Android Application In-App Purchase Programming Guide

Version 1.0.2

Enactment & Revision History

Revision No.	Enacted or revised pages and contents	Enactment & Revision Date
1.0.0	• Programming changes to existing guidelines document format • Significantly modify the contents of an existing set of programming guidelines • Real-Time Purchase History Inquiry I / F adds	2012.07.09
1.0.1	• Real-Time Purchase History Inquiry I / F port modifications Notice	2012.07.17
1.0.2	• Notice have been added.	2012.09.17

Table of Contents

List of the Figures

List of the Tables

■ Terminology

In-App Purchase	Purchasing services or products in the Application in addition to the services or products provided by the Application as defaults
IAP Server	Server providing T store In-App Purchase
IAP API	Android library provided by T store In-App Purchase
Application	Application programs developed and sold by developers
Application ID (AID)	Application ID issued during the registration of the Application in T store Developers Center, which is a unique value in T store
In-App Product	Products or additional service sold through In-App purchase
Product ID (PID)	ID issued to In-App Product registered in T store Developers Center, which is a unique value in T store
Application Server	Server accessed for the service provided by the Application, which is directly managed by the developer
Transaction ID	Value transmitted to IAP Server to distinguish individual In-App Product purchases, which can be created within the Application itself or created through IAP API

■ Acronym

AID	Application ID
PID	Product ID
TID	Transaction ID
MDN	Mobile Device Number
IAP	In-App Purchase

■ Reference Site

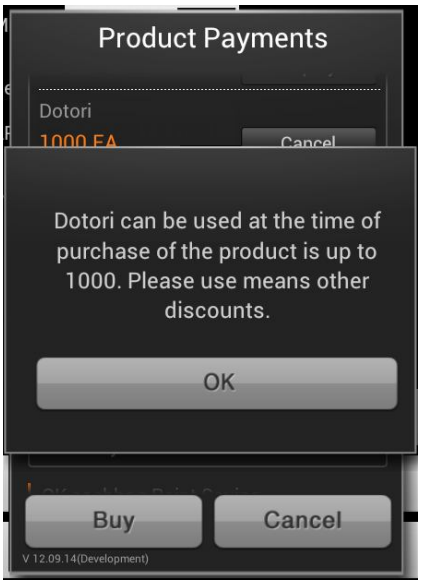
Android	http://developer.android.com/index.html
T store Developers Center	http://dev.Tstore.co.kr/devpoc/main/main.omp

■ Notice

- We announced that there are several changes about distributed library on Friday, September 14th 2012. Developers who are using library should check these changes and upgrade yours.

We expect that you can pursue more stable sale diversification by this upgrade.

SEQ	Items	Contents
1	Deleted "sendBpData()" API.	Is the API to request a data transfer is required to sell membership development server sendBPData () can not use it anymore.
2	Fix the text size error in case of ICS phone.	ICS phone from the Settings menu of the Text Resizing UI Text broken has been fixed.
3	Social security number authentication process of the members of other mobile carriers is changed.	Existing : Once before the first call of the pop-up guidance of payment at the time of the call rises of purchase dialog. Correction: When you press the "Buy" button in the pop-up guidance of the settlement, Only if the settlement amount is greater than $ 0.00 postpaid, call one of the first Correction: When you press the "Buy" button in the pop-up guidance of the settlement, Only if the settlement amount is greater than $ 0.00 postpaid, call one of the first.
4	LGU + SMS authentication process is changed.	Existing: In the case of the members of LG U+ carriers, always proceed SMS auth. Correction: Only if the settlement amount is greater than $ 0.00 postpaid proceed SMS auth.
5	Acorn use limit restriction.	Partial payment methods, which is one of the dotori partial pay when payment products, payment in excess of 10 million(Number 0f 1000) can not. When you try to charge in excess of 10 million (1000), Error pop-up is exposed.

		
6	onJuminNumberDlgCancel() Callback Listener has been deleted.	Authentication Flow settlement is changed, onJuminNumberDlgCancel() listener has been removed. Pressing the resident registration numbers of third-party, the "Cancel" button in the authentication window, to return to the pop-up Payment Product.
7	When releasing the automatic payment of monthly, Change the pop-up phrase (32P~33P)	Automatic payment of the purchased monthly fixed price product withdrawal time, exposed pop-up message of the guide is divided into two Case. Case1) Upon termination within 24 hours after the initial purchase. Case2) After reentrance upon termination in a complete state cancellation within 24 hours after the initial purchase / After 24 hours upon termination.
8	If you do not exist Permission required to AndroidManifest.xml file, add a pop-up routine	If you do not have the required permissions, pop-up box indicates the error. ➔ When initialize library, you should use contain "try-catch".
9	Modified so as to maintain the state of the amount used / unused for each means of payment at the time of switching from the [Payment] pop-up screen.	Maintain the state of the amount used / unused for each means of payment at the time of switching from the [Payment] pop-up screen.

10	SMS authentication, contact Unity3D library.	Added at FAQ No. 8.
11	When pop-up turn it back on again after lock the screen on horizontal mode of pop-up, it appears vertical mode.	Added at FAQ No. 7.
12	Acorn authentication error corrected (when the phone number is 10-digits.)	Both the server and the library was modified.
13	"TID query" error code added.	Error Code (1001,1002) added. (60P)
14	"In-App-Purchase" error code added.	Error Code (2002(5), 2018, 2020) added. (Note the error code table.)
15	When use URLConnection.setDefaultUseCaches(boolean), pop-up appears delayed.	Added at FAQ No. 9.
16	KidsRock operation occurs errors.	Added at FAQ No. 10.
17	OTP auth pop-up modified for visual convenience in horizontal mode.	You can see the OTP auth number also in horizontal mode.

1. Introduction

1.1. Target Readers

This document is targeted for the Application developers who intend to develop In-App Purchase function provided by T store in the Applications sold in T store or Android App. Market.

1.2. Scope

This document describes the T store In-App Purchase and the correct usages of the In-App Purchase API(hereinafter IAP API) required for the development of T store In-App Purchase, in the following order.

- In "Overview", the concept, target customers, available market, product, product model, and settlement for T store In-App Purchase are described.
- In "In-App Purchase Development Procedure", the development preparations, development environment setting, development, test, verification, and sales settlement for the development of a T store IAP API applied application are described according to the overall procedure.
- In "Application Programming Guide", T store IAP API details and the development guideline are described with sample codes, etc.
- In "IAP Server Interface API Guide", T store IAP Server interface specifications are described.

1.3. References

In addition to the description of the development using T store IAP API, the detail information for required Application registration, and sales statistics/settlement, etc. can be obtained in T store Developers Center.

T store Developers Center provides the following services required for Application development and Application sales management.

- Developer Registration
  - Developer registration to T store Developers Center to use T store IAP API
- Contents Production Guide
  - Application production standard conditions required for the registration of an application in T store
- Sales Statistics/Settlement Inquiry
  - Applications registered by developer and the latest sales and settlement details for In-App Product
  - Daily Sales Status / Monthly Sales Status / Tax Invoice Issuance Verification
  - Purchase Cancel List Inquiry

- In addition, additional functions provided by T store Developers Center
  - Technical supports including development supporting guide distribution and development related enquiries
  - Download of development supporting tools (IAP API, Application DRM Verification Tool, Bada OS and Windows Mobile Development Platform)
  - Forums (T store Developers Center, Application/In-App product registration and sales, Development information and opinion exchanges for each platform)

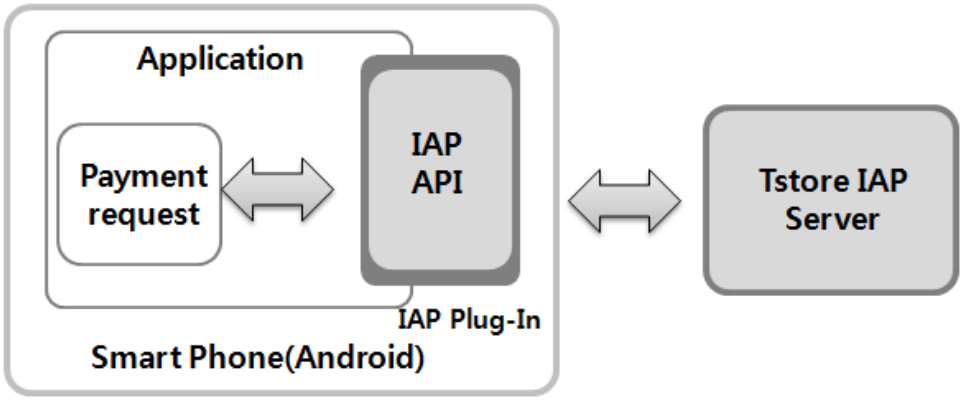
2. Overview

The charged sale of the additional functions or services, other than the basic functions or services provided by the applications purchased by the users, within the applications, to improve the utility value of the applications is called In-App Purchase, and the product sold through In-App Purchase procedure is called In-App Product.

The examples of In-App Product that can be sold through T store are as follows.

- New function or high-level function in a free app.
- Authorization for use of new level in a game
- Intangible digital contents such as E-Book

T store In-App Purchase sends the product information selected by the user to IAP Server of T store through IAP API combined to Application as a Plug-In type to make the payment. Currently, it can be used normally only in Android SDK 2.1 or higher version, and it will be expanded to other platforms in the future.



[Figure 1] T store In-App Purchase

T store In-App Purchase permits the payment only to the Korean citizens who are the subscribers to the Korean mobile carriers and the subscribers of T store at the same time, and Applications with T store IAP can be sold not only in T store, but also in Android Application Markets including "Google Play".

2.1. Types of In-App Product (partial product)

The types of In-App Product(partial product) that can be sold in T store IAP are product per case, periodical product, and conversion to the regular product, and the appropriate In-App product type shall be selected with the considerations of the characteristics of the service to be provided by the application in the stage of planning the application.

The products that can be used without the limitation of usage period after the purchase is called the product per case, and it is divided to permanent product and extinctive product according to

whether the purchased product is maintained permanently.

- Permanent product
  - The product that can be used permanently once it is purchased by the user
  - Duplicate purchase is no possible based on the device number(MDN.)
- Extinctive product
  - The product with the number of times(or the number of items) that can be used by the purchase
  - For example, the product type that is used by decrementing the number of items such as 100 stars, 100 portions, etc. in the application.
  - Duplicate purchase is possible, and the method and management of the decrementing shall be done by the application or the application server itself, and T store does not provide relevant functions in T store IAP.

The product that can only be used for designated period after the purchase is called the periodical product, and it is divided as follows according to the usage period of the product. Duplicate purchase of all of the corresponding products is impossible based on the device number(MDN). But, the product can be purchased again after the usage period of the product.

- Daily product
  - Product with the usable period of 24 hours from the time of purchase
- Weekly product
  - Product with the usable period of 7 days(168 hours) from the time of purchase
- Monthly product
  - Product with the usable period of 30 days from the time of purchase
- Monthly fixed price product
  - Product that can be periodically purchased monthly
  - Unless the user requests to discontinue to use, the payment is automatically made monthly for 1 year

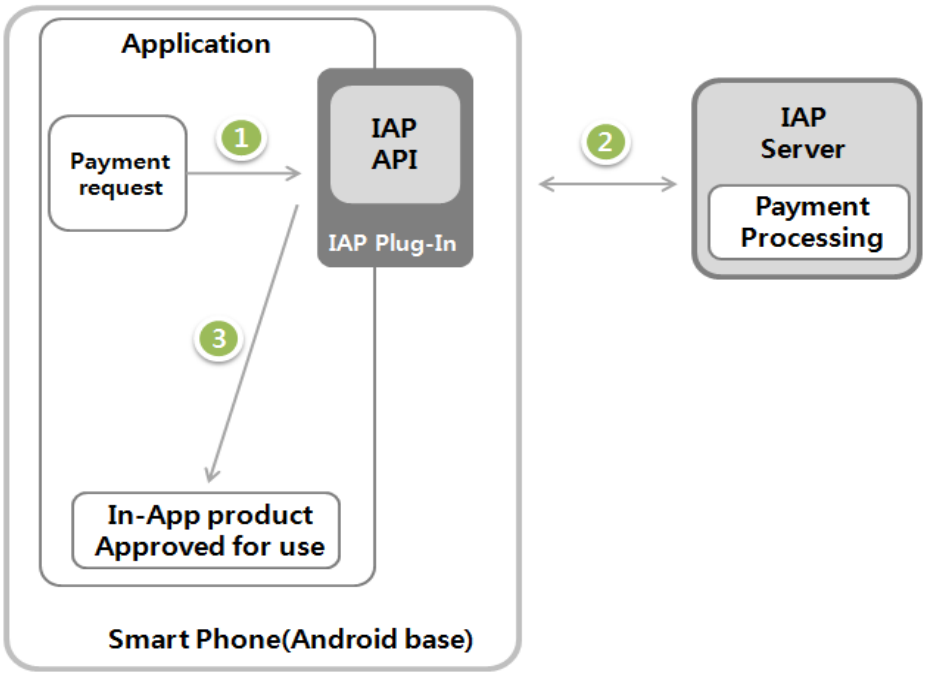
Finally, the product of the conversion to the regular product is the product that downloads the charged regular version application when the user makes In-App payment within the trial application distributed for free of charge.

2.2. In-App Purchase Programming Model

The types of the application developments using T store IAP can be divided to Built0in Model and Server Model according to the method of acquiring the authority to use the purchased In-App product within the application.

