

# Firebase Analytics For IOS

Firebase Analytics 免費提供App開發者，了解使用者的行為和App使用的狀態。

Firebase Analytics是Firebase的核心，Firebase Analytics是免費的，沒有限制分析報告的解決方案。Analytics整合所有Firebase的特性和提供您最多500個不同事件並且沒有限制的分析報告。這些事件可以使用Firebase SDK來定義。Firebase Analytics報告幫助App的開發者了解使用者的行為，這些行為讓開發者都可以得到更好的分析資訊，有助於app的行銷和效能的更佳化。

---

## 主要的功能

沒有限制的分析報告	提供最多500個不同的事件，但不限分析報告數量
區隔使用者族群	可以在Firebase控制台上依據手機資料，自訂事件或使用使用者屬性來自定使用者族群，這些族群也可以被使用在Firebase的其它功能。

---

## 如何工作？

Firebase Analytics幫助你了解人們如何使用您的ios App。這SDK自動截取一些事件和使用者的屬性，它也允許您依據您商業模式的需求，建立一個獨一無二的自訂事件。一但這資料被截取，它馬上可以顯示在Firebase控制台的統計顯示表。這個統計顯示表提供更多內部細項資料，例如有效活動的使用者和使用者的特徵到app內最受歡迎的項目。

Firebase Analytics也整合其它的Firebase功能。例如，它自動記錄Firebase Notification登入的事件，提供每位收到推播後的行為。的報告。透過活動的效能了解那一個方法可以讓您最有效得到高價值的客戶。如果您需要自訂的分析或從別的來源來加入更多的資料或是多重的來源資料，這是允許的，我們將有更複雜的分析。

## 整合其它的服務

<b>BigQuery</b>	連結Firebase Analytics app到巨量需求。前題是可以在巨量需求中自訂分析資料和輸入其它資料來源。
<b>Firebase Crash Reporting</b>	Firebase Analytics 會記錄app的每次閃退，開發者可以得到在不同版本和語言區域內，各自閃退的比例。也可以建立遇到閃退的使用者群組。並且使用Firebase notification直接傳送回應資訊給使用者。
<b>Firebase Notifications</b>	Firebase Analytics 自動記錄log事件，事件如果符合Firebase Notification的條件。將執行Firebase Notification動作，也支援每個使用都的活動報告。
<b>Firebase Remote Config</b>	使用Firebase Analytics 定義的使用者群組。讓app可以因不同群組改變不同的行為和外觀。開發者不用為App建立多種版本。
<b>Google Tag Manager</b>	整合Google Tag Manager，可以讓網頁介面來遠端管理Firebase Analytics實作

## 實作流程

<b>1 連結App到Firebase</b>	3 使用IOS的Firebase Analytics是容易的。只要加入Firebase的SDK到新的或現有的app, 然後將自動開始資料的收集。你可以每小時使用Firebase console觀察分析資料。
<b>2.記錄自訂資料</b>	你可以分析自己自訂的有用事件。像商務app的訂購事件或完成事件。
<b>3.建立使用者群組</b>	可以在Firebase console內定義使用者群組
<b>4.目標群組</b>	可以自訂目標使用者群組，傳送訊息，推銷商品或使用其它Firebase的功能，例如推播和遠端調整。

# 開始使用ios    Firebase Analytics

firebase Analytics 收集方法和行為資料。這SDK Log記錄2個主要的資訊類型。

<b>Events</b>	app發生什麼事情，例如使用都的動作，系統事件或錯誤。
<b>User Properties</b>	用來描素區隔使用者的屬性，例如語系或地理位置。

Analytics 自動記錄一些事件和使用者的屬性。不需要透過任何程式碼來打開這功能。

---

## 事前準備

- 1.增加Firebase到我們的專案。
- 2.(建議使用)增加AdSupport framework到我們的專案，讓專案可以增加功能，例如定義使用者群組和定義活動屬性。

---

## 增加AdSupport framework至專案

一些Analytics 功能，例如使用者群組和活動屬性和一些使用者屬性，例如年齡和興趣。需要增加AdSupport 框架才可以使用這些功能。沒有這個框架，Analytics不能收集這些資訊。

加入AdSupport框架：

- 1.在xcode專案內,選擇專案target.
- 2.選擇General標籤。
- 3 展開這Linked Framework and Libraries區域。
- 4 .按下+到增加一個Framework.
- 5 .選擇AdSupport.framework.

