

Mobile Service > IAP > iOS SDK Guide

[Notice]

New IAP SDK supporting subscription are released as [TOAST SDK](#).

The existing IAP SDK will not be developing new features.

Development Environment

- OSX is required
- Xcode 6.0.1 and higher
- IAP SDK Support for iOS 6.x and higher

Add below framework to application to use IAP SDK.

Name	Description
StoreKit.framework	Framework for in-app purchase sync in App Store
libsqlite3.tbd	TOAST Cloud IAP SDK uses sqlite for managing local data
coreTelephony.framework	This is for acquiring user's country information.

[Reference]

Suppose you have completed registering application/ item to iTunes Connect in order to test in-app purchase.

[iTunes Connect](#)

IAP Console

1. Store Registration - Getting APP ID

1. select [App] tab > click [Add] button
2. select AS(Apple Store) in [Store ID]
 - Market APP ID : Apple App Store Bundle Id
3. check [APP ID].

App

Item

Transaction

Statistics

+ Add

Modify

X Delete

☐	App ID	App 이름	Store ID(스토어 이름)	상태	생성일
☐	1000283	IAP TEST 1	AS	USE	2017-02-14 15:50:06
☐	1000284	IAP TEST 2	TEST	USE	2017-02-14 15:50:16
☐	1000285	IAP TEST 3	TS	STOP	2017-02-14 15:50:29

< >

[Figure 1 Getting APP ID]

2. Item Registration

1. select [Item] tab > click [Add] button
2. select AS(Apple Store) in [Store ID]
3. register item
 - Item Name : item name
 - Store Item ID : item Product ID registered in iTunes Connect
4. check [ITEM]

Setting Xcode Project

Directory Name	Description
/docs	IAP iOS SDK API Reference
/lib	Framework
/samples	Sample Application

[Table. 1 iOS SDK Directory]

1. Add IAP SDK and framework

1. [Xcode] > [Project] > [Targets – Build Phases]
2. Add following frameworks to [Link Binary With Libraries]
 - libsqlite3.tbd
 - TIAPurchase.framework
 - StoreKit.framework
 - CoreTelephony.framework

