

# In-App Purchase Tutorial

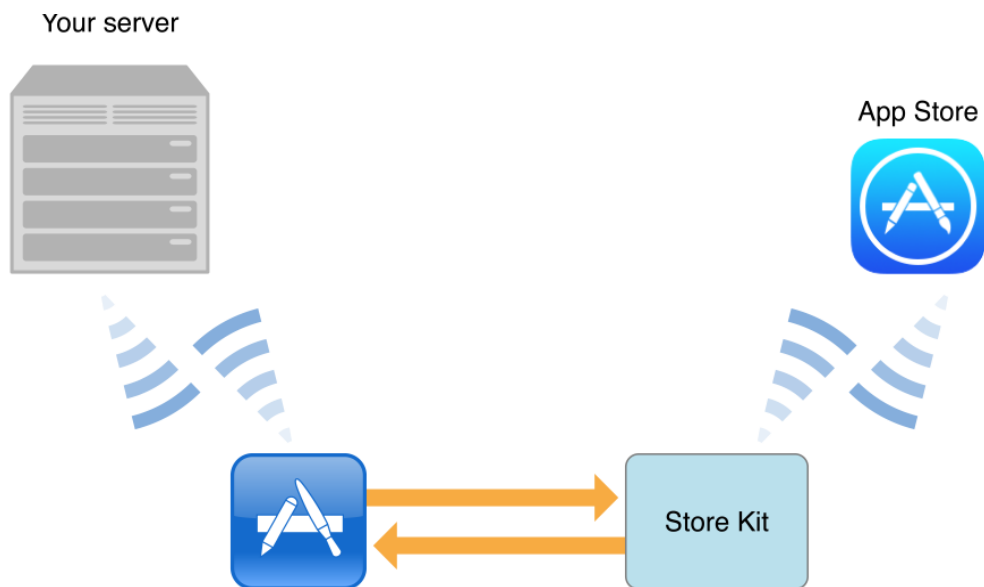
Author by Glee

# Table

IAP 機制	1
了解購買項目	2
消耗品與非消耗品的差異	2
商品限制	3
完成 Paid Contract 合約書	4
基礎稅務問題與資訊說明	5
銀行資訊	7
在 iTunes Connect 上建立商品	8
專案設定	9
宣告商品 ID	10
查詢商品供應狀況	10
設計自己的商城	11
建立購買測試帳號	12
購買商品的前置準備	12
購買商品	13
結帳完畢	13
保存收據	14
更新收據	14
向 Apple 驗證收據	15
收據內容的範例	16
還原之前的購買項目	17
Apple 參考資料	17

# IAP 機制

你的 App 可以使用 StoreKit 向 App Store 溝通，而它會將使用者的購買結果回傳給你，再由你自己負責，提供商品的服務內容，或與自己的伺服器通訊。

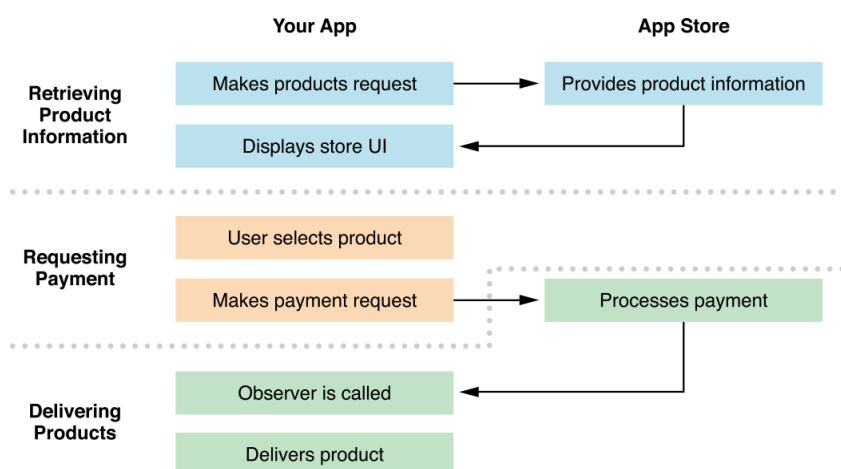


StoreKit 的流程主要有三階段。

第一階段是查詢商詢商品資訊，並將結果顯示在自己的商城。

第二階段是建立訂單，並開始付款。

第三階段是提供商品，會透過 Observer 來觸發特定 Protocol 方法。



## 了解購買項目

Apple 的購買項目需於 iTunes Connect 上建立。所以，你一定要有一組 iOS Developer 的帳號，並且登入 iTunes Connect 開始建立商品。（Apple 將商品稱呼為購買項目）