在上架app前，請確認app有使用IDFA usage guidelines編譯

---

## 加入Analytics 到您的app

- 1.加入這Firebase附屬的功能到PodFile:

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

- 2.執行pod install和打開這被建立的檔案.xcworkspace檔案。
- 3.輸入這Firebase 模組進入專案的UIApplicationDelegate 子類別內。

```
import Firebase
```

- 4 .調整FIRApp這個實體，傳統上會在應用程式的application:didFinishLaunchingWithOptions:方法內加上：

```
FIRApp.configure()
```



# Log events

事件提供開發者洞察app發生何事？例如使用者活動，系統事件，或錯誤。

Analytics自動記錄一些事件，你不需要寫額外的程式碼接收這些事件。如果你的app需要收集額外的資料，你可以記錄最多500個不同的分析事件類型。沒有限制這些類型所產生的事件數量。

---

## 開始之前

如果這是第一次增加Analytics 到app，完成下面步驟：

連結您的app 到Firebase console

- 1.安裝Firebase SDK。
- 2.在Firebase console內，增加您的app到您的Firebase 專案。

---

## 加入Analytics 到您的app

- 1.加入這Firebase附屬的功能到PodFile:

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

- 2.執行pod install和打開這被建立的檔案.xcworkspace檔案。
- 3.輸入這Firebase 模組進入專案的UIApplicationDelegate 子類別內。

```
import Firebase
```

- 4.調整FIRApp這個實體，傳統上會在應用程式的application:didFinishLaunchingWithOptions:方法內加上：

```
FIRApp.configure()
```

---

## Log events

實體FIRApp初始化並調整後，可以開使用logEventWithName()方法記錄事件。

為了加快使用Analytics的速度，AnalyticsSDK已經幫我們事前先定義不少的事件。包括零售，電子商務，旅遊和遊戲類型的app事件。學習更多有關於這些事件使用的方法，請參考Firebase Help Center內的 Events and propertys文章。

註解:要得到最詳細的報告，記錄事前建議的事件和使用它們的參數，將是最好的選擇。

你可以在下方位置找到一些實作的細節：

建議事件：請看FIREventNames.h標頭檔。

規定參數:請看FIRParameterName.h標頭檔。

下面的範例示範如何記錄建議的kFIRSelectContent事件:

```
FIRAnalytics.logEvent(withName: kFIREventSelectContent,
parameters: [
    kFIRParameterItemID: "id-\(title!)" as NSObject,
    kFIRParameterItemName: title! as NSObject,
    kFIRParameterContentType: "cont" as NSObject
])
```

另外這規定參數，你可以增加這下面的參數到任何的事件:

- 自訂的規定參數：雖然這些自訂參數不能被直接展示在Analytics report上，它們可以被使用在定義客戶群組中的過濾項目中。如果app有連結到BigQuery專案，自訂的參數也可以將資料輸出給BigQuery。
- kFIRParameterValue 參數:kFIRParameterValue 是一般目地的參數，它方便使用在一些的單位，例如收入,距離和點。

如果你的應用程式有特定的需求，而建議的事件類型並沒有包括，則可以自訂事件，展示在下面的範例:

```
FIRAnalytics.logEvent(withName: "share_image",
parameters: [
    "name": name as NSObject,
    "full_text": text as NSObject
])
```

---

## 在XCode debug console內觀察事件

你可以藉由SDK觀察記錄的事件，並驗證這些事件在適當時間被記錄。它包含自動和手動2種事件。依照下列步驟打開觀察記錄功能

1. 在Xcode內，選擇Product>Scheme>Edit scheme...
2. 在左邊的項目選擇Run
3. 選擇Arguments標籤
4. 在Arguments Passed on Launch區域，加上 `-FIRAnalyticsDebugEnabled`。

下一次再執行app，事件將會顯示在Xcode debug 控制台上, 幫助您送出事件時，馬上立即驗證是否成功。

---

## 在Firebase console 的觀察事件統計表

你可以在Firebase console儀表板上，觀察所有事件總數統計的值。這儀表板每天週期性的更新。要立即的測試，使用`debug` 控制台來觀察。參考上節。

你可以在*Firebase console* 處理資料，步驟如下：

- 1.在*Firebase console*打開你的專案。
- 2.在項目上選擇*Analytics*在儀表板來觀察分析報告。

這事件的標籤展示了在*app*內記錄所有不同類型的事件報告。了解更多的分析報告儀表板，請至 *Firebase Help Center* 讀 *Analytics reporting dashboard*文章。

# 設定使用者屬性

使用者的屬性是由您自己定義一些特定的群組。例如使用者的語系或使用者的位置。Analytice 自動記錄一些使用者屬性。開發者不需要加任何程式碼就可以使用它。如果您的app需要收集額外的資料，開發者可以在app內設定取多25組的不同屬性。

註解:年齡，性別和興趣的屬性是自動被收集的，但必需要你的app有連結到Ad Support 框架。連結到這框架也自動收集廣告的識別(IDFA)