2.2.1. Built-in Model

It is the model to unlock the usage lock of the In-App product purchased in the application when In-App product is embedded in the Application and the payment is completed. This model is used to immediately provide In-App Product form the application installed in the smart phone by itself without access to Application Server.



[Figure 2] T store In-App Purchase – Built-in Model

① Call IAP API to request payment.

② Process payment request in IAP Server.

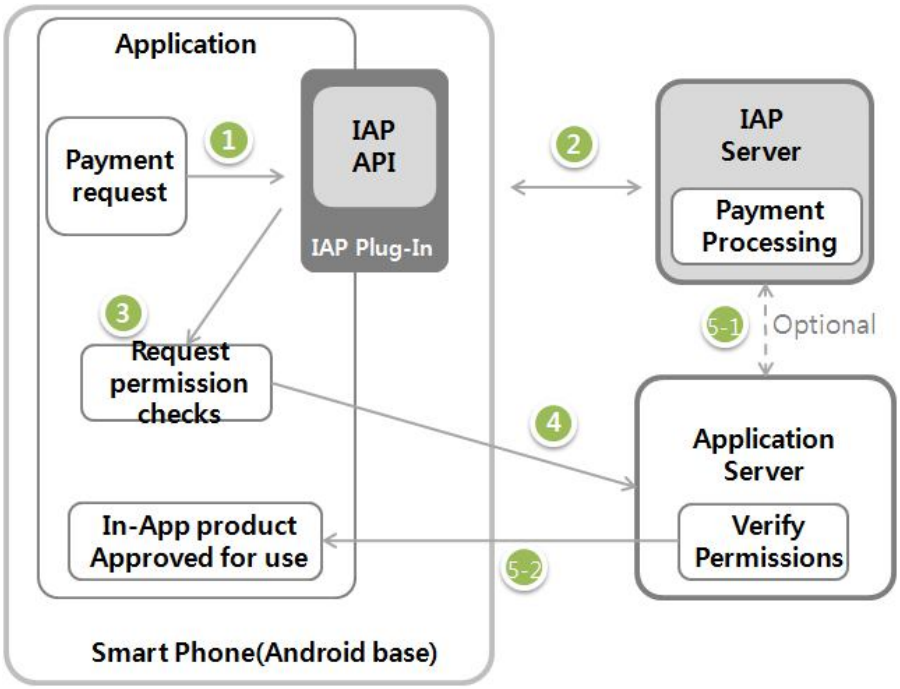
③ Approve the use of the purchased In-App Product.

2.2.2. Server Model

It is the model used to access Application server from the application to request the usage right for In-App Product and receive approval, or when it is necessary to download the corresponding product, after the payment for In-App Product. This model is mostly used to provide In-App Product of extinctive product types or when it is necessary to download In-App Product in Multimedia contents format from the server.

In T store In-App Purchase, no specific model is imposed/enforced for the interface model between the application and Application server to verify the usage right of In-App Product. The corresponding model can be freely developed by the developer.

Server Programming Model can be divided to Download type and Streaming type according to the location of the purchased In-App Product. Download type has the purchased In-App Product within the Application and accesses Application Server to verify the usage rights, and Streaming type has the purchased In-App Product in the Application Server, so it has to access the Application Server for each usage. Streaming type has to make purchase history inquiry to IAP Server for the verification of the usage right (whether the corresponding user purchased the In-App Product), and refer to "Chapter 5. IAP Server Interface API Guide" for the interface specification with IAP Server.



[Figure 3] T store In-App Purchase - Server Programming Model

① Call IAP API to request payment.

② Process payment request in IAP Server.

③ Request the verification of rights for the purchased In-App Product to Application Server.

④ Application Server verifies the usage rights.

⑤ Approve the use of the In-App Product according to the result of the usage right verification from Application Server.

3. In-App Purchase API Development Procedure

This chapter describes the processes from the application development using T store IAP API, application and In-App product registration, to the settlement, according to the procedure.

- Application development preparation
  - Descriptions of the procedures of API download, Application registration, In-App Product registration, etc. required before the development using IAP API
- Development environment setting
  - Description of the development environment setting to use IAP API
- Development
  - The descriptions of IAP API and Coding examples are described in detail in "Chapter 4. Application Development Guide".
- Test
- Description of the process of the self test using the development IAP Server provided by T store before the registration as a commercial product, after the completion of the development
- Application verification
  - Description of the process of the registration of the application that completed the self test as a commercial product in T store
  - The actual purchase is made through the interface with the commercial IAP server for the application registered as a commercial product.
- Sales and settlement
  - Description of the process for the inquiry of the sales status and settlement information for the application that completed proper product registration in T store

3.1. Application Development Preparations

■ Step 1) IAP API Download

Download the latest IAP library, Sample Code, and API Guide required for T store In-App purchase function development from T store Developers Center.

IAP library may be occasionally updated for the change of service policy, addition of functions, or for a bug patch. When IAP library is updated, it is guided in T store Developers Center as a notice, and it will be notified to the developers registered in T store through e-mails and SMS.

During the development of an application, it shall be developed using the latest library registered in T store Developers Center. Otherwise, In-App Product Purchase in the application may function abnormally, and may not be able to register the application to T store Application.

- In T store Developers Center Homepage [download → In-App API] menu, download "T+store+In-App+Purchase+API.zip". "T+store+In-App+Purchase+API.zip" contains the following contents.

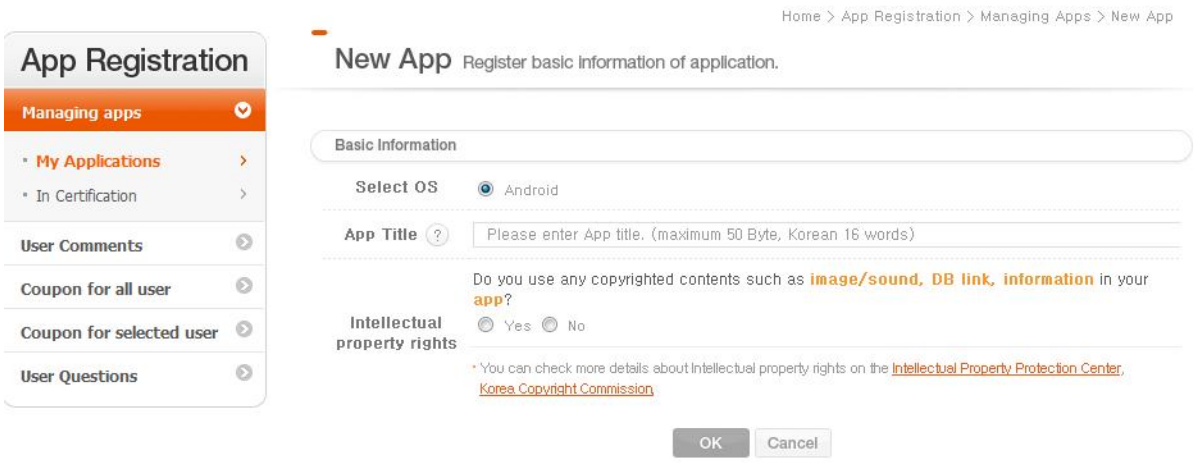
Lib/IAPLibR.jar	T store In-App Purchase library interfacing with commercial IAP server used for Application verification and Applications to be sold to the users.
Lib/ IAPLibD_Eng.jar	T store In-App Purchase library interfacing with development verification IAP server used for initial development and Self Test Application.
Sample Code	T store IAP API application sample source code

[Table 1] T store IAP API Download List

■ Step 2) Application Registration

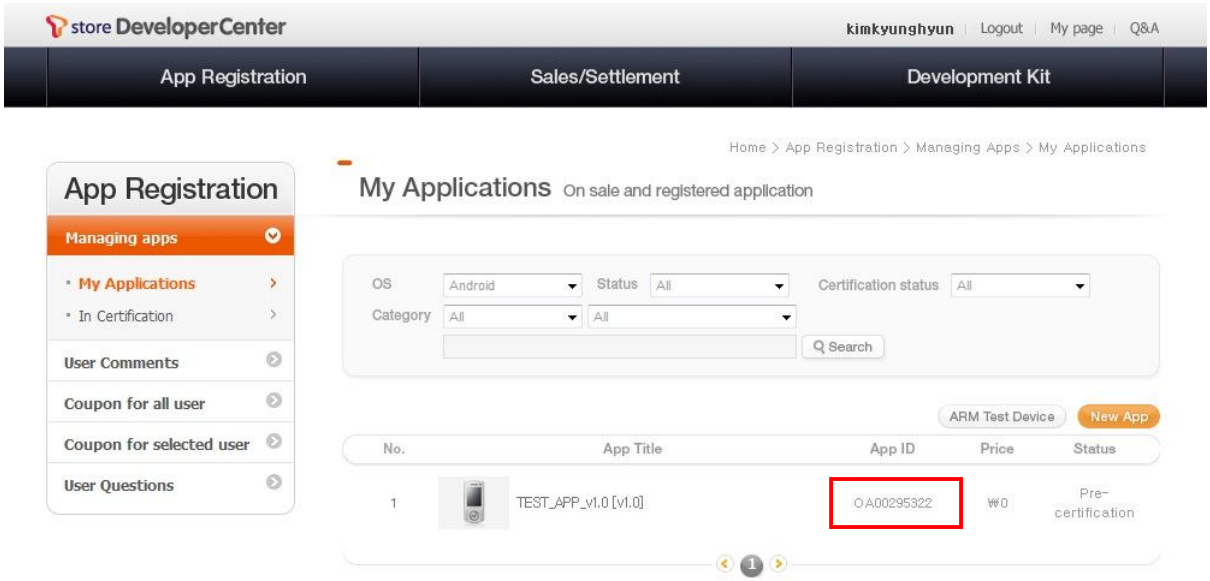
When an application is registered to T store Developers Center, Application ID is automatically issued. This Application ID is used to identify the application during the use of IAP API. Detail description on Application registration is stated in T store Developers Center [Usage Guide].(http://dev.T store.co.kr/devpoc/guide/guideProd.omp#a1_1)

- In T store Developers Center Homepage [Product Registration/Management → Product Status] page, click [Product Registration] button, then [New Registration] page will be displayed for the registration of a new Application. After the completion of the input, click [Registration] button.



[Figure 4] T store Developers Center Application Registration

• You can check the ID of the registered application in [Product Registration/Management → Product Status] page.

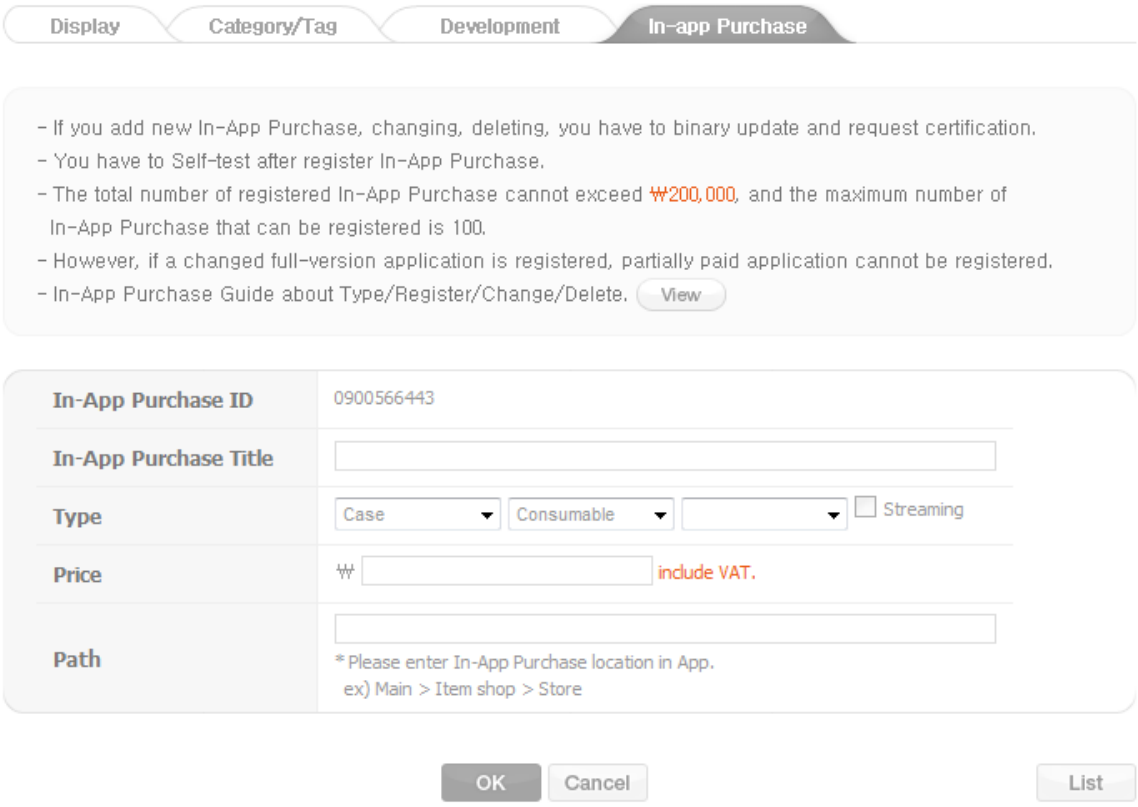


[Figure 5] Application ID Verification

■ Step 3) In-App product Registration

After the Application ID is issued, the developer may register the detail information and In-App product of the corresponding Application. The Product ID issued after the registration of In-App product is used as a parameter to call IAP API within the application along with Application ID. The detail description of In-App Product registration is stated in T store Developers Center [Usage Guide].(http://dev.T store.co.kr/devpoc/guide/guideProd.omp#a1_1)

• In T store Developers Center Homepage [Product Registration/Management → Product Status → In-App Product] page, select [Product Registration] to register new In-App product and receive Product ID.
  - Product type: Select the type of the product that is appropriate for the In-App Product to sell. (Refer to "2.1 Types of In-App Products")
  - Streaming or no streaming: If the format of the IAP to develop is a Streaming Model, check. (Refer to "2.2 In-App Programming Model")
  - App Product Path: Enter the location of the menu where the In-App Product is sold within the application. It is used for self verification by the developer, so there is no restriction on the format.



