3. 當出現輸入你的app's bundle ID時，這個動作非常重要。專案App內的bundle ID必需要Firebase console專案的bundle ID是一樣的。
4. 最後必需下載GoogleService-Info.plist檔。
5. 全部動作完成後，將GoogleService-Info.plist加入到Xcode專案的根目錄內

加入SDK

如果設定新的專案，必需要安裝SDK。或許在建立Firebase project時，已經完成安裝。我們建議使用CoCoadPods來安裝函式庫。可以藉由此安裝方式來安裝CoCoadPods。

<https://cocoapods.org/>

如果您準備下載和執行快速開始範例的其中一個範例,這Xcode專案常未全部完成，必需要安裝pods和下載googleService-info.plist檔。如果要整合Firebase library進入到我們的專案內，我們必需將所需使用的Firebase library加入到我們的專案內。

- 1.如果沒有Xcode專案，請建立一個。
- 2.如果沒有PodFile，請建立一個。

```
$ cd your-project directory
$ pod init
```

- 3.增加想要加入的Pods。設定在Podfile內，設定如下：

```
pod 'Firebase/Core'
pod 'Firebase/Messaging'
```

這將加入事前必需準備好的Firebase libraries和在ios app內執行Firebase Analytics。下面有目前可以加入的pods的列表，並且有對應到特定的功能。

4.安裝pods和使用Xcode打開.xcworkspace file

```
$ pod install  
$ open your-project.xcworkspace
```

5.從Firebase console下載 GoogleService-Info.plist，並且將它加入到App內。

上傳你的APNs憑證

上傳您的APNs憑證到Firebase。如果還沒有APNs憑證，請看提供APNs SSL 憑證。

- 1.進入Firebase控制台，進入專案，選擇工具按鈕，選擇Project Settings,然後選擇雲端訊息標籤。
- 2.在開發者憑證或產品憑證中選擇Upload Certificate。最少需要一個。
- 3.選擇.p12檔案和提供密碼。確定憑證的bundle id和app的bundle是同一個。

在App內，初始化Firebase

你將需要在應用程式內增加Firebase初始化的程式碼。輸入Firebase模組和啟動一個分享的實體。

- 1.在UIApplicationDelegate內輸入Firebase

```
import Firebase
```

- 2.通常在應用程式的application:didFinishLaunchingWithOptions:方法內，調整FIRApp分享實體。

```
FirebaseApp.configure()
```

註冊遠端通知

在應用和式的起點或想要的時間點，在app內註冊遠端通知。呼叫registerForRemoteNotifications:


```

if #available(iOS 10.0, *) {
    // For iOS 10 display notification (sent via APNS)
    UNUserNotificationCenter.current().delegate = self

    let authOptions: UNAuthorizationOptions = [.alert, .badge, .sound]

    UNUserNotificationCenter.current().requestAuthorization(
        options: authOptions,
        completionHandler: {_, _ in })

    // For iOS 10 data message (sent via FCM)
    FIRMessaging.messaging().remoteMessageDelegate = self

} else {
    let settings: UIUserNotificationSettings =
        UIUserNotificationSettings(types: [.alert, .badge, .sound], categories: nil)
    application.registerUserNotificationSettings(settings)
}

application.registerForRemoteNotifications()
AppDelegate.swift

```

存取註冊的Token

在預設的情況下，App在開始執行時，藉由FCM SDK產生一個註冊的token。如果你的目標是一個單一的手機或一群手機，需要依靠存取這個token。

這個章節將描述如何取出token和如何管理token的改變。因為在啟動後，token有可能被改變，所以強烈建議取出最新的註冊token，除非你有特定需求的使用。

下面是token會改變的情況：

- 這app刪除了實體ID
- 這app被重新存在新的手機
- 這使用者移除和重新安裝app

- 這使用者清除app 資料

取出現在註冊的token

當你需要取出現在的token,呼叫FIRInstanceID.instanceID().token()!, 如果這個token尚未被產生, 則傳出null。

```
let token = FIRInstanceID.instanceID().token()!
```

管理token的產生

可以藉由NSNotificationCenter增加觀察者監聽kFIRInstanceIDTokenRefreshNotification, 然後在註冊的function內取出更新的token。假設tokenRefreshNotification是註冊的function:

```
func tokenRefreshNotification(_ notification:
Notification) {
    if let refreshedToken =
FIRInstanceID.instanceID().token() {
        print("InstanceID token: \(refreshedToken)")
    }

    // Connect to FCM since connection may have failed
when attempted before having a token.
    connectToFcm()
}
```

[AppDelegate.swift](#)

當token被產生, kFIRInstanceIDTokenRefreshNotification 會發生, 所以在現有的環境呼叫FIRInstanceID.instanceID().token()!, 取得token, 確保是可以使用的token。

在IOS10以上, 在app完成啟動前, 你必需指定你的代理人物件給UNUserNotificationCenter物件, 以便於接收顯示的通知。也同時給FIRMessaging物件, 以便於接收資料。例如在ios的app這些動作必需要在 applicationWillFinishLaunching:和applicationDidFinishLaunching:內完成。

轉換被關閉:對應APNs token給註冊的token

如果你關閉轉換method, 你必需明確的對應你的APNs token到FCM註冊的token。覆寫這個method

`didRegisterForRemoteNotificationsWithDeviceToken` 來取出APNs token,和然後呼叫 `setAPNsToken`。

在`setAPNsToken:type`:提供你的APNs token和這token的類型。確定這個type的值是正確的。在沙箱環境上使用`FIRInstanceIDAPNsTokenTypeSandbox`。在產品環境上使用 `FIRInstanceIDAPNsTokenTypeProd`。如果沒有設定正確的type，訊息將不會正確的傳送。

```
func application(application: UIApplication,
                 didRegisterForRemoteNotificationsWithDeviceToken
deviceToken: NSData) {
    FIRInstanceID.instanceID().setAPNsToken(deviceToken,
type: FIRInstanceIDAPNsTokenTypeSandbox)
}
```

當這實體ID的token被產生後，你可以使用當轉換method被開啟後，一樣的方式來存取token和監聽重新產生的事件。

供應APNs SSL 憑證

Firestore雲端訊息APNs介面使用Apple Push Notification service(APNs)到送最大到2k的訊息到ios App。包含app在背景的狀態。