另一方面，購買項目又分為五種類別，分別表列如下：

類別	需提供回復購買	回復機制
消耗性項目 Consumable	不需要	需追蹤使用者消費記錄，自行提供消費者的回復作業。
非消耗性項目 Non-Consumable	需要	StoreKit 中有提供 <code>restoreCompletedTransactions</code> 方法
非續訂型訂閱 Non-Renewing Subscriptions	需要	利用 iCloud 或是自行架設的伺服器追蹤使用者購買記錄，並予許使用回復紀錄到多部裝置。
自動續訂型訂閱 Auto-Renewing Subscriptions	需要	StoreKit 中有提供 <code>restoreCompletedTransactions</code> 方法
免費訂閱 Free Subscriptions	需要	StoreKit 中有提供 <code>restoreCompletedTransactions</code> 方法

除了消耗品外，其它所有的商品都必需要能夠讓使用者在重新安裝 App 後，都能回復原本的購買，其各別的回復方式可參考回復機制。

## 消耗品與非消耗品的差異

像是遊戲中的金幣，或是補血藥水等等，這種用過就消失的商品，都可以用消耗品建立。反之，像是提供服務的內容，諸如購買後就能移除廣告的功能，或是升級為全部功能版等，就屬於非消耗品的一種。

## 商品限制

In-App Purchase（簡稱 IAP）並不像 Paypal、微信支付等第三方付費機制，雖然可以處理金流的細節，但同樣會被 Apple 收取 30% 的服務費，另外也有一些限制需先了解。

範例	予許販售	商品類型 / 原因
購買電子書	可以	非消耗性項目
購買點數進行 VoIP 通話	可以	消耗性項目
購買遊戲道具	可以	消耗性項目 / 非消耗性項目
購買實體書	不行	實體商品不得販售
購買商品在多個 App 使用	不行	在哪個 App 販售的商品僅能用於該 App
購買遊戲關卡	可以	非消耗性項目
開通存檔功能	可以	非消耗性項目
六個月的時間存取特定功能	可以	自動續訂型訂閱
水電維修服務	不行	真實世界的服務不得交易
網頁外包設計	不行	真實世界的服務不得交易
每個月的健身房會籍	不行	真實世界的服務不得交易

# 完成 Paid Contract 合約書

開始建立購買項目前，請先確定你已經 W-8BEN 的表格，才可以建立付費的購買項目。否則只可以建立免費訂閱商品。

iTunes Connect

協議、稅務與銀行業務

Hsueh Hsin Tsai

Hsueh Hsin Tsai

協議、稅務與銀行業務

Master Agreements

Contracts In Process

Once you complete setup and the effective date has been reached, the contract will be moved to the Contracts In Effect section.

Contract Region	Contract Type	Contract Number	Contact Info	Bank Info	Tax Info	Download	Status
World	All App Revenue	XXXXXXXXXX	<a href="#">Edit</a>	<a href="#">Edit</a>	<a href="#">Edit</a>	<a href="#">Download Agreement</a>	Pending Tax, Bank, Contract

Contracts In Effect

Contract Region	Contract Type	Contract Number	Contact Info	Bank Info	Tax Info	Effective Date	Expiration	Download
All (See Contract)	Paid Applications	XXXXXXXXXX	<a href="#">Edit</a>	<a href="#">Edit</a>	<a href="#">View</a>	Nov 05, 2015	Jan 03, 2016	<a href="#">Download Agreement</a>
World	Free Applications	XXXXXXXXXX	N/A	N/A	N/A	Dec 07, 2015	Jan 03, 2016	N/A

Transfer Agreements

Once an app transfer has been requested, the recipient's Team Agent has 60 days from the day initiated to accept the request. Contracts that have not been accepted within 60 days will be canceled. App transfers can take up to two business days to complete, depending on an export compliance review.

You have no Transfer Agreements.