---

## 開始之前

如果這是第一次增加Analytics 到app，完成下面步驟：

連結您的app 到Firebase console

- 1.安裝Firebase SDK。
- 2.在Firebase console內，增加您的app到您的Firebase 專案。

---

## 加入Analytics 到您的app

- 1.加入這Firebase附屬的功能到PodFile:

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

- 2.執行pod install和打開這被建立的檔案.xcworkspace檔案。
- 3.輸入這Firebase 模組進入專案的UIApplicationDelegate 子類別內。

```
import Firebase
```

- 4.調整FIRApp這個實體，傳統上會在應用程式的application:didFinishLaunchingWithOptions:方法內加上：

```
FIRApp.configure()
```

---

## 設定使用者屬性

你可以設定Analytic使用者屬性來描述app的使用者。您可透過Firebase console 報告來過濾資料。觀察有多少使用者選擇過濾的item。

設定使用者屬性的步驟：

- 1.在Firebase Console的analytic頁面註冊您想要的屬性。
- 2.加入程式碼setUserPropertyString()方法來設定analytic使用者屬性。你可以使用這屬性名和每個使用者選擇的值。



這下面的範例展示一個活動的使用者，使用最愛食物屬性，給予一個選擇的food定串值。

```
FIRAnalytics.setUserPropertyString(food, forName:
"favorite_food")
```

註解:一但這屬性被註冊,它會花費數個小時來收集資料後，才會顯示在報告內。當這新資料是可以使用。這使用者的屬性可以在過濾器內使用。

使用下列步驟存取下面資料：

- 1.在Firbase console內，打開專案。
- 2.選擇Analytic選單來觀察分析報告表。

這User Properties標籤展示我們在app內所自訂的所有使用者屬性。在分析報告中你可以使用這些設定的屬性當作過濾的條件。在Firebase Help Center中的Analytics reporting dashboard內可以取得更多的資料。

註解:在Analytic的報告內的資料，每天重新整理一次。

# 利用IOS 的WebView使用Analytics

使用WebView呼叫記錄事件和設定使用者屬性，在傳送給Firebase Analytics前，必需先傳遞給原生程式。

## 實作javascript 處理者

在webView內使用Firebase Analytics第一步是建立javascript function來傳遞事件和使用者屬性給原生程式。這下列範例展示如何傳遞事件和使用者屬性，程式相容於android 和 ios。

```
function logEvent(name, params) {
  if (!name) {
    return;
  }

  if (window.AnalyticsWebInterface) {
    // Call Android interface
    window.AnalyticsWebInterface.logEvent(name,
JSON.stringify(params));
  } else if (window.webkit
    && window.webkit.messageHandlers
    && window.webkit.messageHandlers.firebase) {
    // Call iOS interface
    var message = {
      command: 'logEvent',
      name: name,
      parameters: params
    };
    window.webkit.messageHandlers.firebase.postMessage(message);
  } else {
    // No Android or iOS interface found
    console.log("No native APIs found.");
  }
}
```

```
function setUserProperty(name, value) {
  if (!name || !value) {
    return;
  }

  if (window.AnalyticsWebInterface) {
    // Call Android interface
    window.AnalyticsWebInterface.setUserProperty(name,
value);
  } else if (window.webkit
    && window.webkit.messageHandlers
    && window.webkit.messageHandlers.firebase) {
    // Call iOS interface
    var message = {
      command: 'setUserProperty',
      name: name,
      value: value
    };

    window.webkit.messageHandlers.firebase.postMessage(messa
ge);
  } else {
    // No Android or iOS interface found
    console.log("No native APIs found.");
  }
}
```

---

---

## 實作原生介面

使用javascript呼叫ios程式碼，實作一個接收訊息處理的method，這method 採納 WKScriptMessageHandler protocol。實作userContentController:didReceiveScriptMessage:

```
func userContentController(_ userContentController: WKUserContentController, didReceive message: WKScriptMessage) {  
    guard let body = message.body as? [String: Any] else  
    { return }  
    guard let command = body["command"] as? String else  
    { return }  
    guard let name = body["name"] as? String else  
    { return }  
  
    if command == "setUserProperty" {  
        guard let value = body["value"] as? String else  
        { return }  
        FIRAnalytics.setUserPropertyString(value, forName: name)  
    } else if command == "logEvent" {  
        guard let params = body["parameters"] as? [String: NSObject] else { return }  
        FIRAnalytics.logEvent(withName: name, parameters: params)  
    }  
}
```

[ViewController.swift](#)

最後，加入這個訊息處理method到webView的使用內容控制：

```
self.webView.configuration.userContentController.add(self, name: "firebase")
```

[ViewController.swift](#)