要可以透過APNs傳送推播，必需要有下列條件：

- 要有一個推播的SSL憑證並且符合這個App ID。
- 要有這個App ID Provisioning profile。

建立SSL憑證

這個章節描述如何產生App ID的推播的SSL憑證。Firestore 雲端訊息將使用這個包含App ID憑證,傳送推播訊息到應用程式。

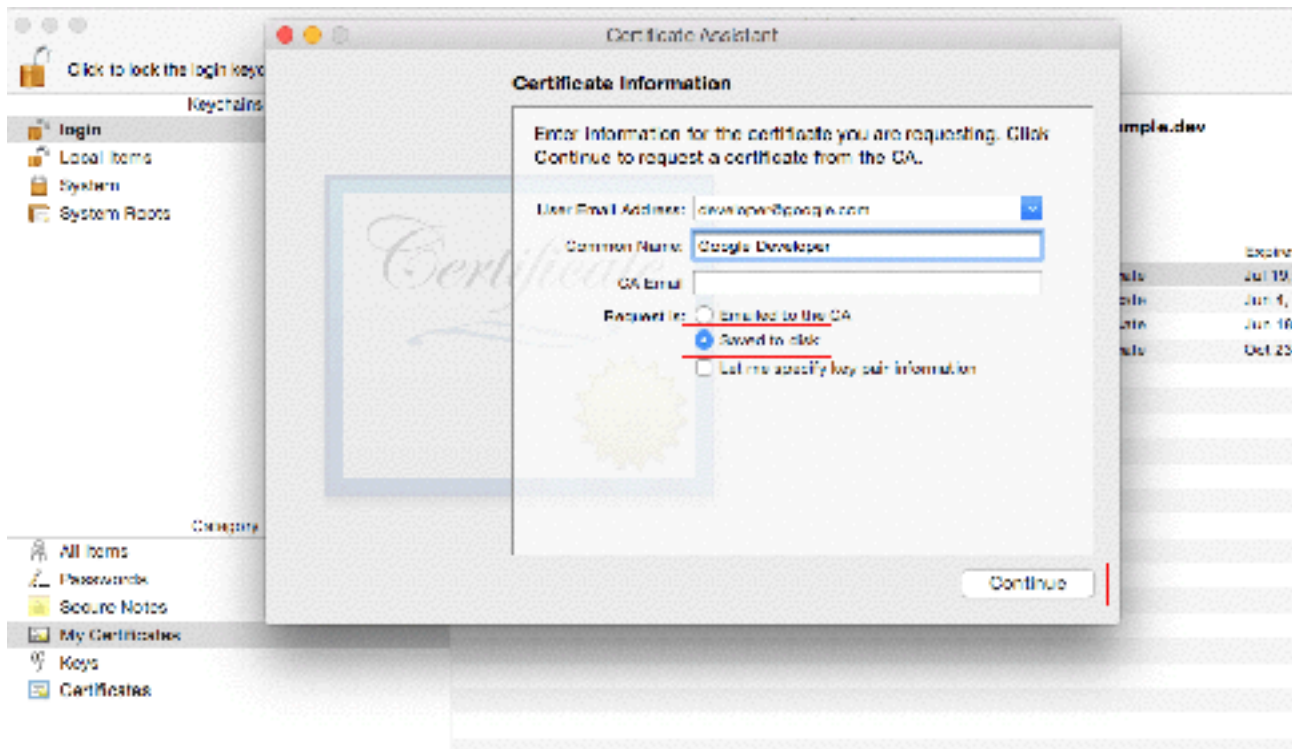
建立一個憑證註冊需求檔

建立一個SSL憑證，你需要憑證需求檔。

建立一個憑證需求檔，必需在一個Mac電腦上建立。

- 1.在mac上打開鑰匙圈存取。
- 2.打開鑰匙圈存取menu,鑰匙圈存取>憑證輔助程式>從憑證授權要求憑證
- 3.輸入你的email和姓名

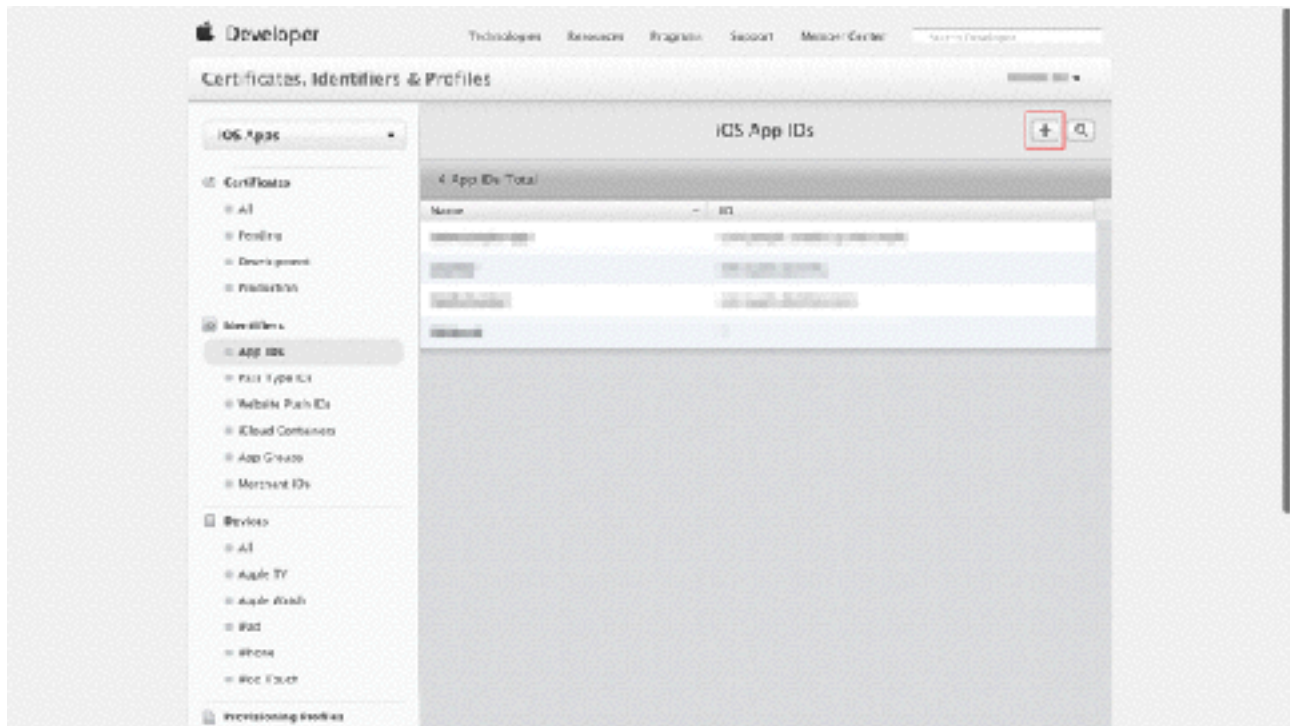
4.在已將要求欄位選取儲存到磁碟，然後點選繼續，儲存這.certSigningRequest檔案到Mac內。