[Watch the help video](#)

[View Pricing Matrix](#)

Done

合約中的聯絡資訊、銀行資料及稅務都必需填寫。

# 基礎稅務問題與資訊說明

所有的個人或公司都必需填入美國稅務的申報書。

## Tax Information

Review the tax forms below and complete the ones applicable to your situation. Note that U.S. tax forms are required for everyone.

### U.S. Tax Forms

Required for selling apps.

Set Up

### Australia Tax Forms

Required for developers based in Australia or registered for Australian GST.

Set Up

### Canada Tax Forms

Required for developers based in Canada or registered for Canadian GST/HST.

Set Up

For developers residing outside of Japan who want to sell apps on the Japan App Store, [learn more](#) about optional Japanese tax forms.

如果不是美國公民，或事業單位不在美國境內，請選擇 NO

Are you a U.S. citizen, U.S. resident, U.S. partnership, or U.S. corporation?

Yes ☐ No ☐

In general, you are considered a U.S. resident if (1) at any time during the calendar year you were a lawful permanent resident of the United States according to United States immigration laws and this status has not been rescinded or administratively or judicially determined to have been abandoned or (2) you have been physically present in the United States on at least 31 days during the current year, and 183 days during the 3 year period that includes the current year and the 2 years immediately before. Click [here](#) for more information about the definition of a "U.S. resident".

Back

Submit

如果在美國境內沒有商業活動進行中，請選擇 NO

Do you have any U.S. Business Activities?

Yes ☐ No ☐

In general, you have U.S. Business Activities if you have employees in the United States, or own, lease or control equipment or other assets in the United States that you use to derive revenue from the Apple Developer Program.

Back

Submit

聲明自己同意合約及相關受益事項，請在 Title 處填入職稱。（個人帳號可以直接填 Product Owner 就好）

☒ I declare that the individual or organization named in the iPhone Developer Program License Agreement is the beneficial owner of any payments made under such agreement. I declare that the beneficial owner does not have any employees in the United States and does not own, lease or control any equipment or other assets in the United States that are used to derive revenue from the Apple Developer Program. I declare that I am either the beneficial owner or that I am authorized to make this declaration on behalf of the beneficial owner.

Who is authorized to sign this certificate?

Name of Person Making this Declaration:  [Not You?](#)

Title:

在第九題選擇收入來源為販售 App 為獲利來源的選項。

**Part II: Claim of Tax Treaty Benefits: (If Applicable)**

9. ☐ I certify that the beneficial owner is a resident of Taiwan within the meaning of the income tax treaty between United States and that country.

10. Special rates and Conditions: The beneficial owner is claiming the provisions of Article  of the treaty identified on line 9a (if applicable – see instructions)

above to claim a  % rate of withholding on (specify type of income):

☒ Income from the sale of applications.

☐ Other (please specify)

為確保為本人簽屬，請於最後一處填入自己的名字。

Who is authorized to sign this certificate?

Beneficial Owner (or individual authorized to sign for the beneficial owner):  [Not You?](#)

Capacity in which acting:

Cancel

Submit



## 銀行資訊

填入自己的銀行資訊，以及開戶名稱（通常是你的護照名或信用卡上的名字，只能輸入英文），在輸入時需要輸入 Taiwanese Bank Code，此時請輸入你的銀行代碼就好。例如中華郵政為 700，中國信託為 822。

### Banking Information

Bank Country: Taiwan  
Taiwanese Bank Code: 822

Bank Name: **CTBC BANK CO LTD**  
Bank Address: 3, Sung Shou Road, Sinyi District, Taipei, 11051, Taiwan

Bank Account Number:  ?

Confirm Bank Account Number:

Account Holder Name:  TSAI ?

Bank Account Currency:  TWD - New Taiwan Dollar ?