[Figure 6] T store Developers Center In-App product Registration

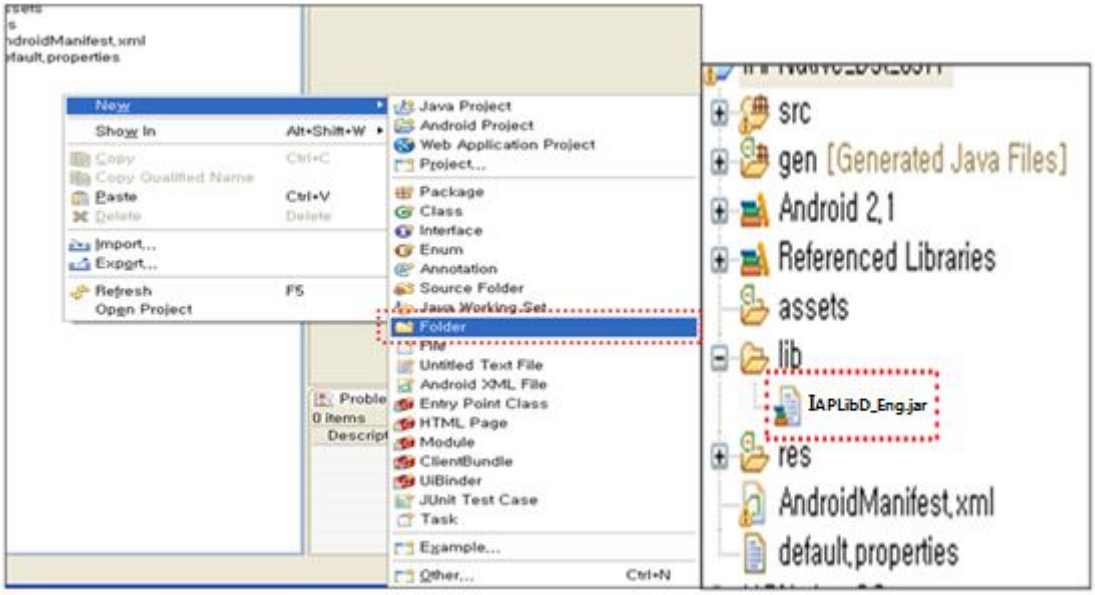
3.2. Development Environment Setting

The method for setting the development environment to include IAP API downloaded from T store Developers Center into Application project is described based on Eclipse. The sample procedures are written based on general usage of Eclipse, and other methods provided by Eclipse may be used for the same result.

■ Step 1) IAP API Addition

① Right click the mouse in Project Explorer to select "New", and set "Folder" to create a new folder.

② Set the name of "Folder" as "lib", copy IAPLibD_Eng.jar file from Windows Explorer, and paste in the corresponding folder. (JDK version 1.7 or higher, set the name of "Folder" as "libs".)



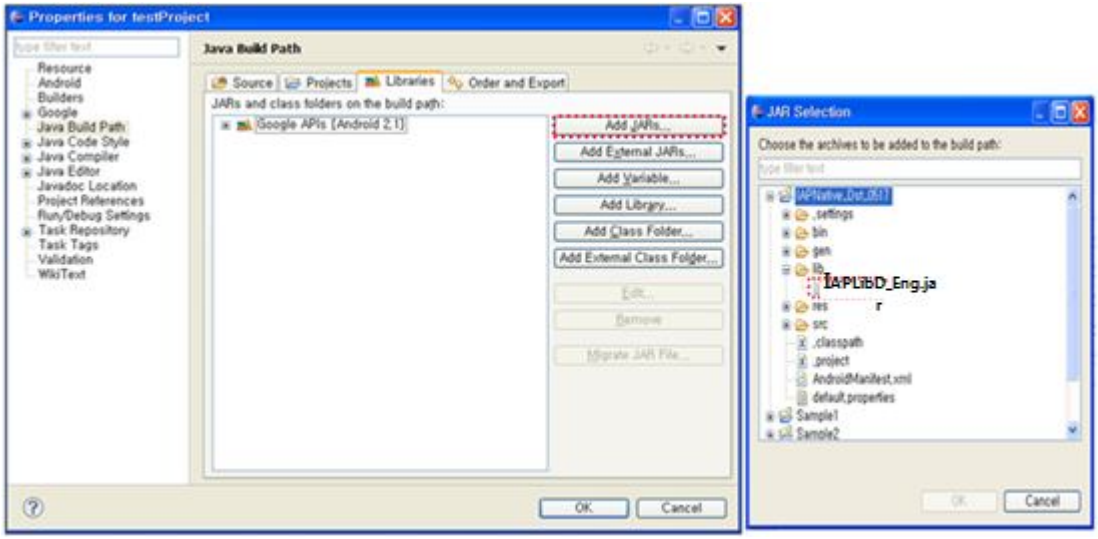
[Figure 7] Adding IAP API in Eclipse Environment

■ Step 2) IAP API Setting

① Right click the mouse in Project Explorer to select "Property", and call the setting window.

② Select Java Build Path in the left side selection tree, and select "Add JAR".

③ When "JAR Selection" window appears, select and set **IAPLibD_Eng**.jar file copied in the previous step.



[Figure 8] IAP API Setting in Eclipse Environment

■ Step 3) AndroidManifest.xml Setting

The following contents must be set in AndroidManifest.xml file for the application using IAP API to function according to the service policy defined by T store. If the following contents are not

set, not only proper operation may be difficult, but the production registration of the application may also fail. The following contents may be added, deleted, or changed according to In-App Purchase policy of T store.

- Kids-Lock

```
<activity android:name="com.feelingk.iap.PwdActivity"
android:label="@string/app_name"/>
<uses-sdk android:minSdkVersion="7" > </uses-sdk>
```

- Add OTP Certification

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

- Add SMS Receiver

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<receiver android:name="com.feelingk.iap.SmsReceiver">
        <intent-filter>
                <action android:name="android.provider.Telephony.SMS_RECEIVED" />
        </intent-filter>
 </receiver>
```

- Add SMS Receiver Permission