建立一個App ID

如果你已經有一個你將要使用的APP ID，確定有一個明確的APP ID(它沒有包含*字元) 然後離開這個章節。

- 1.進入Apple開發者中心的網頁，並且登入。
- 2.導覽到Certificates,Identifiers和profiles
- 3.在頁面左上角下拉選項中，選擇IOS,tvOS,watchOS。然後導覽至Identifiers>APP IDs
- 4.按上+按鈕到建立新的APP ID

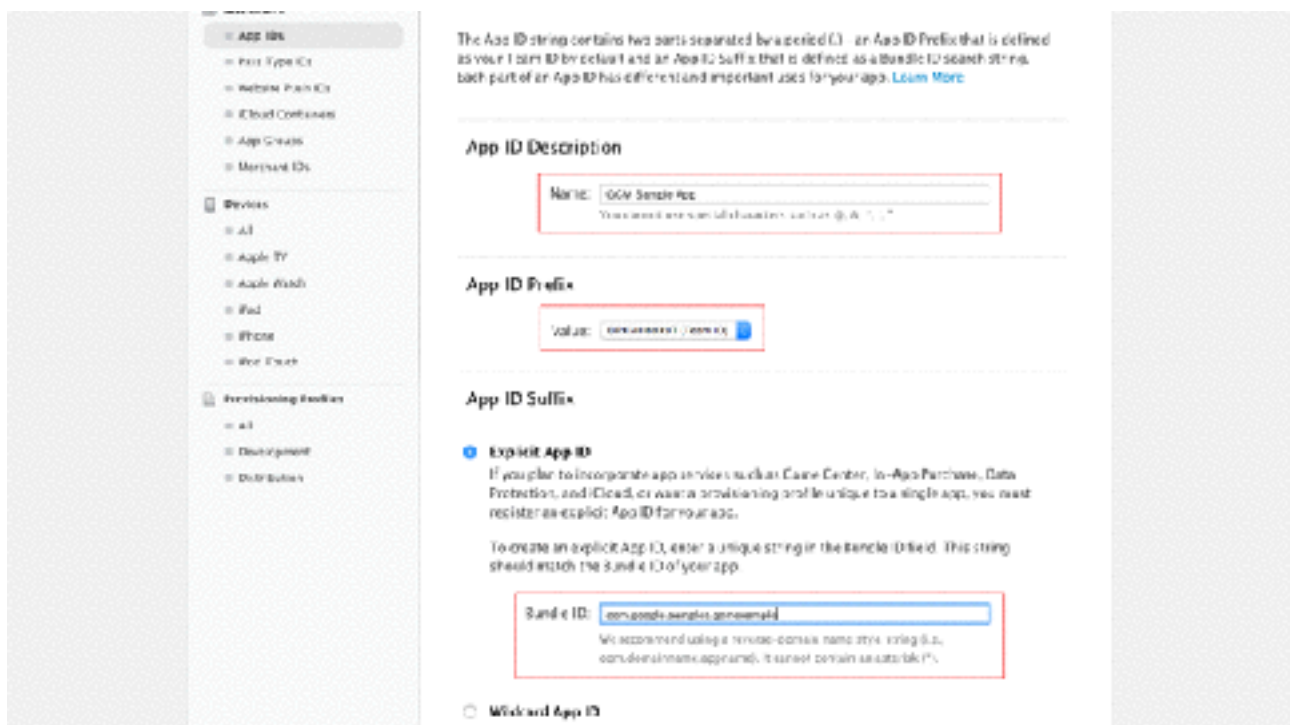


5.建立新的APP ID:

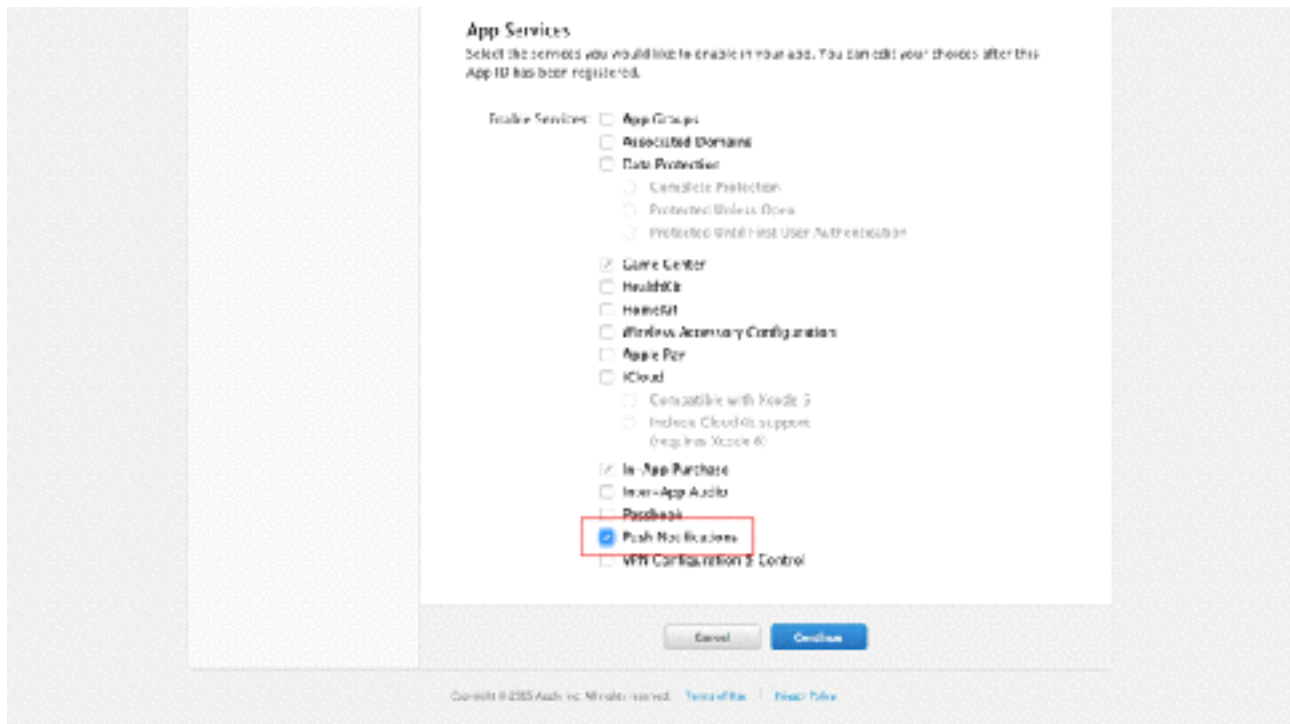
a.輸入APP ID 名字

b.選擇App ID前固字的文字(使用預設，不用更改)

c.在App ID Suffix選擇區內，請選擇Explicit App ID,然後輸入你的Bundle ID(例如:com.google.samples.firebaseioexample)。這個bundle ID的值必需符合目前Xcode專案內的bundle ID,也符合在FCM設定的bundle ID.