Telephone Number:

Can't find your Bank Account Currency? [contact us](#)

# 在 iTunes Connect 上建立商品

請點選需要提供商品的 App，並點選功能 > App 內購買項目，並開始建立商品。

**消耗性項目**  
使用者每次下載都必須購買消耗性「App 內購買項目」。一次性服務 (例如釣魚 App 中的魚飼料) 便通常以消耗性產品形式執行。

[選取](#)

**非消耗性項目**  
使用者只需購買非消耗性的 App 內購買項目一次。沒有使用期限、不會因使用而減少的服務通常都屬於非消耗性項目，例如遊戲 App 中的新賽道。

[選取](#)

**自動續訂型訂閱**  
「自動續訂型訂閱」可讓使用者在設定的期限內，購買更新與動態內容。除非使用者選擇退出，否則會自動續訂，例如訂閱雜誌。

[選取](#)

**免費訂閱**  
免費訂閱是開發人員宣傳「報章雜誌」內容的最佳方式。只要使用者註冊免費訂閱，就能在與使用者 Apple ID 相關聯的所有裝置上取得訂閱內容。免費訂閱沒有使用期限，但僅適用於屬於「報章雜誌」類別的 App。

[選取](#)

**非續訂型訂閱**  
「非續訂型訂閱」只允許在一段限定期間內銷售服務。「非續訂型訂閱」必須用於提供限時存取靜態內容的「App 內購買項目」，像是訂閱一週的導航類 App 語音指示功能，或是訂閱一年已封存影片或音訊的線上目錄。

[選取](#)

每個商品都會有自己的參照名稱，這個名稱只是在未來查看銷表報表時讓你辨識的名字而已，不會顯示在使用者前。另外特別重要的是產品 ID，我們未來會在程式碼中使用該資訊，提供使用者購買商品。

## App 內購買項目摘要

輸入此「App 內購買項目」的參照名稱與產品 ID。

參照名稱：

產品 ID：

**定價與供應狀況**

為下方的「App 內購買項目」輸入價格與供應狀況的詳細資訊。

准許銷售 ☒ 是 ☐ 否

價格層級

[檢視定價對照表](#)