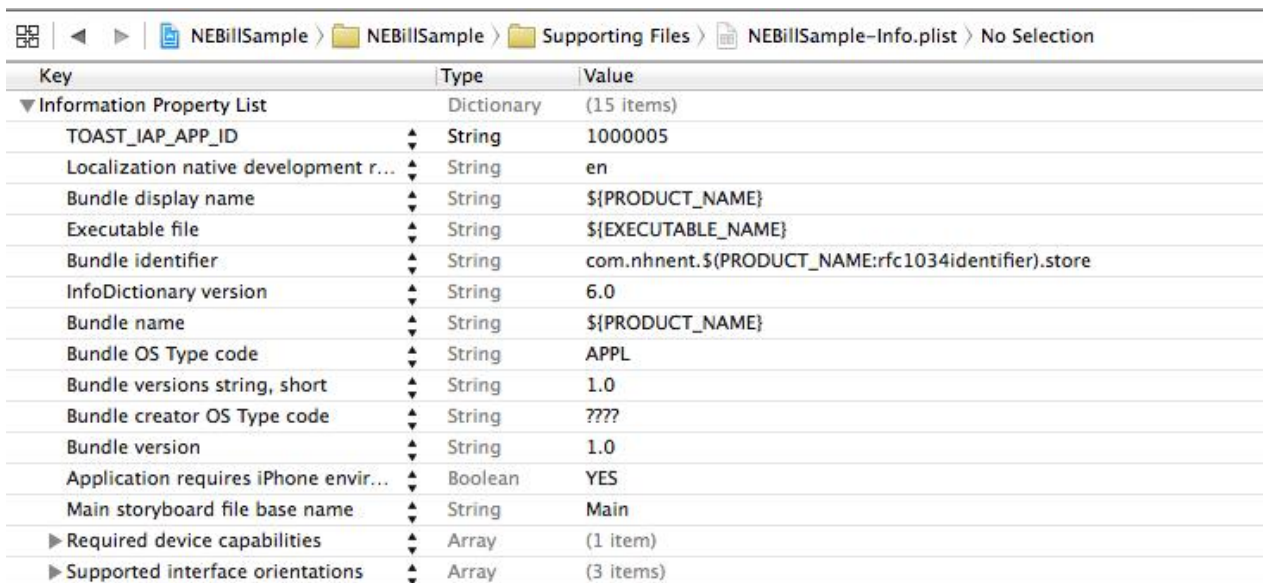
▼ Linked Frameworks and Libraries

Name	Status
 TIAPurchase.framework	Required ⇅
 libsqlite3.tbd	Required ⇅
 CoreTelephony.framework	Required ⇅
 StoreKit.framework	Required ⇅
 CoreGraphics.framework	Required ⇅
 UIKit.framework	Required ⇅
 Foundation.framework	Required ⇅
+ –	

[Figure 2 Add library for IAP]

2. Setting plist

Create string value with TOAST_IAP_APP_ID key to [plist] and enter app ID. Once completed, .plist will be in the following format.



Key	Type	Value
▼ Information Property List	Dictionary	(15 items)
TOAST_IAP_APP_ID	String	1000005
Localization native development r...	String	en
Bundle display name	String	\$(PRODUCT_NAME)
Executable file	String	\$(EXECUTABLE_NAME)
Bundle identifier	String	com.nhnent.\$(PRODUCT_NAME:rfc1034identifier).store
InfoDictionary version	String	6.0
Bundle name	String	\$(PRODUCT_NAME)
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	????
Bundle version	String	1.0
Application requires iPhone envir...	Boolean	YES
Main storyboard file base name	String	Main
► Required device capabilities	Array	(1 item)
► Supported interface orientations	Array	(3 items)

[Figure 3 Set App ID to plist]

[Reference - Setting iOS9 ATS]

Apply ATS (App Transport Security) setting from iOS9 SDK When building in system beyond XCode7, you need to set exceptions for certain IAP domains as below.

[Apple Document](#)

```
<key>NSAppTransportSecurity</key>
  <dict>
    <key>NSExceptionDomains</key>
    <dict>
      <key>api-iap.cloud.toast.com</key>
      <dict>
        <key>NSIncludesSubdomains</key>
        <true/>
        <key>NSExceptionAllowsInsecureHTTPLoads</key>
        <true/>
        <key>NSExceptionRequiresForwardSecrecy</key>
        <false/>
      </dict>
    </dict>
  </dict>
```

API Reference

1. Import Header File

If you are ready to use SDK in application, add header file to IAP SDK as below.

```
#import "TIAPurchase.h"
```

2. Enable Log

Activate log information exposure for debugging.

[Example Code]

```
[TIAPurchase setDebugMode:YES];
```

3. Registration User

Register user identifier value after user verification in application

[Example Code]

```
NSError *purchaseError = nil;
BOOL result = [TIAPurchase registerUserId:@"user001" error:&purchaseError];
If (!result) {
    // An error occurred, we need to handle the error.
    NSLog(@"errorInfo = %@", purchaseError);
}
// register user id succceccfully.
```

4. Request Payment

Send in-app purchase request. Once payment is successfully done, payment details will be sent via completionHandler.

[Example Code]

```
[TIAPurchase startPurchaseWithViewController:self itemId:1000004
completionHandler:^(id result, NSError *error) {

    if (error)
    {
        // An error occurred, we need to handle the error
        NSLog(@"purchase error, %@ %d", [error domain], [error code]);
        return;
    }
    else
    {
        /**
         Success! Include your code to handle the results here

         JSON data to 'NSDictionary', This is nil if there was an error.
         [keys]
         - paymentSeq : generated payment id
        */
    }
}
```

```

        - itemSeq : item id
        - purchaseToken : represent to token for consume by
server.
        - currency : represent to item currency
        - price : represent to item price.
    */

    NSDictionary *purchaseResult = result;
    NSLog(@"purchase success, purchase = %@", purchaseResult);
}

}];

```

5. Inquiry unconsumed payment

Inquire unconsumed payment history.