d.在這App Services選項，確定這個Push Notification已經被選取



6.按下繼續和檢查輸入是正確的。

a.這識別的值必需符合prefix和Bundle ID

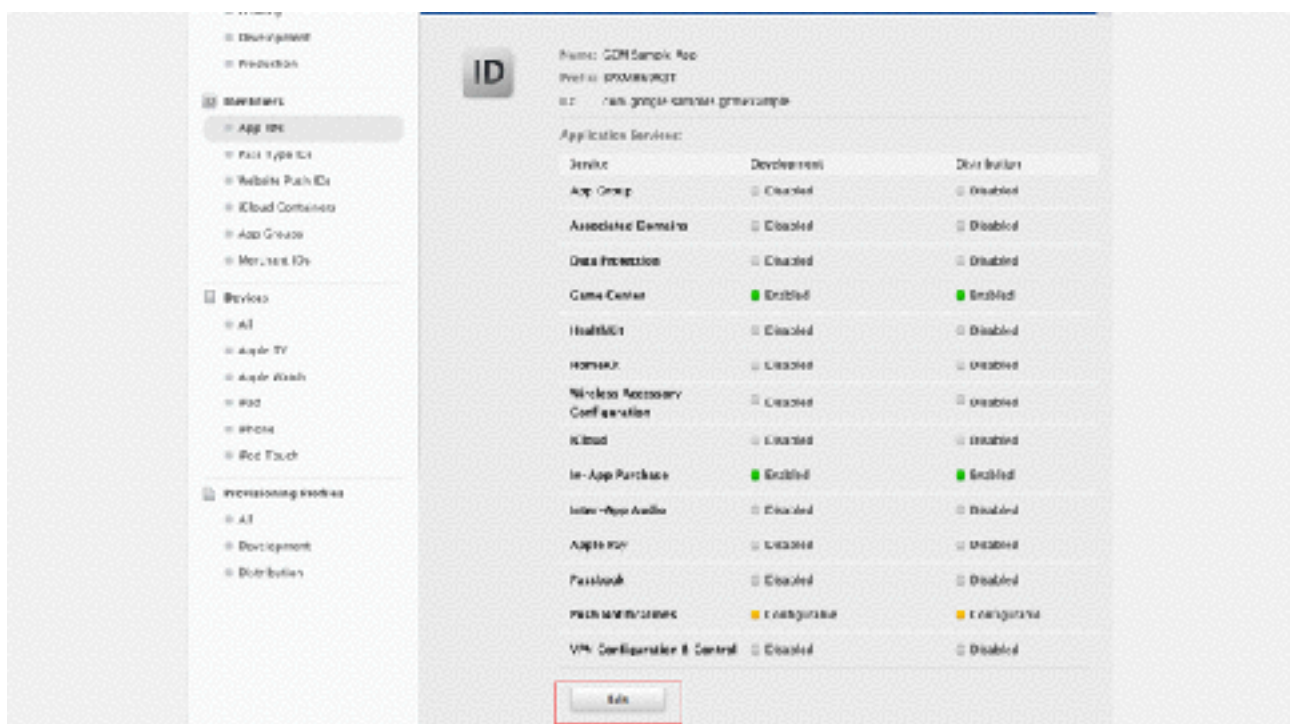
b.Push Notifications應該是Configurable

7.按下註冊來建立這個App ID。

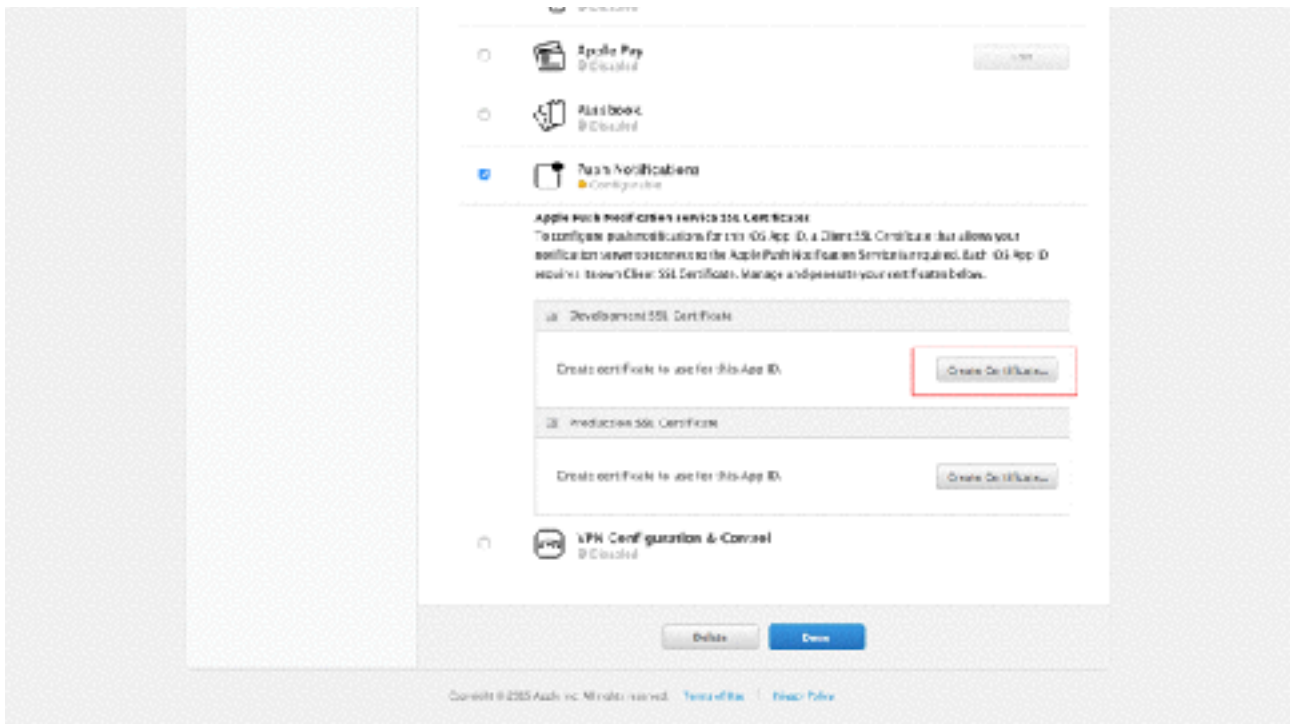
修改App ID的Push Notifications

要送Push Notifications到一個iOS APP。你必需修改App ID。

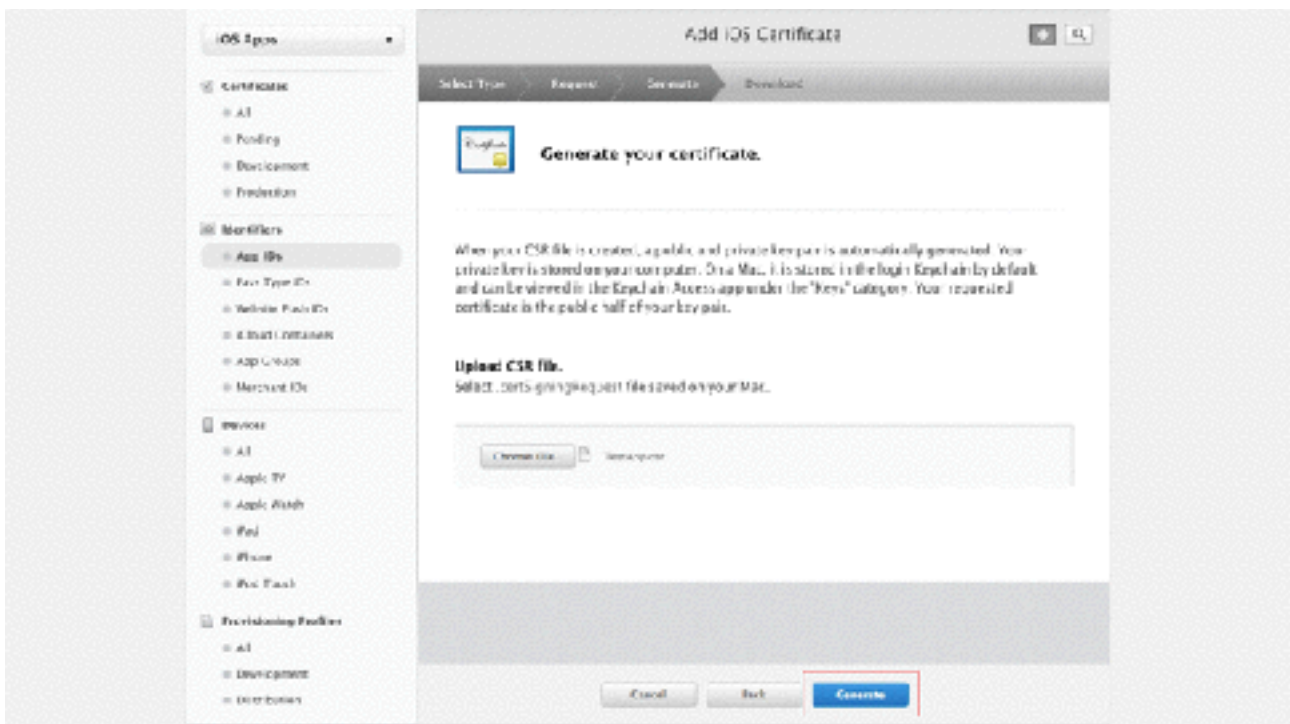
1.在App IDs內，選擇想要編輯的AppID和按下Edit按鈕



2.在Push Notifications選項內，你可以建立開發的SSL憑證或產品憑證。建立開發者憑證，在開發者SSL憑證區，按下Create Certificate