## 專案設定

在 iTunes Connect 建立商品後，請開啟專案。請確認你的專案 Bundle ID 與剛才在 iTunes Connect 的套裝組 ID 相符。



General Capabilities Resource Tags Info Build Settings Build Phases Build Rules

▼ Identity

iTunes Connect 上的畫面

一般資訊

套裝組 ID ? 註冊新套裝組 ID

選擇

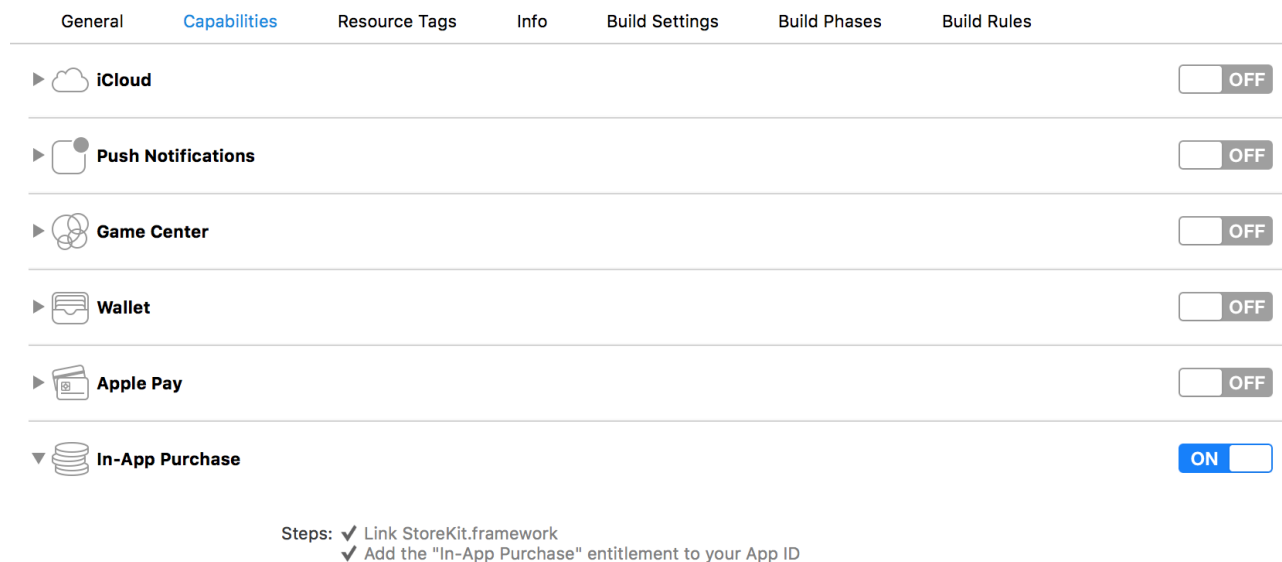
您的套裝組 ID tw.pitcher

Bundle Identifier tw.pitcher

Version 0.1

Build 1

接著請至 Capabilities 開啟 In-App Purchase



General Capabilities Resource Tags Info Build Settings Build Phases Build Rules

► iCloud OFF

► Push Notifications OFF

► Game Center OFF

► Wallet OFF

► Apple Pay OFF

▼ In-App Purchase ON

Steps: ✓ Link StoreKit.framework  
✓ Add the "In-App Purchase" entitlement to your App ID

在需要的地方 `#import <StoreKit/StoreKit.h>`

## 宣告商品 ID

若要查詢商品供應狀況，請先確認該類別已經採用 SKProductsRequestDelegate Protocol。

```
@interface StoreTableViewController ()<SKProductsRequestDelegate>
@end
```

將所有產品 ID 都宣告至 NSSet：

```
[NSSet initWithObjects:@"tw.pitch.unlimited",@"tw.pitch.30coin", nil]
```

Apple 其實建議各位開發者，如果你經常變動購買項目的話，可以不用把產品 ID 寫死在 App 中，可以將 ID 用文字檔放在伺服器上，每次 App 執行時去更新文字檔，讓使用者永遠都能取得最新的可購買商品，而你不需要每次提供新商品就將 App 重新送審。

所以每次更新完這份清單後，最好確認一下是不是都能正常拿到商品。如果有任何商品無法使用的話，就會出現在 invalidIdentifier 裡。

## 查詢商品供應狀況

查詢商品的程式碼會用到 SKProductsRequest 類別，這個類別採用 Delegate 設計模式，所以需指派委託人。

```
SKProductsRequest * productRequest = [[SKProductsRequest alloc] initWithProductIdentifiers:
[NNSSet initWithObjects:@"tw.pitch.unlimited",@"tw.pitch.30coin", nil]];
productRequest.delegate = self;
[productRequest start];
```

查詢後的結果會透過 Delegate 觸發：

```

-(void)productsRequest:(SKProductsRequest *)request didReceiveResponse:(SKProductsResponse *)response
{
    for (NSString *invalidIdentifier in response.invalidProductIdentifiers) {
        // Handle any invalid product identifiers.
        NSLog(@"%@",invalidIdentifier);
    }
}

```

如果有任何無法使用的商品，會列在 `response.invalidProductIdentifiers`。而可以用的商品，會被做成 `SKProduct` 的物件，存放在 `response.products`。如此一來，你就可以拿這裡面的商品顯示在自己的商城上，提供使用者購買。

## 設計自己的商城

我們利用一個 `TableView` 設計自己的商城，當 `response.products` 的商品數量大於 0 時，就重新整理 `TableView`，使用者按下去後就可以開始購買。

```

-(void)productsRequest:(SKProductsRequest *)request didReceiveResponse:(SKProductsResponse *)response
{
    for (NSString *invalidIdentifier in response.invalidProductIdentifiers) {
        // Handle any invalid product identifiers.
        NSLog(@"%@",invalidIdentifier);
    }
    if (response.products.count > 0) {
        // 如果商品數量大於 0，就將商品陣列指給 TableView 在讀的陣列
        self.products = response.products;
        [self.tableView reloadData];
    }
}

```

設計自己的商城時，你會需要顯示商品的價格跟資訊，這些資訊都可以在 `SKProduct` 物件中拿到。

```

SKProduct * product = [self.products objectAtIndex:indexPath.row];
cell.textLabel.text = product.localizedTitle; //會顯示多語系的商品名稱

```

```
cell.detailTextLabel.text = product.price.stringValue; //商品價格
```

## 建立購買測試帳號

接著將嘗試購買。不過我們不會真的花錢去購買，所以會利用 Apple 提供的沙箱帳號讓我們測試購買功能是不是能正常運作。所以你需要建立一組自己的購買帳號。

請到 iTunes Connect 中的使用者和職能 > 沙箱技術測試人員 新增。

測試人員 (1) 