[Example Code]

```

[TIAPurchase purchasesWithCompletionHandler:^(id result, NSError *error) {
    if (error)
    {
        // An error occurred, we need to handle the error
        NSLog(@"purchasesWithCompletionHandler occured error, %@ %d", [error
domain], [error code]);
        return;
    }

    /**
    Success! Include your code to handle the results here

    JSON data to 'NSArray', This is nil if there was an error.
    [keys]
    - paymentSeq : generated payment id
    - itemSeq : item id
    - purchaseToken : represent to token for consume by server.
    - currency : represent to item currency.
    - price : represent to item price.
    */

    NSArray *purchases = result;
    NSLog(@"purchasesWithCompletionHandler, size:%d npurchases:%@", [purchases
count], purchases);

}];

```

6. Inquiry all purchasable items

Inquire all purchasable items

[Example Code]

```
[TIAPurchase itemListWithCompletionHandler:^(id result, NSError *error) {
    if (error)
    {
        // An error occurred, we need to handle the error
        NSLog(@"itemListWithCompletionHandler occurred error, %@ %d", [error
domain], [error code]);
        return;
    }

    /**
    Success! Include your code to handle the results here

    JSON data to 'NSArray', This is nil if there was an error.
    [keys]
    - itemSeq : item id
    - itemName : item name
    - usingStatus : item status on IAP server
    - regYmdt : item registration date on IAP server
    - appName : app name
    - marketId : market id (AS : APPLE STORE)
    - marketItemId : market item id (product id)
    - currency : represent to item currency
    - price : represent to item price
    - localizedPrice - represent to localized item price
    */

    NSArray *itemList = result;
    NSLog(@"itemListWithCompletionHandler, size:%lu \nitemList:%@", [itemList
count], itemList);
}];
```

7. Batch process of unconsumed payment

processes whole of unconsumed(cause of verification failure or network loss) payments.

[Example Code]

```
[TIAPurchase processesIncompletePurchasesWithCompletionHandler:^(id result,
NSError *error) {
    if (error)
    {
        // An error occurred, we need to handle the error
        NSLog(@"processesIncompletePurchasesWithCompletionHandler occurred
error, %@ %d", [error domain], [error code]);
        return;
    }
}
```

```

/**
    Include your code to handle the results here

    JSON data to 'NSDictionary'.
    [keys]
    - successList : success data list (NSArray)
        [keys]
        - paymentSeq : generated payment id
        - itemSeq : represent item id
        - purchaseToken : represent token for validation
        - marketItemId : market item id (product id)
        - currency : represent to item currency
        - price : represent to item price

    - failList : fail data list (NSArray)
        [keys]
        - paymentSeq : generated payment id
        - itemSeq : represent item id
        - purchaseToken : represent token for validation
        - marketItemId : market item id (product id)
        - currency : represent to item currency
        - price : represent to item price

    */
    NSDictionary *data = result;
    NSLog(@"processesIncompletePurchasesWithCompletionHandler data:%@",
data);
}];

```

8. Payment Consume

User application server should notify IAP server for payment consume before issuing item. Use Payment Purchase Token to check validity of the payment between user server and IAP server.

[Reference]

[Payment Consume API](#)

9. App Store Promoting In-App Purchase

If you use App Store promoting in-app purchase, register your delegate and use payment details.

You must register delegate before initialize to receive payment history.

[Example Code]

TIAPurchaseDelegate

```

@interface ViewController () <TIAPurchaseDelegate>
...
@end

// Delegate registration

```

```

[TIAPurchase setDelegate:self];

// Delegate implementation
- (void)didFinishPurchasingItemFromStore:(TIAPItem *)item
    paymentSeq:(NSString *)paymentSeq
    purchaseToken:(NSString *)purchaseToken
    withError:(NSError *)error {

    if (error == nil) {
        NSLog(@"App Store promoting in-app purchase successful");
    } else {
        NSLog(@"App Store promoting in-app purchase failed");
    }
}

```

TIAPItem

```

@interface TIAPItem : NSObject

@property (nonatomic, copy, readonly) NSString *itemSeq;
@property (nonatomic, copy, readonly) NSString *storeId;
@property (nonatomic, copy, readonly) NSString *marketItemId;
@property (nonatomic, copy, readonly) NSString *currency;
@property (nonatomic, copy, readonly) NSDecimalNumber *price;
@property (nonatomic, copy, readonly) NSString *localizedPrice;

@end

```