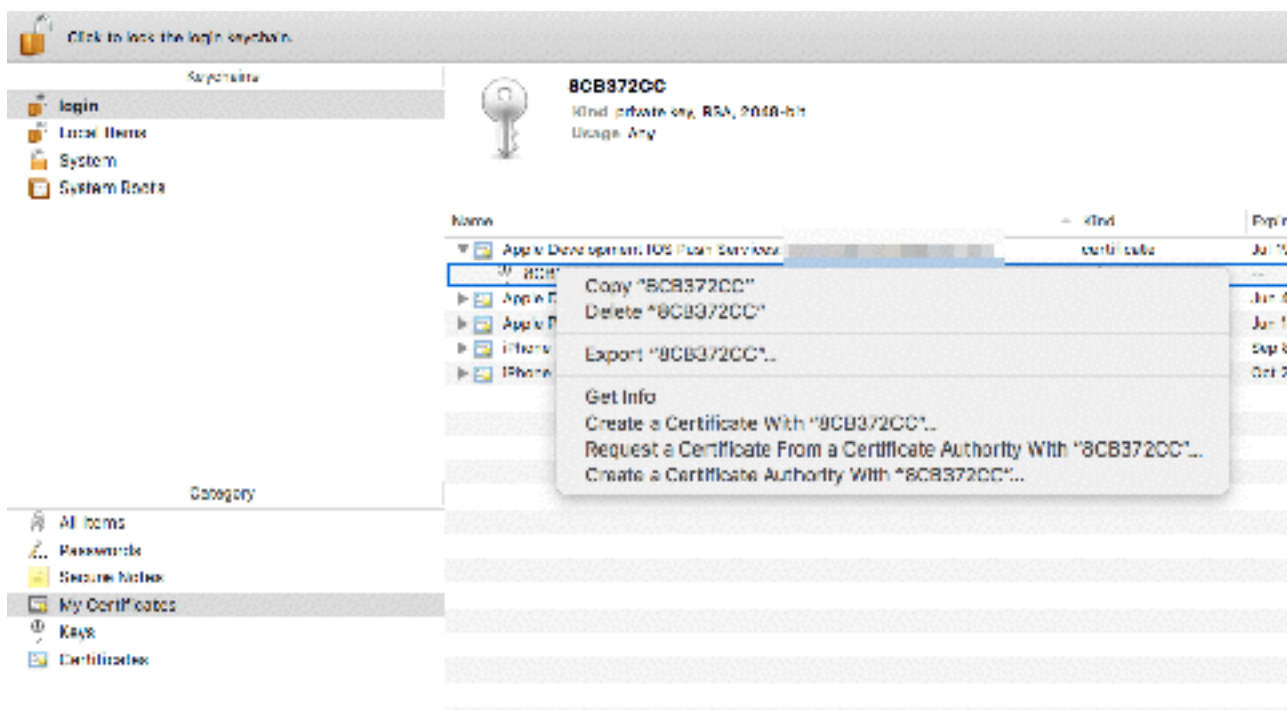


3.你現在應該看到一個選項About Creating a Certificate Signing Request(CSR)，按下繼續。
4.跟隨著步驟，將會來到Generate選項。按下Choose File 和選取我們先前建立的certSigningRequest，當可以活動時，然後按下Generate。



5.當SSL憑證已經完成，按下Download和儲存在mac上。並且按下Done完成憑證建立的步驟。

- 6.在mac上找到下載的SSL憑證和點2下以將它安裝到鑰匙圈。
- 7.在mac上，打開鑰匙圈存取，點選我的憑證，找到你剛才加入的憑證; 它應該叫Apple Development iOS Push Services:你的bundleID。
- 8.展開箭頭然後按下右鍵和輸出這私人鑰匙.p12檔。如果有需要，加入密碼來保護輸出的憑證。如果你沒有看到選項輸出你的憑證，請再確認你有先點選有邊欄位的我的憑證。



這個app現在已經可以在開發環境使用推播。當你完成你的App在上架前，你需要啟動app的推播產品環境；重覆這些步驟，但是在建立憑證時，使用Production SSL Certificate選擇區來取代Development SSL Certificate。

如果在呼叫 `setAPNSToken`，請先確認這個type的值是否正確，在sandbox環境，請使用 `FIRInstanceIDAPNSTokenTypeSandbox`，或者在產品環境，請使用 `FIRInstanceIDAPNSTokenTypeProd`。如果沒有設定正確的type，訊息將不會被傳遞到APP

建立Provisioning Profile

在開發階段，測試你的app必需使用Provisioning Profile，以便認證你的手機可以執行這個app。這個app還沒有上傳到App Store。

- 1.導覽到apple的開發者會員中心的頁面並登入。
- 2.導覽到Certificates,Identifiers and Profiles。
- 3.在頁面左上角下拉選項中，選擇iOS,tvOS,watchOS。然後導覽至Provisioning Profile>All
- 4.按下+按鈕到建立新的APP ID
- 5.選擇iOS App Development當作provisioning profile類別，然後按下繼續。
- 6.在下拉的menu，選擇你想要的App ID,然後按下繼續。
- 7.選擇先前設定完成的IOS開發憑證。然後按下繼續。

- 8.選擇你想要app執行的手機，這手機編號將會加入到目前的provisioning Profile，然後按下繼續。確定選擇所有你想要測試的手機。
- 9.輸入provisioning profile的名字,(例如:Firebase Sample App Development Profile)，然後按下產生。
- 10.按下下載和儲存這Provisioning profile到你的mac。
- 11.快點2下provisioning profile. 以便安裝它。

傳遞第一個訊息到背景模式的App

在開始使用FCM，建立一個最簡單的案例就是使用Notifications composer送出一個訊息。並且在Notifications composer內指定一個使用者手機，這個手機已經安裝了app，並且是在背景的模式下。這章節將從設定到驗證一步一步解說，以達到訊息傳遞的效果。如果你已經有在FCM內設定了一個使用者端的app，或許有些步驟你已經完成。

增加Firebase 到IOS專案

如果你的app已經有其它Firebase 的功能，這個章節所含蓋的任務你可能已經完成。為了FCM雲端訊息，你將需要上傳你的APNs憑證和註冊遠端訊息。

事前準備

在開始前，你必需現設定一些環境：

Xcode 7.0以上

Xcode專案，目標是IO7以上

app bundle名稱

CocoaPods1.0.0以上

為了雲端訊息：

- 一個實體手機
- 一個APNs憑證和推播通知打開
- 在Xcode內的APP->Capabilities內打開Push Noifications

如果您沒有XCode專案，而且只是想要測試，你可以下載quickstart sample。如果是使用一個squickstart，記得在專案設定中取行bundle識別名稱，在下個章節將會使用到。



註解：如果你是使用Firebase SDK 2.0更新，請參考 [upgrade gide for ios](#)到開始。

增加一個Firebase到App

現在增加Firebase到你的app，實作這個動作時，將需要在專案內加入Firebase的專案和Firebase的設定環境檔。

6. 如果還沒有Firebase 專案,則在Firebase控制台建立Firebase 專案。如果已經有Google專案連結到手機的app，請使用import Google Project。不然請使用Create New Project。
7. 使用Add Firebase to your IOS app和跟隨它的步驟。如果輸入的是一個已經出現過的Google專案，這些步驟將會自動執行。