```
<uses-permission android:name="android.permission.RECEIVE_SMS" />
```

3.3. Test

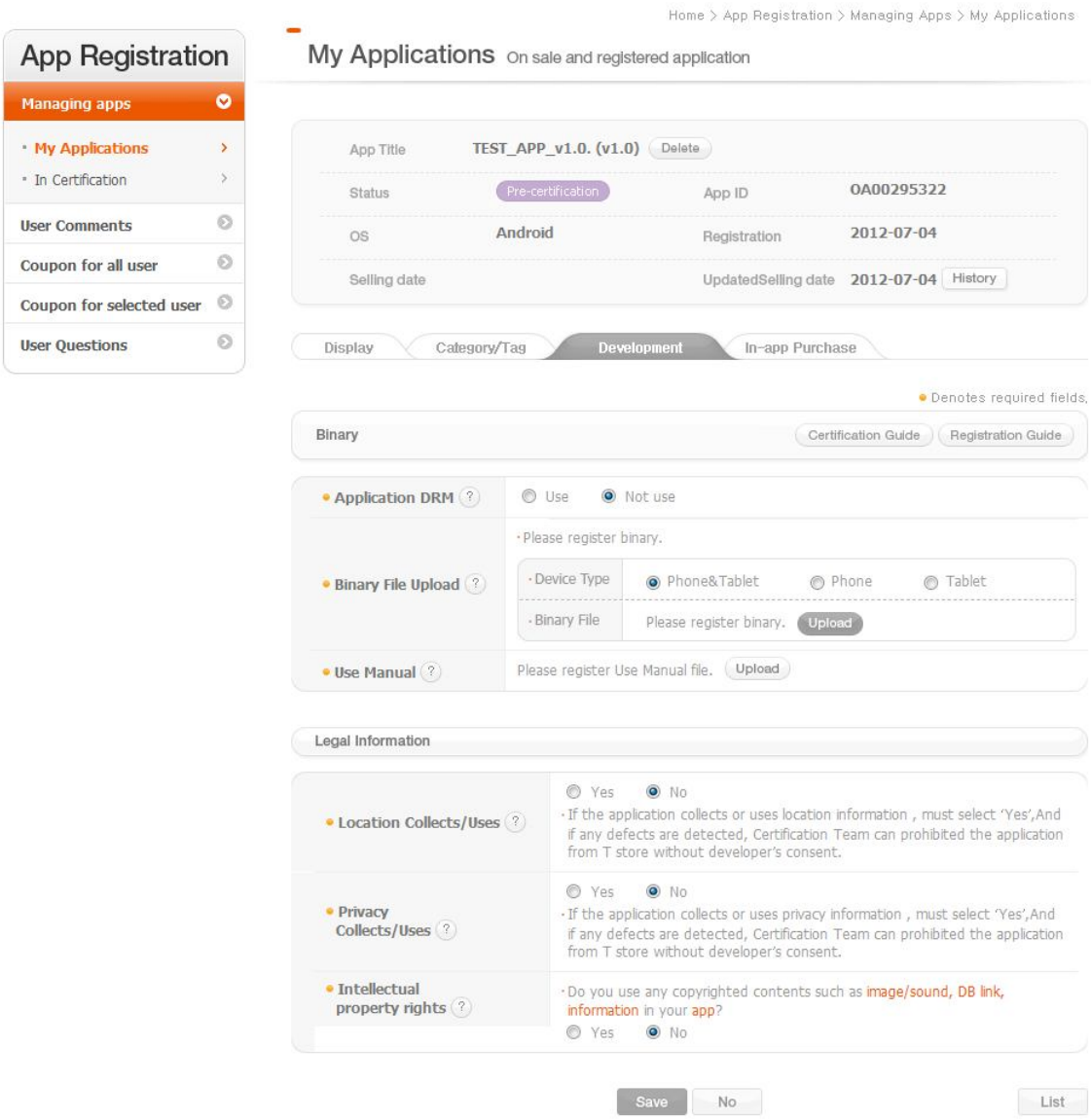
Register APK file and detail information of the development completed Application in T store Developers Center, and verify the application through Self Test process.

☞ There must be charge log and purchase history created by a successful test result to go through the verification procedure and to start the sales of the product in T store.

☞ If an IAP development error is found during the Self Test, the test may be stopped at any time, and it can be tested again by registering the modified APK again. When you register new APK, it is recommended to retest for In-App Product that were successfully tested before.

■ Step 1) Register the Developed Application

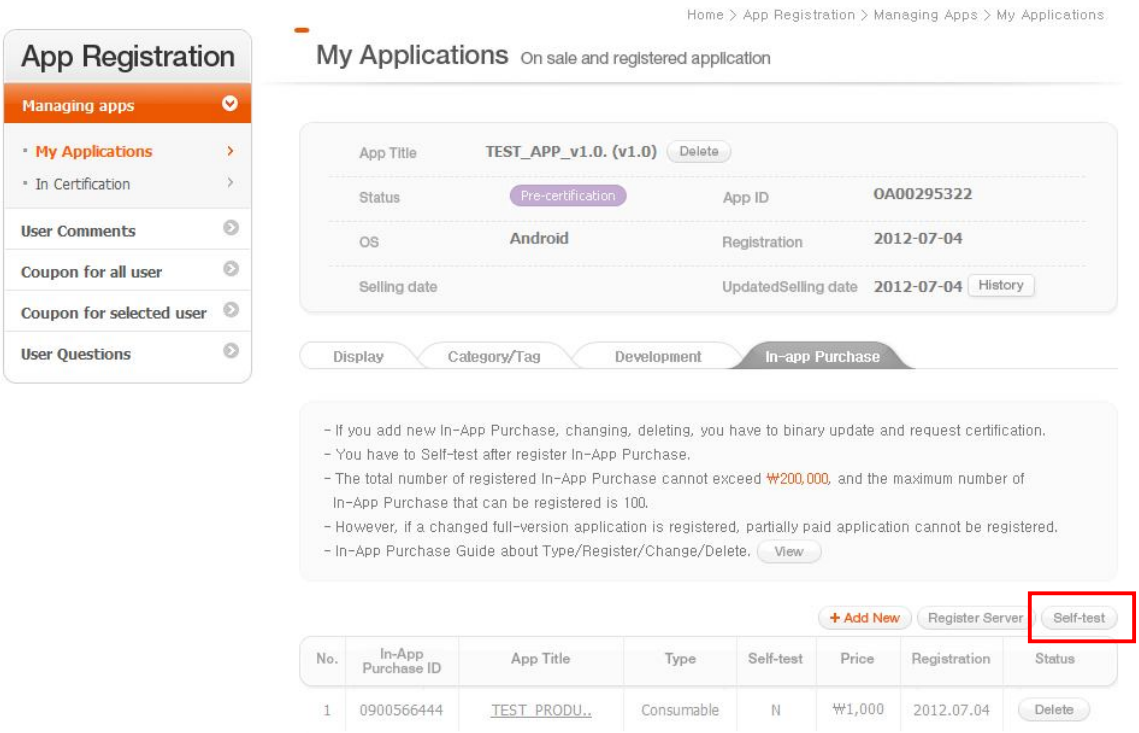
In [Product Registration/Management → Product Status] page of the application registered in T store Developers Center in Application development preparation stage, select [Development Information] tab to register APK file and detail information of the corresponding Application.



[Figure 9] Application APK Registration

■ Step 2) Start Self Test

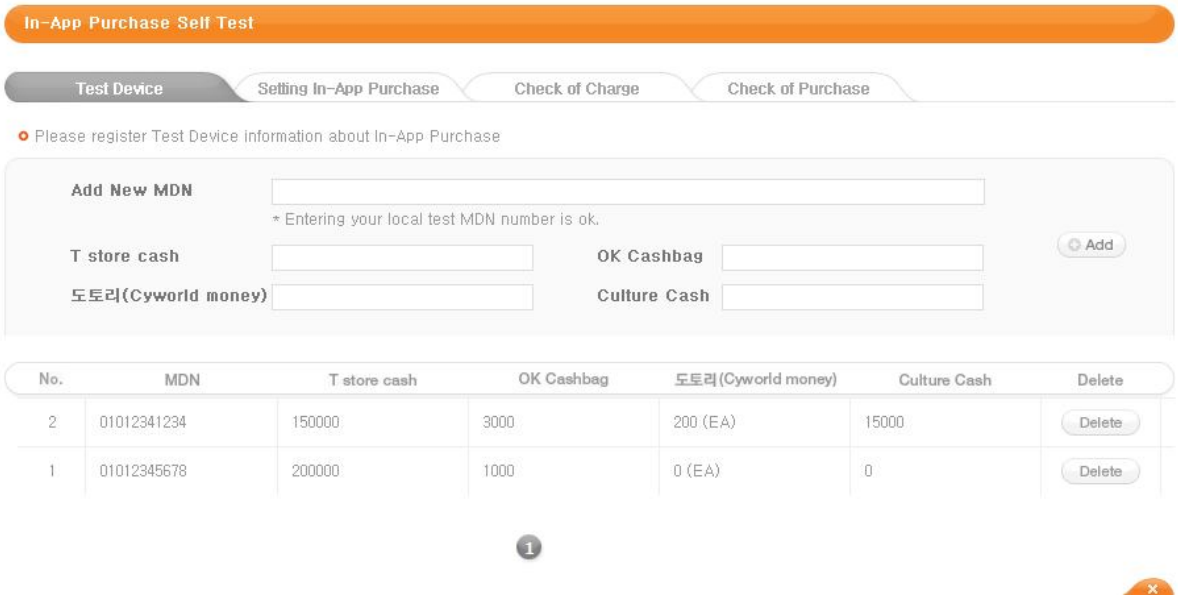
In [Product Status] page, select [In-app Purchase] tab and click [Perform Test] button.



[Figure 10] Carry Out Self Test

■ Step 3) Register Test Device

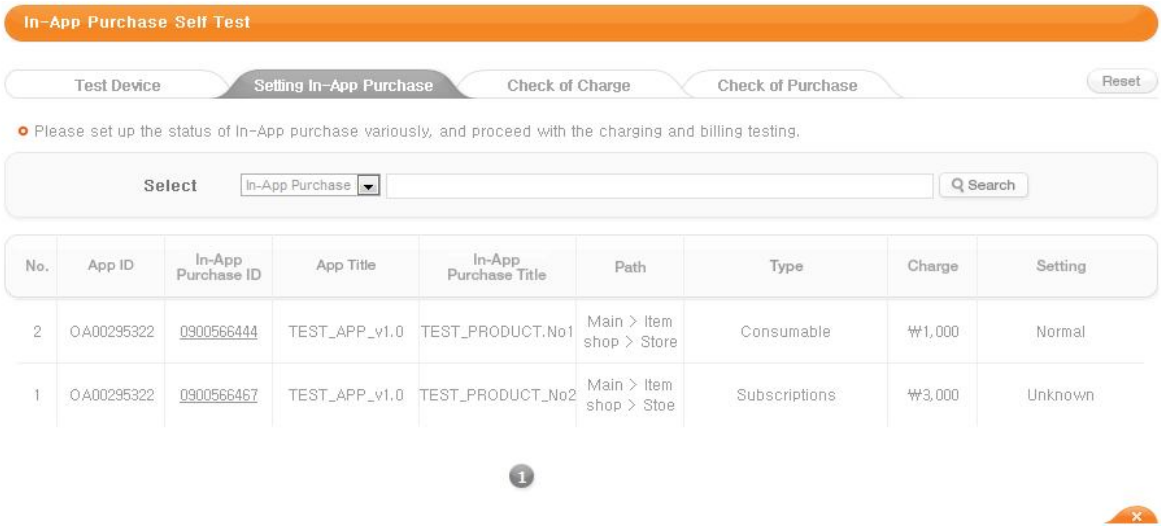
Same as in Step 2, select [In-app Purchase] tab and click [Perform Test] button, then the following Self Test Pop-Up window appears. In [Test Device] tab, register the MDN of the devices to be used in the test and T store Cash for the test.



[Figure 11] View Test Devices

■ Step 4) Test each In-App Product

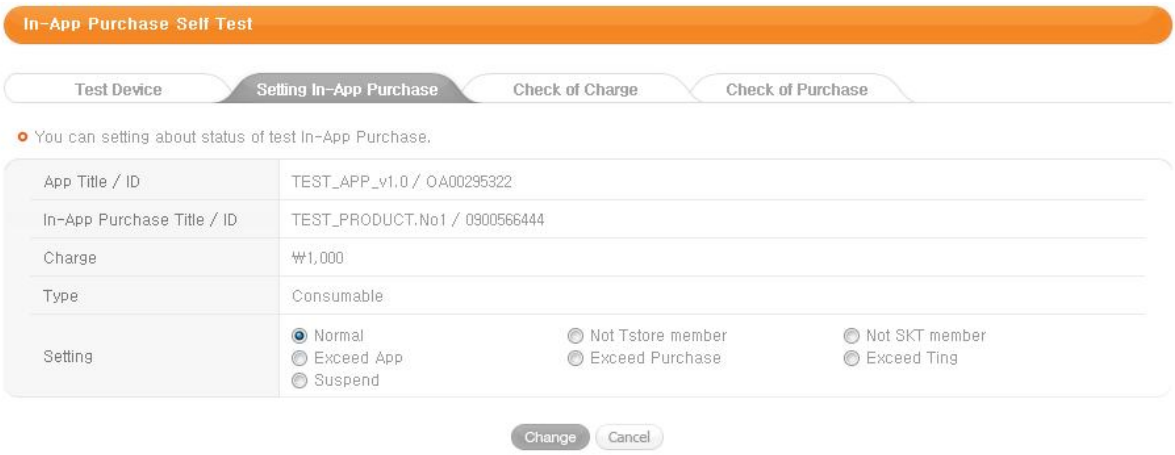
In [Setting for each Product] tab of Self Test Pop-Up window, In-App Product list of the application to test will be displayed as a table. Select any In-App Product to test from the list and move to the individual In-App Product test environment setting page.



[Figure 12] In-App product List

In In-App Product test environment setting page, various payment situations may be selected and tested. Select one of the values presented in [Result Setting] and click [Modify] button, then it will move to the previous In-App Product list screen.

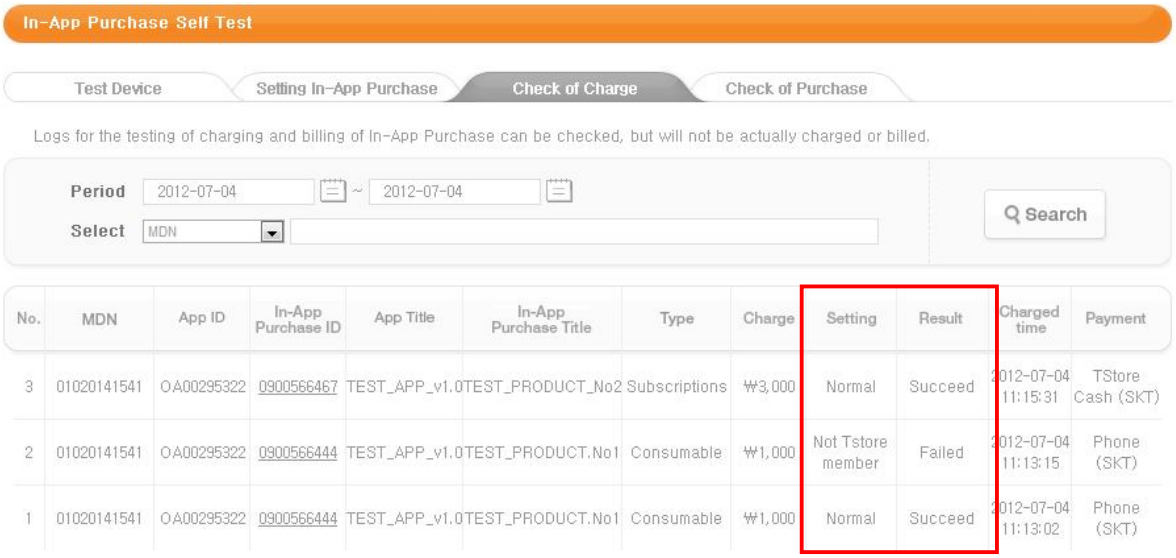
The charge shall be generated only when "Normal" is selected in [Result Setting], and selecting other values shall not generate the charge.



[Figure 13] Details of In-App product Test Environment Setting

■ Step 5) Charge Log Inquiry

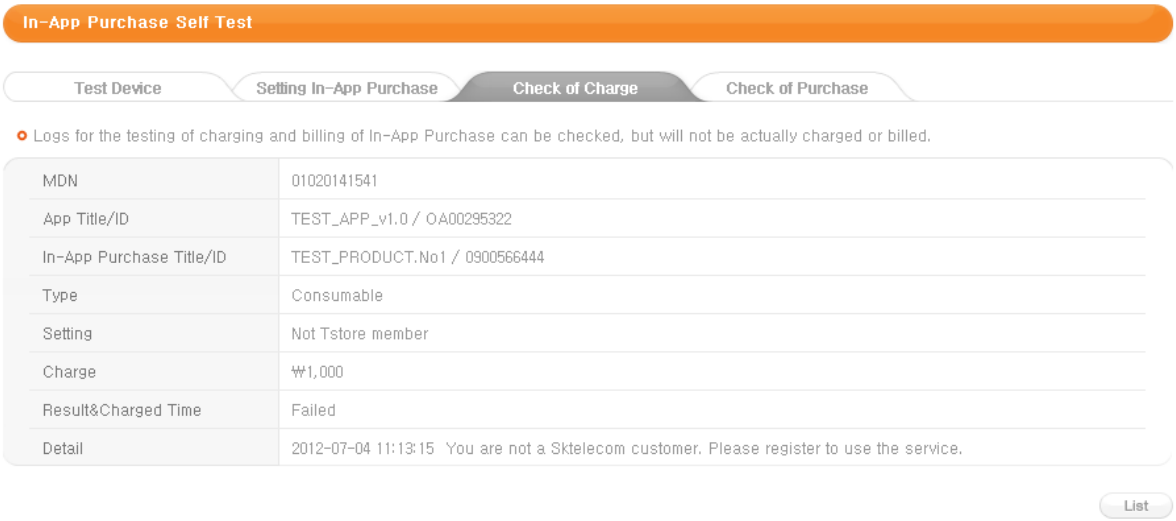
In [Charge Log Inquiry] tab of Self Test Pop-Up window, check the result tested according ot the test environment set in Step 4. In [Result Setting Information] column, "Result Setting" value set in Step 4 is displayed, and in [Charge Result] column, whether the charge is normal is displayed as "Succeeded" or "Failed".



[Figure 14] Charge Log Inquiry Screen

[Charge Result] is "Succeeded" only when [Result Setting Information] is "Normal", and for the remaining [Result Setting Information], the test shall be considered to be performed correctly only when all [Charge Result] are "Failed".

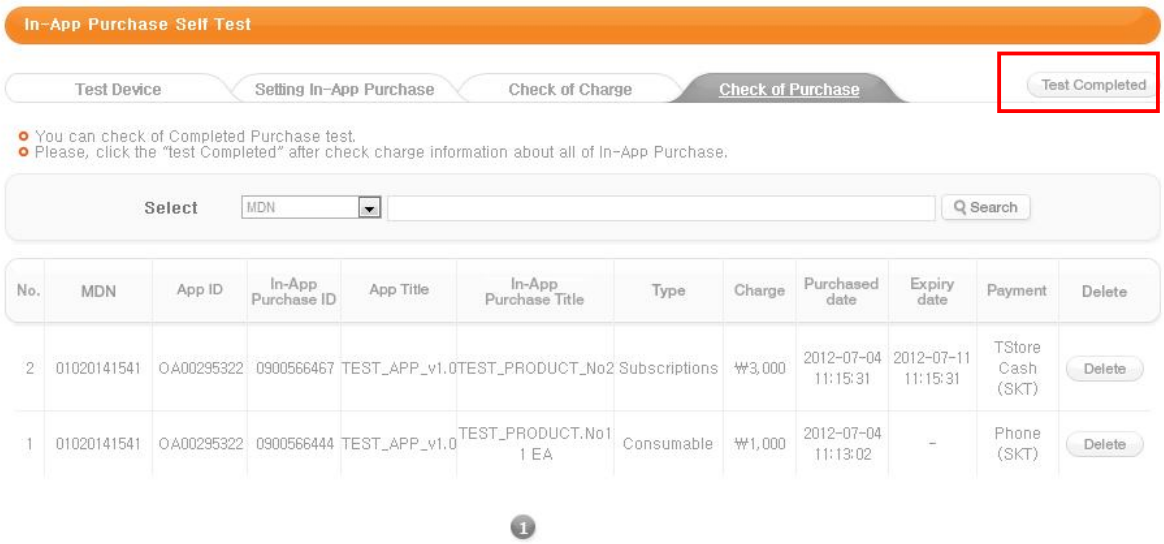
When a charge log with Charge Result of "Failed" is selected, the screen will be changed to the following Charge Log Details page, and you can see the explanations for the cause of the failure.



[Figure 15] Charge Log Inquiry Detail Screen

■ Step 6) Purchase History Inquiry

When normal purchase history is generated through the tests for all In-App Products included in the Application, you can end the Self Test, and request verification to T store. In [Purchase History Inquiry] tab of Self Test Pop-Up window, review the history of the purchase, and if the purchase histories of all In-App Products are not generated, click [Test Completed] button on the top right side of the screen to end the Self Test.



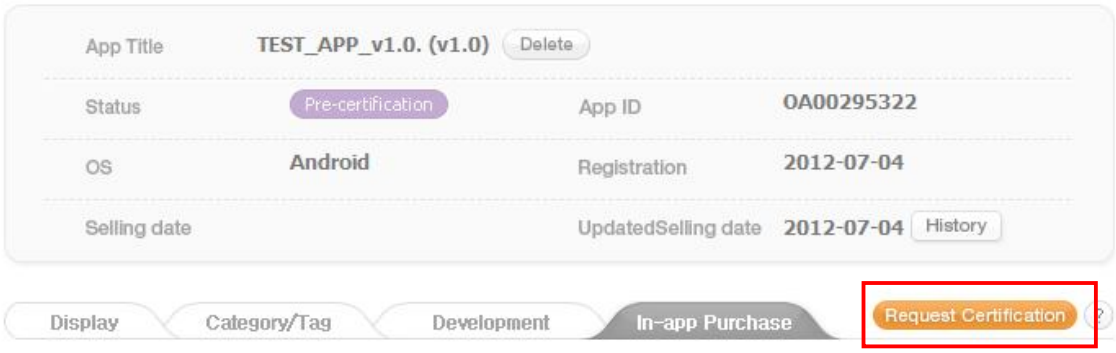
[Figure 16] Purchase History Inquiry

3.4. Application Certification

All applications can be sold only after completing the commercial verification by the application verifying personnel in T store. If Self Test is completed, developer shall replace IAP API (Lib/IAPLibD_Eng.jar) applied to the application with commercial IAP API (Lib/IAPLibR.jar) and recompile the application. Register the recompiled Application to T store Developers Center as a modification registration and request verification. Detail description on Application verification is stated in T store Developers Center [Usage Guide].

(http://dev.T store.co.kr/devpoc/guide/guideProd.omp#a1_1)

In T store Developers Center [Product Registration/Management → Product Status], find the application and request the verification of the corresponding application.

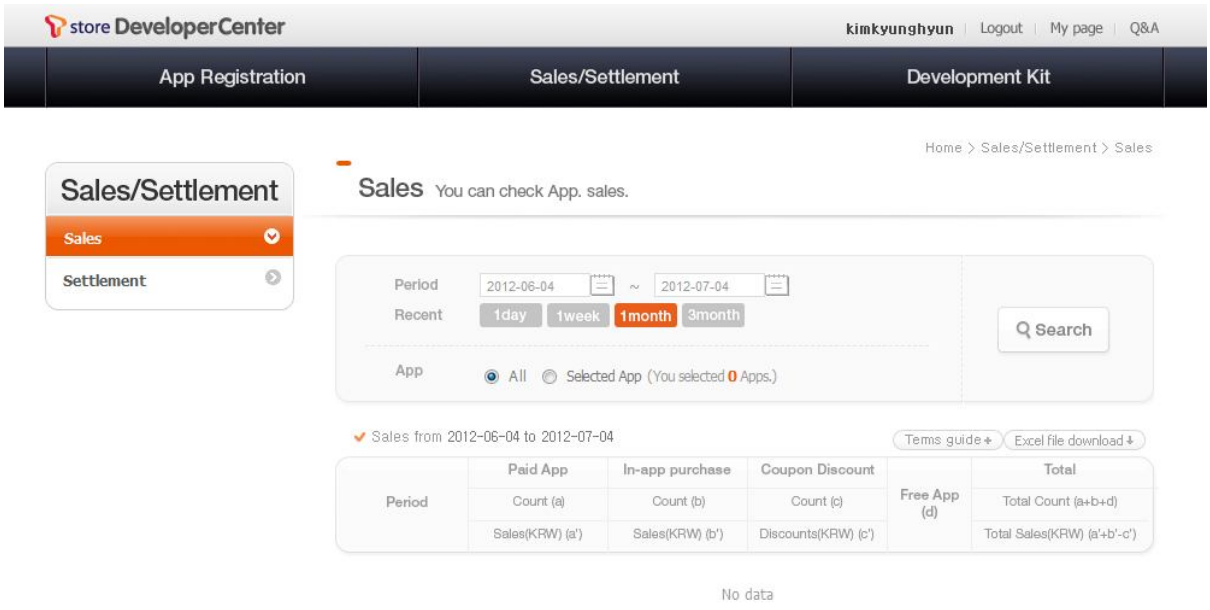


[Figure 17] Application Certification Request

3.5. Sales and Settlement

You can see that the status of the In-App product that completed the verification of the application verification personnel in T store is changed to [Standby for Sales] in T store Developers Center [Product Registration/Management → Product Status] page.

When the developer himself or T store operator changes the status from [Standby for Sales] to [On Sale], it can be distributed to T store for the sales. Daily/monthly sales status, settlement data, and tax invoice of the In-App Product that began the sales can be reviewed in T store Developers Center. Detail description on Application Sales/Settlement is stated in T store Developers Center [Usage Guide]. (http://dev.T store.co.kr/devpoc/guide/guideProd.omp#a1_1)



[Figure 18] Sales/Settlement Management

4. Application Programming Guide

This chapter describes how to use IAP API to develop T store In-App Purchases in the application.

It is generally developed in the order of library initialization, Result event listener setting, and API call.

■ Step 1) Library Initialization

Initialize library to use IAP API in the Application.

■ Step 2) Result Event Listener Setting

T store IAP uses asynchronous method that receives Request Result of API through Event Listener. Set the Event Listener to receive the Request Result of the called IAP API. Appropriate processing shall be defined and developed by the developer in the application according to the received result.

■ Step 3) Call API

After the initialization of the library, call In-App Product purchase related API (product purchase request, purchase product validity verification request, valid purchase product list request, extinctive product decrement request, monthly fixed price product automatic payment cancel request, etc.) according to its purpose. When this API is called, whether App user is member of T store is verified, and [Member Subscription Guide] popup is displayed to guide the member subscription for the non-members.

[Figure 19] Member Subscription Guide Popup

➢ MDN input Screen

- In-app billing payment provides MDN input Screen for foreign developer. Using this feature, phone number and social security number can be set.
- This feature is only available in foreign developer mode. (IAPLibD_Eng.jar)

[Figure 19-1] MDN input

4.1. API Details

It describes the individual API's provided by T store IAP.

API	Description
IAPLIBInit	IAP library initialization
popPurchaseDlg	In-App Product purchase request
sendItemAuth	Request to verify whether the purchased In-App Product can be used
sendItemWholeAuth	Request for the entire list of the products that can be used among the purchased In-App Products
sendItemUse	Request for remaining number after decrementing the number of the extinctive products
sendPurchaseDismiss	Request for the cancellation of monthly fixed price product automatic payment

[Table 2] T store IAP API Details List

4.1.1. IAPLibInit

■ Usage

It is an API to initialize the library to use IAP API in the Application, and it shall be called once

before calling other API's. AID issued at the registration of the application in T store must be input, and if Application Server is used, additionally input access information.

■ **Syntax**