[編輯](#)

電子郵件	姓名 ^	iTunes Store
 test@gmail.com	 蔡	 台灣

## 購買商品的前置準備

購買商品的結果是透過 Observer 的模式在進行。

```
[[SKPaymentQueue defaultQueue] addTransactionObserver:self];
```

另外別忘記在 dealloc 的地方，移除註冊的 observer

```
[[SKPaymentQueue defaultQueue] removeTransactionObserver:self];
```

另外，需加入下面這幾個程式碼，採用 SKPaymentTransactionObserver 的 Protocol，實作下列方法：

```
- (void)paymentQueue:(SKPaymentQueue *)queue updatedTransactions:
(NSArray<SKPaymentTransaction *> *)transactions
```

這個方法屆時會回傳購買的結果，而我們通常會關注下列幾個狀態：

```
- (void)paymentQueue:(SKPaymentQueue *)queue updatedTransactions:
(NSArray<SKPaymentTransaction *> *)transactions NS_AVAILABLE_IOS(3_0)
{
    for (SKPaymentTransaction * transaction in transactions) {
        switch (transaction.transactionState)
```

```

{
    case SKPaymentTransactionStatePurchased:// 購買成功
        break;
    case SKPaymentTransactionStateFailed: //購買失敗
        break;
    case SKPaymentTransactionStateRestored: //回復成功
        break;
    default:
        break;
}
};
}

```

## 購買商品

在查詢完商品可用性後，就可以拿 StoreKit 回傳的 SKProduct 來建立 SKPayment 物件。並且在真正購買前，先檢查一下購買功能是否已經啟用（有時候會因為家長監控的功能導致無法購買）。

```

if ([SKPaymentQueue canMakePayments]) {
    SKProduct * product = [self.products objectAtIndex:indexPath.row];
    SKPayment * payment = [SKPayment paymentWithProduct:product];
    [[SKPaymentQueue defaultQueue] addPayment:payment];
}

```

將 SKProduct 包至 SKPayment 物件後，就能用 SKPaymentQueue 的 defaultQueue 這種 Singleton 物件開啟購買通道。購買結果會透過上述的 Delegate 模式觸發。

## 結帳完畢

當使用者購買成功後，你也將相關的功能開啟後，並保存收據後，就可以結束這次的交易。

```

[[SKPaymentQueue defaultQueue] finishTransaction: transaction];

```

## 保存收據

自 iOS 7 之後，Apple 推出了 App Receipt，如果你需驗證使用者是否合法持有你的 App 並執行，可使用 App Receipt 驗證使用者是否有購買你的 App 紀錄。不過這裡要關注的，是如果你提供的商品是訂閱制的，或是非消耗品，你可能會希望將消費者的購買收據保存下來，向 Apple 驗證收據的合法性後，再提供完整版的服務。

除了消耗性商品在結束交易後，不會主動保留收據外，其餘類型的商品都會將其保存在 App 內，其位置在：

```
[NSBundle mainBundle].appStoreReceiptURL
```

如果你為 iOS6（含）以前打造應用，可能需要手動保存收據，可從每次的 Transaction 交易物件裡，拿到收據的原始檔。

```
@property(nonatomic, readonly, nullable) NSData *transactionReceipt
```

## 更新收據

當使用者更換手機，或將 App 移除再重新安裝後，之前購買的收據將會遺失。你可以透過 SKReceiptRefreshRequest 更新收據。更新完畢後，會透過 Delegate 觸發 SKProductRequestDelegate 等 Protocol 方法。

```
SKReceiptRefreshRequest * refresh = [[SKReceiptRefreshRequest alloc] init];  
refresh.delegate = self;  
[refresh start];
```