如要下載 iOS 應用程式的設定檔，請按照以下步驟進行：

1. 登入 Firebase，然後開啟專案。
2. 點選 ，然後選取 [專案設定]。
3. 在「您的應用程式」卡片中，從清單中選取您要下載設定檔的應用程式的繫結 ID。
4. 點選 [ GoogleService-Info.plist]。
3. 當出現輸入你的app's bundle ID時，這個動作非常重要。專案App內的bundle ID 必需要Firebase console專案的bundle ID是一樣的。
4. 最後必需下載GoogleService-Info.plist檔。
5. 全部動作完成後，將GoogleService-Info.plist加入到Xcode專案的根目錄內

加入SDK

如果設定新的專案，必需要安裝SDK。或許在建立Firebase project時，已經完成安裝。我們建議使用CoCoadPods來安裝函式庫。可以藉由此安裝方式來安裝CoCoadPods。

<https://cocoapods.org/>

如果您準備下載和執行快速開始範例的其中一個範例,這Xcode專案常未全部完成，必需要安裝pods和下載googleService-info.plist檔。如果要整合Firebase library進入到我們的專案內，我們必需將所需使用的Firebase library加入到我們的專案內。

- 1.如果沒有Xcode專案，請建立一個。
- 2.如果沒有PodFile，請建立一個。

```
$ cd your-project directory
$ pod init
```

3.增加想要加入的Pods。設定在Podfile內，設定如下：

```
pod 'Firebase/Core'
pod 'Firebase/Messaging'
```

這將加入事前必需準備好的Firebase libraries和在ios app內執行Firebase Analytics。下面有目前可以加入的pods的列表，並且有對應到特定的功能。

4.安裝pods和使用Xcode打開.xcworkspace file

```
$ pod install
$ open your-project.xcworkspace
```

5.從Firebase console下載 GoogleService-Info.plist，並且將它加入到App內。

上傳您的APNs憑證

上傳您的APNs憑證到Firebase。如果你還沒有一個APNs憑證，請先建立。

- 1.進入Firebase控制台的專案內。選擇工具的按鈕，選擇Project Settings，然後選擇這雲端訊息的標籤。
 - 2.在開發環境區選擇Upload Certificate按鈕，或者是產品環境區，2個至少選擇一個。
 - 3.選擇這個.p12檔和提供密碼。確定這bundleID是使用正確的憑證bundleID。
-

在App內初始化Firebase

你將需要加入Firebase初始化的程式碼進入到應用程式。輸入Firebase 模組和調整一個實體如下。

- 1.在UIApplicationDelegate內輸入Firebase

```
import Firebase
```

- 2.通常在應用程式的application:didFinishLaunchingWithOptions:方法內，調整FIRApp分享實體。

```
FirebaseApp.configure()
```

註冊遠端通知

在應用和式的起點或想要的時間點，在app內註冊遠端通知。呼叫registerForRemoteNotifications:

```

if #available(iOS 10.0, *) {
    // For iOS 10 display notification (sent via APNS)
    UNUserNotificationCenter.current().delegate = self

    let authOptions: UNAuthorizationOptions = [.alert, .badge, .sound]

    UNUserNotificationCenter.current().requestAuthorization(
        options: authOptions,
        completionHandler: {_, _ in })

    // For iOS 10 data message (sent via FCM)
    FIRMessaging.messaging().remoteMessageDelegate = self
} else {
    let settings: UIUserNotificationSettings =
        UIUserNotificationSettings(types: [.alert, .badge, .sound], categories: nil)
    application.registerUserNotificationSettings(settings)
}

application.registerForRemoteNotifications()
AppDelegate.swift

```

存取註冊的Token

在預設的情況下，App在開始執行時，藉由FCM SDK產生一個註冊的token。如果你的目標是一個單一的手機或一群手機，需要依靠存取這個token。

這個章節將描述如何取出token和如何管理token的改變。因為在啟動後，token有可能被改變，所以強烈建議取出最新的註冊token，除非你有特定需求的使用。

下面是token會改變的情況：

- 這app刪除了實體ID
- 這app被重新存在新的手機
- 這使用者移除和重新安裝app
- 這使用者清除app 資料

取出現在註冊的token

當你需要取出現在的token,呼叫FIRInstanceID.instanceID().token()!, 如果這個token尚未被產生, 則傳出null。

```
let token = FIRInstanceID.instanceID().token()!
```

管理token的產生

可以藉由NSNotificationCenter增加觀察者監聽kFIRInstanceIDTokenRefreshNotification, 然後在註冊的function內取出更新的token。假設tokenRefreshNotification是註冊的function:

```
func tokenRefreshNotification(_ notification:
Notification) {
    if let refreshedToken =
FIRInstanceID.instanceID().token() {
        print("InstanceID token: \(refreshedToken)")
    }

    // Connect to FCM since connection may have failed
when attempted before having a token.
    connectToFcm()
}
```

[AppDelegate.swift](#)

在IOS10以上, 在app完成啟動前, 你必需指定你的代理人物件給UNUserNotificationCenter物件, 以便於接收顯示的通知。也同時給FIRMessaging物件, 以便於接收資料。例如在ios的app這些動作必需要在 applicationWillFinishLaunching:和applicationDidFinishLaunching:內完成。

當token被產生, kFIRInstanceIDTokenRefreshNotification 會發生, 所以在現有的環境呼叫FIRInstanceID.instanceID().token()!, 取得token, 確保是可以使用的token。

送出一個通知訊息

- 1.在手機上安裝和執行app。你必需接受允許接收遠端訊息的請求。
- 2.確定你的app是在背景模式。
- 3.在Firebase 控制台，打開Notifications標籤和選擇新訊息。
- 4.填寫訊息文字。
- 5.訊息目標請選擇Single Device。
- 6在FCM Registration Token欄位，加入在上個步驟取得的註冊token。

按下送出後，目標手機的app在背景模式，可以在通知中心接收到通知的訊息。

送一個訊息給多個手機

Firebase 雲端訊息提供2種方式，傳遞給多個手機：

- 主題訊息，允許你傳送訊息給多個手機，這些手機已經訂閱了一個相同的特定的主題。
- 手機群組訊息，允許你傳送單一訊息給多個同一群組的app實體。

這個說明主要著重於在從app 伺服器內使用HTTP或者是XMPP協定送出主題訊息，和在ios app內，如果接收和處理它們。這章將會解說所有的流程，從設定到驗證。所以它含蓋一些你先前可能已經完成的步驟。