```
IAPLibInit(IAPLibSetting setting);
```

■ **Parameters**

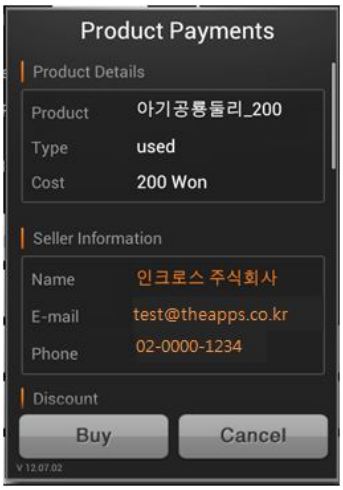
setting	NOT NULL	Environment setting class object
		
		public class IAPLibSetting {
		public String AppID = null;
		public String BP_IP = null;
		public int BP_Port = 0;
		public OnClientListener ClientListener = null;
		}

AppID	NOT NULL	Application ID
BP_IP		Application Server IP
BP_PORT		Application Server PORT
ClientListener	NOT NULL	Designate IAP API Call Result Event Listener

4.1.2. popPurchaseDlg

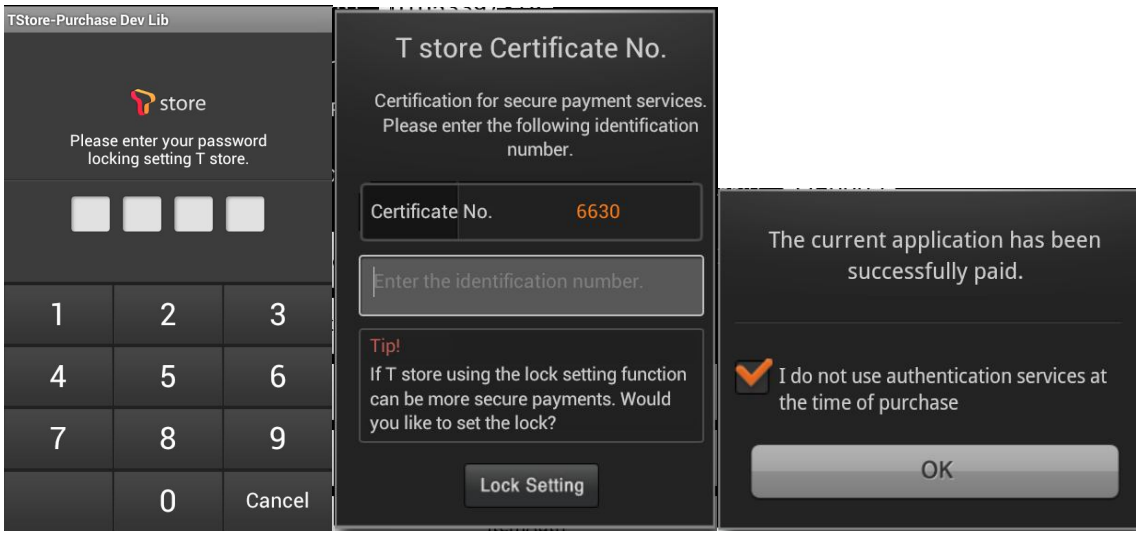
■ **Usage**

It is an API to request In-App Product purchase to IAP Server. When this API is called, [Product Payment] popup will be displayed. But, for the members of other mobile carriers, [Resident Number Input] popup is first displayed for resident number input for verification of the person, and then [Product Payment] popup will be displayed. Resident number input is performed only once for the first time.



[Figure 20] Product Payment Popup

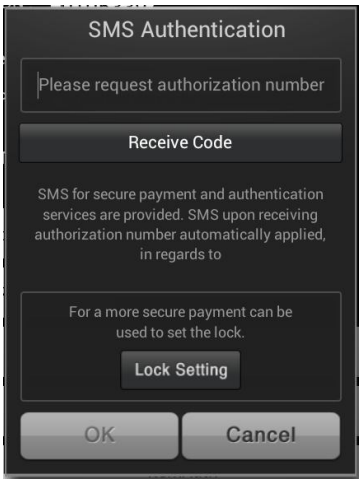
If "Purchase" button in [Product Payment] popup is pressed, to prevent payment by other person, it goes through the stages of T store lock password input or OTP certification input. If T store lock is set, [T store lock setting password input] window will be displayed, or if T store lock is not set, [OTP certification screen] window will be displayed.



[Figure 21] [Lock Setting Password] Popup/ [T store Certification Number] Popup/ Billing success the pop-up

If check "the box OTP accept unused" in purchase success window, it will not showing after checked. Add permission in AndroidManifest.xml because OTP accept unused is written to SD card. The method of reset OTP accept unused in SD card. : Delete "TstoreOTPLog.txt" file in terminal's USB storage.

When T store lock password input or OTP certification input is successfully completed, payment request of In-App Product is sent to IAP Server. But, for the members of other mobile carriers, the payment request is sent after additionally going through the following [SMS certification] stage.



[Figure 22] [SMS Certification] Popup

■ Syntax

popPurchaseDlg(String pID, String pName , String pTID, String pBPInfo)

☞ **Required value, except if you are not using their parameters must be enclosed with a null**

☞ 예) popPurchaseDlg("0000123456", null, null, null);

☞ **The following API is not used in the library distributed after 2012. 6. 27.**

popPurchaseDlg(String pID)
popPurchaseDlg(String pID, String pName)
popPurchaseDlg(String pID, String pName, String pTID)

■ Parameters

pID	NOT NULL	PID(Product ID) of In-App Product registered in T store
pName		It is the product name of In-App Product displayed as "product name" in [Product Payment] popup. Developer may input any name differently from the product name registered in T store.
pTID		It is an identifier that Application Server issues to IAP Server for each purchase case to make the inquiry of the history of the purchased In-App Product, and it shall not be duplicated in the Application. The input string shall be 100 bytes or less.
pBPInfo		Developers freely write additional information of In-App Product for the sales. This information is saved in IAP Server to be viewed later for purchase history, so it is mostly used to input the information to check temporary product price variation by event. The input string shall be 1024 bytes or less.

■ Listener

onItemQueryComplete()	It is called when "Confirm" is selected in [Product Payment] popup, and it can designate the actions before purchase request is sent to IAP Server.
onItemPurchaseComplete()	It is called when product purchase is completed.
onDlgPurchaseCancel()	It is called when "Cancel" is selected in [Product Payment] popup.
~~onJuminNumberDlgCancel()~~	~~It is called when "Cancel" is selected in [Resident Number input] popup.~~
onError()	It is called when error occurs, and it returns the error code.
onDlgError()	It is called when error occurs, and [Error Guide] popup is

	displayed.

4.1.3. sendItemAuth

■ **Usage**

Application user requests to verify whether the purchased In-App Product in the Application is valid. It is mostly used to verify expiration of the period for permanent / extinctive / periodical / monthly fixed price products.

■ **Syntax**

sendItemAuth(String pID)

■ **Parameters**

pID	NOT NULL	Product ID

■ **Listeners**

onItemAuthInfo(ItemAuthInfo itemAuth)	It is called when request for the verification of the purchased In-App Product is completed.
onError()	It is called when error occurs, and it returns the error code.
onDlgError()	It is called when error occurs, and [Error Guide] popup is displayed.

4.1.4. sendItemWholeAuth

■ **Usage**

Application user requests the valid product list among all of the purchased In-App Products in the application. As the same as sendItemAuth, it is mostly used to verify expiration of the period for permanent / extinctive / periodical / monthly fixed price products.

■ **Syntax**

sendItemWholeAuth()

■ **Parameters**

None

■ **Listener**

onWholeQuery(ItemAuth[] items)	It is called when valid product list request is completed.
onError()	It is called when error occurs, and it returns the error code.

onDlgError()	It is called when error occurs, and [Error Guide] popup is displayed.

4.1.5. sendItemUse

■ **Usage**

Application user requests to decrement the number of the purchased extinctive products, and requests the remaining number after the decrement.

■ **Syntax**

sendItemUse(String pID)

■ **Parameters**

pID	NOT NULL	Product ID

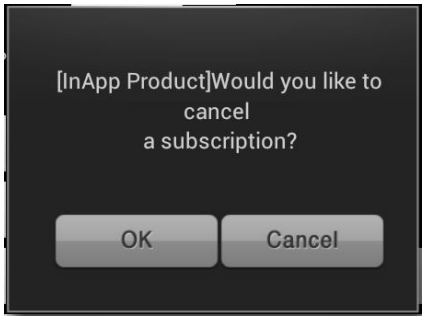
■ **Listeners**

onItemUseQuery(ItemUse item)	It is called when product decrement and remaining number request is completed.
onError()	It is called when error occurs, and it returns the error code.
onDlgError()	It is called when error occurs, and [Error Guide] popup is displayed.

4.1.6. sendPurchaseDismiss

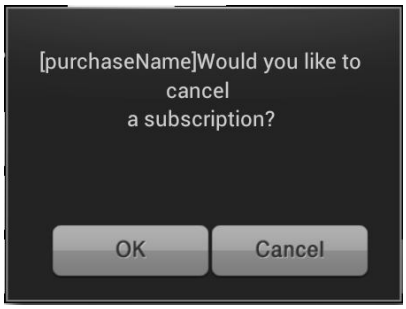
■ **Usage**

Application user requests the cancellation of the automatic payment of the purchased monthly fixed price product. [In-App Product Interim Cancel] popup will be displayed.



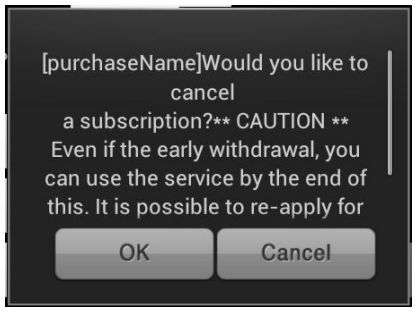
[Figure 23] [In-App Product Interim Cancel] Popup

Case 1) Upon termination within 24 hours after the initial purchase



[Figure 24-A] [In-App Product Interim Cancel] Popup

Case 2) After reentrance upon termination in a complete state cancellation within 24 hours after the initial purchase / After 24 hours upon termination



[Figure 25-B] [In-App Product Interim Cancel] Popup

■ Syntax

sendPurchaseDismiss (String pID, String pName)

■ Parameters

pID	NOT NULL	Product ID
pName	NOT NULL	It is the product name of In-App Product displayed as "product name" in [In-App Product Interim Cancel] popup. Developer may input any name different from the product name registered in T store.

■ Listeners

onPurchaseDismiss()	It is called when "Confirm" is selected in [In-App Product Interim Cancel] popup.
onDlgAutoPurchaseInfoCancel()	It is called when "Cancel" is selected in [In-App Product Interim Cancel] popup.
onError()	It is called when error occurs, and it returns the error code.

onDlgError()	It is called when error occurs, and [Error Guide] popup is displayed.

4.2. Result Event Listener

It describes the Event Listeners that receive request result of T store IAP API.

Listener	Description
onItemQueryComplete	Listener is called after popPurchaseDlg API call when purchase is possible, and if true is set as the return value in the listener, the final payment approval will be requested.
onItemPurchaseComplete	It is called when the final payment approval request is completed.
onWholeQuery	User returns the valid product list in the purchased product list.
onItemUseQuery	It returns the remaining number of products after the decrement.
onItemAuthInfo	User returns the result for certification request of the purchased product.
onError	It returns error code and message if error occurs during the use of API.
onDlgError	It is called when confirm button is clicked in error notice popup window.
onDlgPurchaseCancel	It is called when cancel button is clicked in payment UI popup window.
~~onJuminNumberDlgCancel~~	~~For the members of other mobile carriers' services, it is clicked when cancel button is clicked in resident number input window.~~
onJoinDialogCancel	It is called when cancel button is clicked in T store member subscription guide popup window.
onDlgAutoPurchaseInfoCancel	It is called when cancel button is clicked in monthly automatic payment product cancellation guide popup window.
onPurchaseDismiss	It is called when confirm button is clicked in monthly automatic payment product cancellation guide popup window.

[Table 3] T store IAP API Listener List

4.2.1. onItemQueryComplete

■ **Usage**

It is the Result Event Listener of popPurchaseDlg API, it is called when "Confirm" is selected in [Product Payment] popup to be able to add any functions before sending purchase request to IAP Server. It must be set to return true to send purchase request to IAP Server.

■ **Syntax**