## 向 Apple 驗證收據

在正式環境中與在沙箱環境中完成的購買，其驗證網址不同。

環境	網址
沙箱環境	<a href="https://sandbox.itunes.apple.com/verifyReceipt">https://sandbox.itunes.apple.com/verifyReceipt</a>
正式環境	<a href="https://buy.itunes.apple.com/verifyReceipt">https://buy.itunes.apple.com/verifyReceipt</a>

```
// 取得收據，並轉成 JSON 物件
NSURL *receiptURL = [NSBundle mainBundle].appStoreReceiptURL;
NSData *receipt = [[NSData alloc] initWithContentsOfURL:receiptURL];
NSError *error;
NSDictionary *requestContents = @{
    @"receipt-data": [receipt base64EncodedStringWithOptions:0]
};
NSData *requestData = [NSJSONSerialization dataWithJSONObject:requestContents
                                                    options:0
                                                    error:&error];

// 驗證網址並使用 POST 向 iTunes Store 驗證。
NSURL *storeURL = [NSURL URLWithString:@"https://sandbox.itunes.apple.com/verifyReceipt"];
NSMutableURLRequest *storeRequest = [NSMutableURLRequest requestWithURL:storeURL];
[storeRequest setHTTPMethod:@"POST"];
[storeRequest setHTTPBody:requestData];

// 建立非同步連線並開始驗證
NSOperationQueue *queue = [[NSOperationQueue alloc] init];
[NSURLConnection sendAsynchronousRequest:storeRequest queue:queue
                    completionHandler:^(NSURLResponse *response, NSData *data, NSError
*connectionError) {
    if (connectionError) {
        /* ... Handle error ... */
    } else {
        NSError *error;
        NSDictionary *jsonResponse = [NSJSONSerialization
JSONObjectWithData:data options:0 error:&error];
```

```
//收據內容在此
NSLog(@"收據內容:%@", jsonResponse);
}
};
```

## 收據內容的範例

拿到從 iTunes Store 回傳的收據後，就可以檢查收據裡使用者有沒有購買特定項目，藉此提供功能。

```
{
    environment = Sandbox;
    receipt = {
        "adam_id" = 0;
        "app_item_id" = 0;
        "application_version" = 1;
        "bundle_id" = "tw.pitcher";
        "download_id" = 0;
        "in_app" = (
            {
                "is_trial_period" = false;
                "original_purchase_date" = "2015-11-08 07:08:08 Etc/GMT";
                "original_purchase_date_ms" = 1446966488000;
                "original_purchase_date_pst" = "2015-11-07 23:08:08 America/Los_Angeles";
                "original_transaction_id" = 1000000179213452;
                "product_id" = "tw.pitch.unlimited";
                "purchase_date" = "2015-11-08 07:08:08 Etc/GMT";
                "purchase_date_ms" = 1446966488000;
                "purchase_date_pst" = "2015-11-07 23:08:08 America/Los_Angeles";
                quantity = 1;
                "transaction_id" = 1000000179213452;
            }
        );
        "original_application_version" = "1.0";
        "original_purchase_date" = "2013-08-01 07:00:00 Etc/GMT";
        "original_purchase_date_ms" = 1375340400000;
        "original_purchase_date_pst" = "2013-08-01 00:00:00 America/Los_Angeles";
    }
}
```

```
"receipt_creation_date" = "2015-11-08 11:33:33 Etc/GMT";
"receipt_creation_date_ms" = 1446982413000;
"receipt_creation_date_pst" = "2015-11-08 03:33:33 America/Los_Angeles";
"receipt_type" = ProductionSandbox;
"request_date" = "2015-11-08 12:04:49 Etc/GMT";
"request_date_ms" = 1446984289711;
"request_date_pst" = "2015-11-08 04:04:49 America/Los_Angeles";
"version_external_identifier" = 0;
};
status = 0;
}
```

## 還原之前的購買項目

如前文所提，除了消耗性項目外，其它商品都需提供使用者回復的管道。其中幾項可以用 StoreKit 內建的方法來還原購買項目。

```
[[SKPaymentQueue defaultQueue] restoreCompletedTransactions];
```

## Apple 參考資料

### 1. Apple IAP Guideline

<https://developer.apple.com/in-app-purchase/In-App-Purchase-Guidelines.pdf>

### 2. iTunes Connect 商品上架資訊

[https://developer.apple.com/library/ios/documentation/LanguagesUtilities/Conceptual/iTunesConnectInAppPurchase\\_Guide/Chapters/Introduction.html](https://developer.apple.com/library/ios/documentation/LanguagesUtilities/Conceptual/iTunesConnectInAppPurchase_Guide/Chapters/Introduction.html)

### 3. IAP 開發指南

<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StoreKitGuide/Introduction.html>