增加Firebase 到IOS專案

如果你的app已經有其它Firebase 的功能，這個章節所含蓋的任務你可能已經完成。為了FCM雲端訊息，你將需要上傳你的APNs憑證和註冊遠端訊息。

事前準備

在開始前，你必需現設定一些環境：

Xcode 7.0以上

Xcode專案，目標是IO7以上

app bundle名稱

CocoaPods1.0.0以上

為了雲端訊息：

- 一個實體手機
- 一個APNs憑證和推播通知打開
- 在Xcode內的APP->Capabilities內打開Push Noifications



如果您沒有XCode專案，而且只是想要測試，你可以下載quickstart sample。如果是使用一個squickstart，記得在專案設定中取行bundle識別名稱，在下個章節將會使用到。

註解：如果你是使用Firebase SDK 2.0更新，請參考 [upgrade gide for ios](#)到開始。

增加一個Firebase到App

現在增加Firebase到你的app，實作這個動作時，將需要在專案內加入Firebase的專案和Firebase的設定環境檔。

6. 如果還沒有Firebase 專案,則在Firebase控制台下建立Firebase 專案。如果已經有Google專案連結到手機的app，請使用import Google Project。不然請使用Create New Project。

7. 使用Add Firebase to your iOS app和跟隨它的步驟。如果輸入的是一個已經出現過的Google專案，這些步驟將會自動執行。
如要下載 iOS 應用程式的設定檔，請按照以下步驟進行：
 1. 登入 Firebase，然後開啟專案。
 2. 點選 ，然後選取 [專案設定]。
 3. 在「您的應用程式」卡片中，從清單中選取您要下載設定檔的應用程式的繫結 ID。
 4. 點選 [ GoogleService-Info.plist]。
3. 當出現輸入你的app's bundle ID時，這個動作非常重要。專案App內的bundle ID必需要Firebase console專案的bundle ID是一樣的。
4. 最後必需下載GoogleService-Info.plist檔。
5. 全部動作完成後，將GoogleService-Info.plist加入到Xcode專案的根目錄內

加入SDK

如果設定新的專案，必需要安裝SDK。或許在建立Firebase project時，已經完成安裝。我們建議使用CoCoadPods來安裝函式庫。可以藉由此安裝方式來安裝CoCoadPods.

<https://cocoapods.org/>

如果您準備下載和執行快速開始範例的其中一個範例,這Xcode專案常未全部完成，必需要安裝pods和下載googleService-info.plist檔。如果要整合Firebase library進入到我們的專案內，我們必需將所需使用的Firebase library加入到我們的專案內。

- 1.如果沒有Xcode專案，請建立一個。
- 2.如果沒有PodFile，請建立一個。

```
$ cd your-project directory
$ pod init
```

- 3.增加想要加入的Pods。設定在Podfile內，設定如下：

```
pod 'Firebase/Core'
pod 'Firebase/Messaging'
```

這將加入事前必需準備好的Firebase libraries和在ios app內執行Firebase Analytics。下面有目前可以加入的pods的列表，並且有對應到特定的功能。

- 4.安裝pods和使用Xcode打開.xcworkspace file

```
$ pod install
$ open your-project.xcworkspace
```

5.從Firebase console下載 GoogleService-Info.plist，並且將它加入到App內。

上傳您的APNs憑證

上傳您的APNs憑證到Firebase。如果你還沒有一個APNs憑證，請先建立。

- 1.進入Firebase控制台的專案內。選擇工具的按鈕，選擇Project Settings，然後選擇這雲端訊息的標籤。
 - 2.在開發環境區選擇Upload Certificate按鈕，或者是產品環境區，2個至少選擇一個。
 - 3.選擇這個.p12檔和提供密碼。確定這bundleID是使用正確的憑證bundleID。
-

在App內初始化Firebase

你將需要加入Firebase初始化的程式碼進入到應用程式。輸入Firebase 模組和調整一個實體如下。

- 1.在UIApplicationDelegate內輸入Firebase

```
import Firebase
```

- 2.通常在應用程式的application:didFinishLaunchingWithOptions:方法內，調整FIRApp分享實體。

```
FirebaseApp.configure()
```

註冊遠端通知

在應用和式的起點或想要的時間點，在app內註冊遠端通知。呼叫registerForRemoteNotifications:

```

if #available(iOS 10.0, *) {
    // For iOS 10 display notification (sent via APNS)
    UNUserNotificationCenter.current().delegate = self

    let authOptions: UNAuthorizationOptions = [.alert, .badge, .sound]

    UNUserNotificationCenter.current().requestAuthorization(
        options: authOptions,
        completionHandler: {_, _ in })

    // For iOS 10 data message (sent via FCM)
    FIRMessaging.messaging().remoteMessageDelegate = self

} else {
    let settings: UIUserNotificationSettings =
        UIUserNotificationSettings(types: [.alert, .badge, .sound], categories: nil)
    application.registerUserNotificationSettings(settings)
}

application.registerForRemoteNotifications()
AppDelegate.swift

```

app訂閱主題

使用者的apps可以訂閱已經出現的topic,或者可以建立一個新的topic。當一個使用者訂閱一個新的主題名稱(在Firebase 專案內沒有出現的名稱)，FMC會依據這個名字，建立新的主題，任何其它的使用者app之後可以馬上訂閱。

這FIRMessaging 類別有專門處理專題訊息的功能。要訂閱一個主題，使用主執行緒呼叫 `subscribeToTopic:topic` (FCM不支援執行緒安全)。

```

[[FIRMessaging messaging] subscribeToTopic:@" /topics/news"];
NSLog(@"Subscribed to news topic");

```

這個製造了一個非同步的需求到這FCM端和這使用者訂閱了個個topic.如果訂閱需求失敗，FCM會一直重覆執行直到你訂閱成功。每一次這app開始，FCM確認所有的需要的主題已經被訂閱。

解除訂閱，呼叫unsubscribeFromTopic:topic,和FCM在背景解除訂閱。