```
public Boolean onItemQueryComplete()
```

■ **Response**

None

4.2.2. onItemPurchaseComplete

■ **Usage**

It is the Result Event Listener of popPurchaseDlg API, and it is called when product purchase request processing is completed.

■ **Syntax**

```
public void onItemPurchaseComplete()
```

■ **Response**

None

4.2.3. onItemAuthInfo

■ **Usage**

It is the Result Event Listener of sendItemAuth API, and user returns the result of the verification of whether the purchased product can be used as ItemAuthInfo.

■ **Syntax**

```
public void onItemItemAuthInfo(ItemAuthInfo itemAuth)
```

■ **Response**

itemAuth	NOT NULL	public class ItemAuthInfo {
		public int pCount;
		public byte[] pExpireDate = null;
		public byte[] pToken = null;
		}

pCount		Remaining number of products If it is a monthly fixed price product, remaining valid period (number of days)

pExpireDate		Valid period expiration date If it is a monthly fixed price product, monthly fixed price expiration date
pToken		Certification token of Streaming product

4.2.4. onWholeQuery

■ **Usage**

It is the Result Event Listener of sendItemWholeAuth API, and user returns the valid product list among the purchased product list as ItemAuth[].

■ **Syntax**

```
public void onItemPurchaseComplete (ItemAuth[] items)
```

■ **Response**

items	NOT NULL	public class ItemAuth {
		public String pID;
		public String pName;
		}

pID		Product ID
pName		Product Name

4.2.5. onItemUseQuery

■ **Usage**

It is the Result Event Listener of sendItemUse API, and the remaining number of the products after the decrement is returned as ItemUse.

■ **Syntax**

```
public void onItemUseQuery (ItemUse item)
```

■ **Response**

items	NOT NULL	public class ItemUse {
		public String pId;
		public String pName;
		public String pCount;
		}

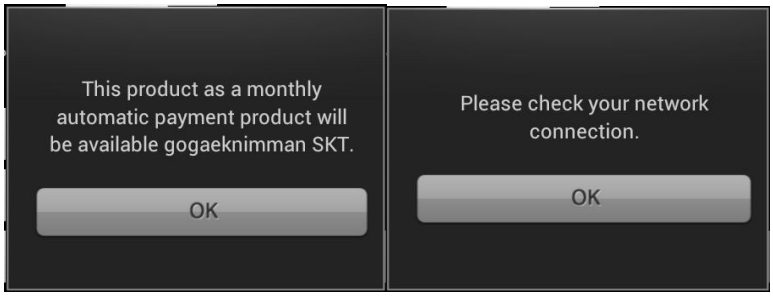
pID		Product ID
pName		Product Name
pCount		Remaining number of the products

		*If the remaining number is less than 0, it returns error code as onError

4.2.6. onError

■ **Usage**

It is the Event Listener called when error occurs during the use of IAP API, it returns error code, and displays [error guide] popup. Refer to "4.4 Error Message" for description of each error code.



[Figure 26] Error Guide Popup

■ **Syntax**

```
public void onError (int ErrorCode, int SubErrorCode)
```

■ **Response**

ErrorCode		Main error code
SubErrorCode		Detail error code

4.2.7. onDlgError

■ **Usage**

It is the Event Listener called when "Confirm" button is clicked in [error guide] popup.

■ **Syntax**

```
public void onDlgError ()
```

■ **Response**

None

4.2.8. onDlgPurchaseCancel

■ **Usage**

It is the Event Listener called when "Cancel" is selected in [Product Payment] popup.

■ **Syntax**

```
public void onDlgPurchaseCancel()
```

■ **Response**

None

4.2.9. ~~onJuminNumberDlgCancel~~

■ ~~**Usage**~~

~~It is the Event Listener called when "Cancel" is selected in [Resident Number input] popup.~~

■ ~~**Syntax**~~

```
~~public void onJuminNumberDlgCancel()~~
```

■ ~~**Response**~~

~~None~~

4.2.10. onJoinDialogCancel

■ **Usage**

It is the Event Listener called when "Cancel" is selected in [Member Subscription guide] popup.

■ **Syntax**

```
public void onJoinDialogCancel()
```

■ **Response**

None

4.2.11. onDlgAutoPurchaseInfoCancel

■ **Usage**

It is the Event Listener called when "Cancel" is selected in [In-App Product Interim Cancel] popup.

■ **Syntax**

```
public void onDlgAutoPurchaseInfoCancel()
```

■ **Response**

None

4.2.12. onPurchaseDismiss

■ **Usage**

It is the Event Listener called when "Confirm" is selected in [In-App Product Interim Cancel] popup.

■ **Syntax**

```
public void onPurchaseDismiss()
```

■ Response

None

4.3. Sample Code

This sample code is the description of In-App product purchase request among the sample projects included in the package distributed as IAP API guide.

```
// IAP API Class import declaration
package com.feelingk.test;
import com.feelingk.iap.IAPActivity;
import com.feelingk.iap.IAPLib;
import com.feelingk.iap.IAPLib.OnClientListener;
import com.feelingk.iap.IAPLibSetting;
import com.feelingk.iap.net.ItemAuth;
import com.feelingk.iap.net.ItemAuthInfo;
import com.feelingk.iap.net.ItemUse;

public class sample extends IAPActivity {
    // Sample Data for Built-In Model
    String AppID = "OA00095725";
    String PID   = "0900075541";
    String BP_IP  = null;
    int BP_Port   = 0;

@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        // Generates IAP setting object
        IAPLibSetting setting   = new IAPLibSetting();
        setting.AppID           = AppID;     // AID setting
        setting.BP_IP           = BP_IP;     // BP server IP
        setting.BP_Port         = BP_Port;   // BP server Port
        setting.ClientListener  = mClientListener;  //Result Listener
        // Initializes IAP library
        try{
          IAPLibInit(setting);
        }catch(Exception e) {
          e.printStackTrace();
        }
```

```
// Sample Code: Button Object setting
Button btnShowDlg = (Button) findViewById(R.id.Button01);
Button btnItemAuth = (Button) findViewById(R.id.Button03);
Button btnWholeAuth = (Button) findViewById(R.id.Button04);
Button btnItemUse = (Button) findViewById(R.id.Button05);

btnShowDlg.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {

        // IAP API Call: In-App product purchase request)
        popPurchaseDlg(PID,"In-App Product","20124568","Permanent item1");
    }
});

btnItemAuth.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // IAP API Call: Request for the verification of whether purchased In-App Product can be used
        sendItemAuth(PID);
    }
});

btnWholeAuth.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // IAP API Call: request for entire valid purchased In-App Product list
        sendItemWholeAuth();
    }
});

btnItemUse.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        // IAP API Call: request for decrement and remaining number of the purchased extinctive products
        sendItemUse(PID);
    }
});

OnClickListener mOCLPurchaseDismiss = new OnClickListener() {
    public void onClick(View v) {
```

```
                // IAP API Call: request for the cancellation of the purchased monthly
fixed price product automatic payment
                sendPurchaseDismiss(PID, PNAME);
            }
        };


        // Event Listener object definition
        OnClientListener mClientListener = new OnClientListener() {

            // IAP API: product purchase verification event processing
            public Boolean onItemQueryComplete() {
                return true;
            }

            // IAP API: product purchase completion event processing
            public void onItemPurchaseComplete() {
                // developed by development company
            }

            // IAP API: valid purchase product list reception event processing
            // Example of displaying the received product list as Toast popup
            public void onWholeQuery(ItemAuth[] items) {
                String strOut = "";
                int n = items.length;
                for(int i=0; i<n; ++i)
                {
                    strOut += items[i].pId + " : " + items[i].pName + "₩n";
                }
                ShowToast(getApplicationContext(), strOut);
            }

            // IAP API: Number decrement and remaining number of extinctive product
            reception event processing
            // Example of displaying the decrement result of the product as Toast Pop-Up
            public void onItemUseQuery(ItemUse item) {
                String strOut = "";
                strOut += item.pId + " : " + item.pName + " : " + item.pCount;
                ShowToast(getApplicationContext(), strOut);
```

```
}

// IAP API: Usage validity verification result event processing of the designated product
// Example of showing the verification result as Debugging Log
public void onItemAuthInfo(ItemAuthInfo itemAuth) {

    String strOut = "";
    strOut += PID + " : " + itemAuth.pCount + " : " + new String(itemAuth.pExpireDate);
    ShowToast(getApplicationContext(), strOut);

    if(itemAuth.pToken != null)
        Log.i("Sample", new String(itemAuth.pToken);

}

// IAP API: Process error result event of the called IAP API
// Example of showing the error results as Debugging Log according to the error code
@Override
public void onError(int arg0, int arg1) {
    // TODO Auto-generated method stub
    switch (arg0) {
        // initialization failed
        case IAPLib.HND_ERR_INIT:
            break;
        // certification processing error
        case IAPLib.HND_ERR_AUTH:
            break;
        // item purchase possible processing error
        case IAPLib.HND_ERR_ITEMQUERY:
            break;
        // item information reception error
        case IAPLib.HND_ERR_ITEMINFO:
            break;
        // item charge processing error
        case IAPLib.HND_ERR_ITEMPURCHASE:
            break;
```

```
            // server data processing error
            case IAPLib.HND_ERR_DATA:
                break;
            }
}

// IAP API: Process “Confirm” selection event in [error guide] popup
@Override
public void onDlgError() {
    // TODO Auto-generated method stub
}
// IAP API: Process “Cancel” selection event in [Product Payment] popup
@Override
public void onDlgPurchaseCancel() {
            // TODO Auto-generated method stub
}
// IAP API: Process “Cancel” selection event in [Resident Number input] popup
@Override
public void onJuminNumberDlgCancel() {
    // TODO Auto-generated method stub
}
// IAP API: Process “Cancel” selection event in [Member Subscription guide] popup
@Override
public void onJoinDialogCancel() {
    // TODO Auto-generated method stub
}
// IAP API: Process “Cancel” selection event in [In-App Product Interim Cancel] popup
@Override
public void onDlgAutoPurchaseInfoCancel() {
    // TODO Auto-generated method stub
}
// IAP API: Process “Confirm” selection event in [In-App Product Interim Cancel] popup
@Override
public void onPurchaseDismiss() {
    // TODO Auto-generated method stub
```

```
                }
            }
        };

        // Demonstration Toast popup
        public void ShowToast(Context context, String str) {
            Toast toast = Toast.makeText(context, str, Toast.LENGTH_LONG);
            toast.show();
        }
}
```

4.4. Error Message

It describes the error codes sent as a parameter to onError Listener called when an error occurs. Error message is displayed in [error guide] popup.

NAME	Status	Value	Code	Message	note
IAPLib.HND_ERR_INIT	Occur during the initialization fails	1999	-	Initialization failed	
IAPLib. HND_ERR_AUTH	An authenticati on failure occurs	2000	1	Session authentication failed.(No MDN requested)	
			1	Session authentication failed.((Unregistered server authentication failed due to server IP)	
			1	Application of information does not exist.	
			2	Your not a member of the SKtelecom. Sign up.	prima ry
			2	POC testing for the developer handset is not registered. Sign up..	Devel opme nt
			9	You are using prepaid mobile phone(PPS) usage fees payment is not possible.	
			-3	Is a connection error when	

				authenticating with the authentication server. lease check the network status.	
			11	Checking the system. After a moment, please use.	
IAPLib. HND_ERR_ITEMINFO	Product information acquisition failure occurs	2001	0	Sum payment to the funds usage fees will be charged the following month.	
			2	Your not a member of the SKtelecom. Sign up.	
			4	Period has expired. If you wish to continue using materials should be purchased.	
			7	Your trial version of the current use of this content is. Do you want to switch to the final version?	
			8	Its product is the automatic payment products. This monthly product auto-billing customers who SKT product is available.	
			9	Period has expired. If you wish to continue to repurchase is required by.	
			11	The goods for sale has been terminated.	
IAPLib. HND_ERR_ITEMQUERY	Product Availability failure occurs acquisition	2002	1	Your use of the content of this fee is required for T store Register. Do you want to proceed?	
			2	Your not a member of the SKtelecom. Sign up.	
			3	① Your state is the terminal stop.	Terminal state

					check
				② Your purchase is a blocking state.	Possible state payment
				③ Your normal is not a member of the T-Store	Members Check
				④T-store membership status lookup error occurred during. Please try again later.	Members of the error checking
				⑤ Your terminal information is not valid.(UACD Inconsistency)	UACD Check
				⑥ T-Store terminal information error occurred during lookup. Please try again later.	
				⑦ Your terminal information is not valid.(IMEI Inconsistency)	IMEI Check
				⑧ICAS Member information lookup failed.	Check for information on membership
			4	Include partial pay to purchase the current version can not be. Please use the version after the	

				upgrade.	
			5	Network transmission error.	
			6	By monthly limit has been exceeded the amount of 500,000 won. Please visit next month.	
			7	Limit the amount of 500,000 won per product use has been exceeded. Please visit next month.	
			8	According to plan and use limit has been exceeded. For more inquiries please contact the Customer Service Center.	
			9	The goods for sale has been terminated.	
			11	Checking the system. Please try again later.	
			12	Is the verification that goods.	
			15	① You currently are under permanent per-use products. Please use the buy back after the expiration of.	Permanent Not repurchase products
				② Your current period product use are under. Please re-purchased used after the expiration of. .	Check-Term Product
				③ You currently are under permanent per-use products . Please use the period after the expiration	Permanent item

				date of re-purchasing	
				④ Your current period product use are under. Please use the period after the expiration date of re-purchasing.	Term Product
				⑤ The final version of your product purchases there is a transition.	Today the final version switch
				⑥ Whether to buy the final version has an error occurs during query. Please try again later.	
				⑦ The monthly automatic payment product SKT customer its product is available	Monthly Product
				⑧ Your automatic payment products currently are being used. Please re-purchased used after the expiration of.	
				⑨ Your automatic payment products currently are be in Termination of the product by guests in your product based on repurchase of the month following termination of the current month is available from used. Please re-purchased used after the expiration of.	

				⑩ Termination of the product by guests in your product based on repurchase of the termination of the current month is available from the following month. purchase goods based on the information for authentication of the message listed in the appropriate message will appear.	Monthly Product
			16	Product testing is impossible not set. -> Part of the state value(setting) of goods will occur if this setting is not..	
			17	Termination products purchased monthly automatic payments have been completed.	Monthly Product
			18	Your monthly auto-payment products currently are being used. Please re-purchased after the expiration date.	Monthly Product
			19	Threesome monthly automatic payments using your product or service by the expiration of the period that will be notified. Thank you for using.	Monthly Product
			20	Its product is not for sale items is not payment.	

			21	Sales during status query error occurred. Please try again later.	
IAPLib. HND_ERR_ITEMPURCHASE	Failure occurs when purchasing goods	2003	0	Select the payment for goods have been processed successfully.	
			3	T-Store terminal information error occurred during lookup. Please try again later.	
			9	The goods for sale has been terminated.	
				T Cash Error Message	
				Out of the required parameters.	
				An undefined request.	
				No member information.	
				T store Cash Balance is low.	
				Deducted when T store Cash system error has occurred.	
				There is not enough T store Cash Balances	
			11	Checking the system. After a moment, please use.	
			12	According to plan and use limit has been exceeded. For more inquiries, please contact the Customer Service Center.	
			14	Check the limit for the purchase of goods is not	

				preceded by a.	
			15	In case of Danal peristalsis error.	
			-2	Wi-Fi <-> 3G did not change the network connection disconnects are smoothly.	
			-4	Brother works if the server, the server is a connection failure error.	
			-5	Due to network problems listening socket fails to respond is.	
IAPLib. HND_ERR_ITEMPURCHASE	Failure occurs when purchasing goods	2003	50	Billing month limit has been exceeded.	
			405	Please check the information you have entered	
			406	Social Security numbers do not match	
			410	Payments have been exceeded	
			411	Stop using the registered phone number.	
			412	Phone number is revoked	
			413	Mobile phones are not available for Corporate Clients	
			414	Communication can not be used as payment of fees	
			415	Payments have been exceeded	
			416	Please check the information you have	

				entered	
			419	Temporarily disable the phone is registered	
			421	Payments have been exceeded	
			422	Payment is limited small cell phone	
			423	Please check the information you have entered	
			424	Limit has been exceeded, per payment	
			432	There is a history of delinquency	
			434	Payment limit is not enough	
			435	Billing month limit has been exceeded.	
			436	Payment is limited small cell phone	
			450	Please check the information you have entered	
			455	Minors will not be able to use the service, subscribers	
			468	Billing month limit has been exceeded.	
			469	Payment is limited small cell phone	
IAPLib. HND_ERR_WHOLEQUERY	Occur during the entire product certification	2004	0	Item, the entire authentication information lookup has succeeded.	
			9	Item does not exist.	

	Product certification Occur during the entire		11	Checking the system. Please visit later.	
IAPLib. HND_ERR_USEQUERY	Items deducted failure occurs	2006	9	The remaining number of goods is zero.	
			11	Checking the system. Please visit later.	
IAPLib. HND_ERR_ITEMAUTH	Authenticati on failed items	2007	0	Items for authentication was successful.	
			2	You not a member of the SKtelecom. Sign up	
			4	Period has expired. If you wish to continue to repurchase is required by.	
				There is no item purchases.	
				Expiration, and authentication information already exists, the remaining products are zero if the number of messages will occur. Expiration, and authentication information already exists, the remaining products are zero if the number of messages will occur.	
IAPLib. HND_ERR_NORAMALTIMEOUT	Server Not Responding	2008	-	Network Error	
IAPLib. HND_ERR_PAYMENTIMEOUT	Server Not Responding	2009	-	Network Error	
IAPLib. HND_ERR_SERVERTIMEOUT	Server Not Responding	2010	-	Network Error	

IAPLib. HND_ERR_MEMBERSHIP	Failure occurs Register	2011	0	Register T-store succeeded.	
			9	Register T-store fails	
			11	Checking the system. Please visit later	
IAPLib. HND_ERR_OCBCARDUPDATE	OCB Card update error.	2018	-	Failed to change the OK Cashbag card.	
IAPLib. HND_ERR_CULTURELANDCASHINFO	Culture-Land Cash inquiry error.	2020	-	Failed to lookup the Culture-Land.	
IAPLib. HND_ERR_PURCHASEDISSMISS	Billing occurs automatically terminate upon failure	2012	0	Termination of the automatic payment has been successful in value-added services.	
				Normal termination of a monthly automatic payment has been processed product will be available until the end of the current month. Thank you.	
				Normal termination of a monthly automatic payment has been processed goods. Thank you.-> Revocation in accordance with the appropriate message is output	
			4	There is no automatic billing purchases	
			9	Additional services failed to terminate the automatic payment	

				Monthly automatic payments failed to terminate purchases.	
				Automatic monthly payments on purchases revocation information transmission failed.	
				Out of the required parameters.	
				Termination processing was already.	
● library itself, a pop-up message					
Check acceptance occurs when personal information			0	Did not agree to the collection of personal information.	
Check your personal information upon request consent occurs			0	Collection of personal information has been stored for acceptance.	
			1	Already agreed to collect personal information.	
			9	Failed to agree personal information collection. Please contact your administrator	
SMS authentication occurs when customers LGU			0	①SMS authentication successed	
				② SMS authorization number successful transmission	
			9	The goods for sale has been terminated.	
			10	LGU + customers are not	

5. IAP Server Interface API Guide

IAP Server provides API for the inquiry of the In-App purchase history and real time purchase certification for IAP development in the form of Server Programming Model. Application Server makes the requests in HTTP GET method, and IAP Server responds in XML Over HTTP method, and no separate library is provided.

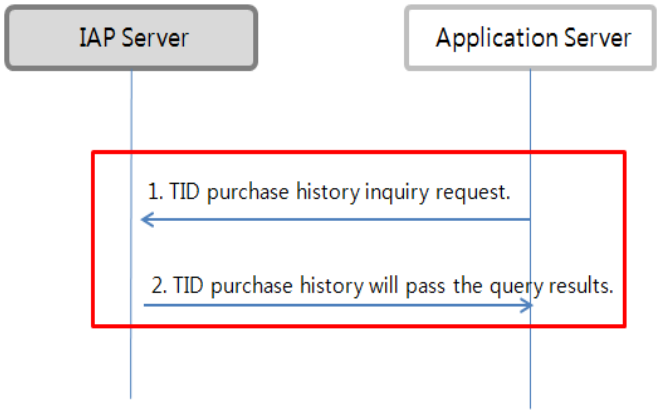
The interface specification with IAP Server provides the following 2 types of interfaces.

- Purchase History Inquiry API
  - It is an API for the inquiry of In-App Product purchase history. It shall not be used for real time purchase history inquiry. If it is found to be used a real time inquiry, T store operation team may take necessary actions. (Payment block, IP based access block, etc.)
- Purchase History Real Time Inquiry API
  - It is the interface specification to verify the purchase of In-App product in real time for an application with Download type Server Programming Model. To use this API, the application shall be registered T store Developers Center and request encryption/decryption module and KEY to T store operation team. (T store operation team: ompadmin@incross.com)
  - E-mail to be delivered upon request are as follows.
    ① Developer Center ID of the requester T store
    ② Contact (name, e-mail address, telephone number)
    ③ The product information below

Product name	AID

5.1. Purchase History Inquiry API

TID(Transaction ID) generated in the application is used as an identifier required for In-App Product purchase history inquiry.



[Figure 27] Purchase History Inquiry Procedure

• TID Purchase History Inquiry Request

- Request URL (development verification IAP Server)

http://211.234.231.208:8082/billIntf/billinglog/billloginquiry.action

- Request URL (commercial IAP Server)

http://211.234.231.209:8090/billIntf/billinglog/billloginquiry.action

- request URL Parameter

Parameter	Description
DATE	Product purchase date (ex)20101130)
APPID	Application ID
TIDCNT	Number of TID's to verify Payment Information. Maximum of 20 cases may be requested.
TID	Transaction ID issued by T store IAP API when a user requests In-App product purchase N TID's may be requested according to the number of TIDCNT, and it is classified with '|'.

- Example(request to development verification IAP Server)

http://211.234.231.208:8082/billIntf/billinglog/billloginquiry.action?DATE=20101130&APPID=OA00027256&TIDCNT=2&TID=12313|12324

- Example(request to commercial IAP Server)

http://211.234.231.209:8090/billIntf/billinglog/billloginquiry.action?DATE=20101130&APPID=OA00027256&TIDCNT=2&TID=12313|12324

• TID Purchase History Inquiry Response

- Response Message

Tag	Length (Byte)	Description	Value
type		Request identifier	Fixed as "BillingLog"
result	-	Payment Information Group Tag	
status	4	Payment Information Inquiry Result code	<status>0</status> 0: Success 9: There is no Inquiry Result or parameter or system error
detail	4	Payment Information Inquiry Result Detail	<detail>0000</detail> 0000: Success

		code	1000: Mandatory parameter error 2000: Undefined request 3000: Number of requests error 9100: No Payment Information Inquiry Result 9200: Exceeded the maximum value(20) of the number of the requests 9999: System error
message	-	Payment Information Inquiry Result Message	<message>The inquiry is made normally.</message>
appid	10	Applet ID	<appid>OA00027256</appid>
count	2	Verified number of the Payment Information	<count>20</count>
billing_log	-	It is a group of individual payment history information verified in IAP Server, and it includes multiple <Item> elements.	<billing_log> <item> ... </item> <item> ... </item> ... </billing_log>
Item	-	Individual payment category provided to Billing log, and it has the following lower level elements to explain the details of the payments.	<item> <tid>...</tid> <product_id>...</product_id> ... </item>
tid	< 100	Transaction ID	<tid>201012226_01047637315_0000023942</tid>
product_id	14	T store In-App Purchase product ID	<product_id>0000044056</product_id>
log_time	10	T store In-App Purchase product purchase time	<log_time>20101227103643</log_time>
charging_id	11	User MDN	<charging_id>01047637315</charging_id

		corresponding to TID	>
charge_amount	7	Product price	<charge_amount>500</charge_amount>
detail_pname	256	Detail product information	<detail_pname>Detail product information</detail_pname>
bp_info	1024	Information to be verified by the application server of the developer	<bp_info> data to be transferred by bp </bp_info>
tcash_flag	1	Whether T store Cash is used	<tcash_flag>N</tcash_flag>

- Example(when the Result is a success)

```
<?xml version="1.0" encoding="euc-kr" ?>
<GXG_RES type="BillingLog">
  <result>
   <status>0</status>
   <detail>0000</detail>
   <message>The inquiry is made normally.</message>
   <appid>0000023943</appid>
   <count>3</count>
 <billing_log>
    <item>
        <tid>201012226_01047637315_0000023941</tid>
        <product_id>0000044056</product_id>
        <log_time>20101227102658</log_time>
        <charging_id>01047637315</charging_id>
        <charge_amount>500</charge_amount>
        <detail_pname>DetailInfoTest....</detail_pname>
        <bp_info>null</bp_info>
        <tcash_flag>N</tcash_flag>
    </item>
```

```
        ...
    </billing_log>
 </result>
</GXG_RES>
```

- Example(when the Result is a failure)

```
<?xml version="1.0" encoding="euc-kr" ?>
<GXG_RES type="BillingLog>
 <result>
    <status>9</status>
    <detail>9999</detail>
    <message>System is under inspection.  Please try again later.</message>
    <appid>OA00027256</appid>
    <count>0</count>
 </result>
 </GXG_RES>
```

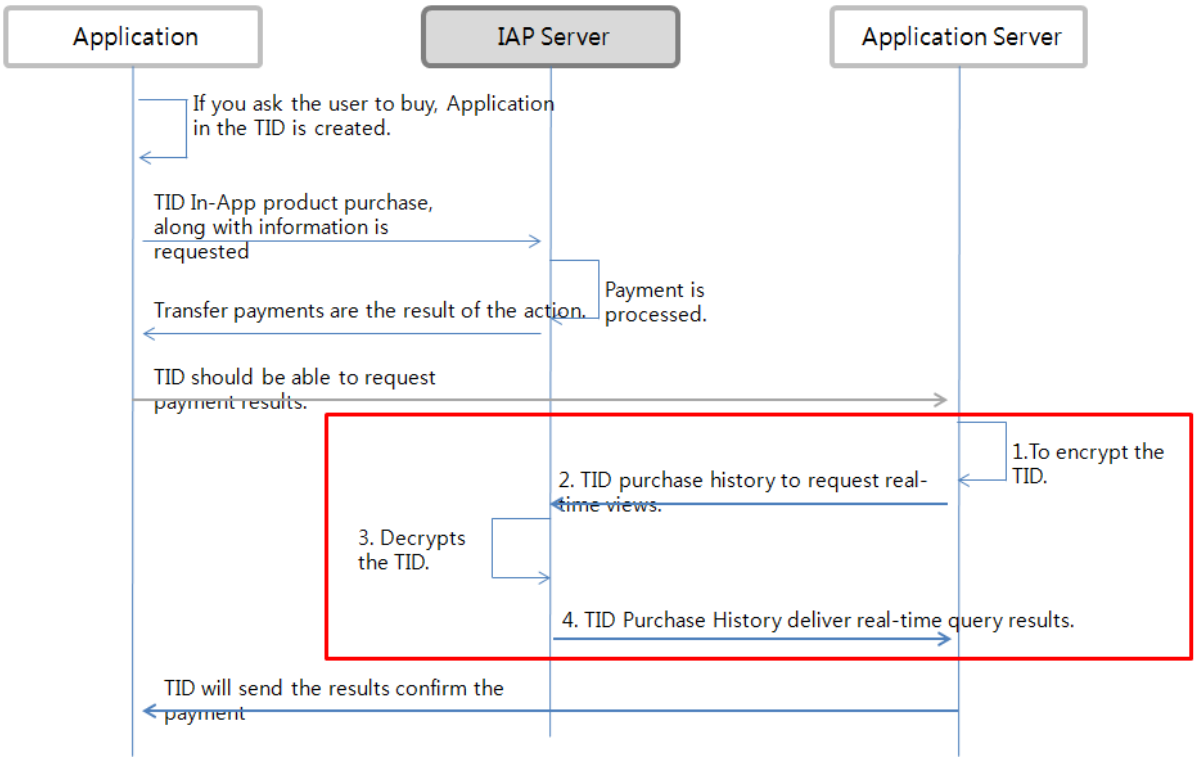
• Caution

- TID Purchase History Inquiry API is for a simple purchase history inquiry, and it cannot be called and used in the Application.

5.2. Purchase History Real Time Inquiry API

Same as Purchase History Inquiry API, this API also uses TID(Transaction ID) generated by T store IAP API included in the application as the identifier required for In-App Product Purchase History Inquiry. Since it is for the real time inquiry, the request shall be made and the response shall be received within 5 minutes from the completion of the purchase, and the inquiry of the data before 5 minutes from the current time is not possible.

All data exchanged between Application Server and IAP Server are encrypted with AES128 encryption algorithm, one password KEY is issued for each application, and key may not be changed after being sold to the user.



[Figure 28] Purchase History Real Time Inquiry Procedure

• TID Purchase Real Time Inquiry Request

- development device interface URL

http://211.234.231.208:8082/billIntf/billinglog/billlogconfirm.action?TID=xxxxx&APPID=xxxx

- commercial device interface URL

http://211.234.231.209:8090/billIntf/billinglog/billlogconfirm.action?TID=xx&APPID=xxx

- request URL Parameter

Parameter	Description	Mandatory	Description
TID	Transaction ID	Mandatory	Password module applied
APPID	Application ID	Mandatory	Not applied

• TID Purchase History Real Time Inquiry Response

- If the value of Result XML status is 0, it is a success, and other value is a failure.
- Response Message

※ Mandatory: M(Mandatory), O(Optional) – whether the value of the corresponding field is mandatory

※ Password applied: Y(YES), N(NO) – whether a password shall be applied to the corresponding field

Tag	Description	Application of password	Mandatory
type	"AuthToken" fixed value	N	M
result		Payment Information Group Tag	M
status	Inquiry Result status code 0: Success 9: Failure	N	M
detail	Inquiry Result detail code **0000: Normal** 1000: Mandatory parameter is not sufficient (code generated when one of TID or APPID is missing during the Inquiry request) 1001: APPID is Not valid. 1002: Unregistered APPID. 9100: No Purchase History (when there is no history of purchasing with TID) 9110: No Password KEY (when there is no password key registered for AID) 9111: Invalid password key (when password key that is different from the specification is input in DB) 9112: Password key not matching (when password key and the encryption key do not match) 9999 : SYSTEM ERROR (system error is an internal error in IAP,	N	M

	contact operation team)		
message	Payment Information Inquiry Result Message	N	M
appid	Application ID	N	M
billing_log	Real time payment history information verified in IAP Server, and it includes one <Item> element.	N	M
Item	Individual payment category provided to Billing log, and it has the following lower level elements to explain the details of the payments.	N	M
tid	Password applied Transaction ID	Y	M
product_id	T store In-App Purchase product ID	N	M
log_time	T store In-App Purchase product purchase time	N	M
charge_amount	Product price	N	M
detail_pname	product detail information (product detail information is the value returned when the detail product name is input required for purchase request)	N	O
bp_info	Value delivered by the application	N	O
tcash_flag	Whether T store Cash is used Y: used N: not used	N	M

- Example(When the Result is a success)

```
<?xml version="1.0" encoding="euc-kr"?>
<GXG_RES type="BillingLog">
<result>
        <status>0</status>
        <detail>0000</detail>
        <message>The inquiry is made normally.</message>
       <appid>OA00000001</appid>
</result>
<billing_log>
       <item>
                <tid>A5E9340E25BC88AA6FA6A833268832F0FA19CA082B8031858329168
```

```
D90899C54</tid>
                <product_id>OP00000001</product_id>
                <log_time>20120321154451</log_time>
                <charge_amount>1,517</charge_amount>
                <detail_pname>X</detail_pname>
                <bp_info>X</bp_info>
                <tcash_flag>N</tcash_flag>
        </item>
</billing_log>
</GXG_RES>
```

- Example(When the Result is a failure)

```
<?xml version="1.0" encoding="euc-kr" ?>
<GXG_RES type="AuthToken">
<result>
<status>11</status>
<detail>0012</detail>
<message>Token value is not correct.</message>
</result>
</GXG_RES>
```

6. FAQ.

SEQ	Items	Contents
1	Q	Inquiry of error code 2003 -5.
		purchase in Wi-Fi state -> input 4-digit authentication number and then a pop-up before completing the purchase, just as having to switch to 3G comes with an error code 2003-5, get a callback error message but has remained a successful self-test billing record in T store.
	A	in-app billing library use internal socket communications. When Network Wi-Fi -> 3g, or 3g -> Wi-Fi network disconnection(-2) because of bad connection, when socket fails to response listen (-5). In such cases, using authentication API to check implementing a defense code first before purchasing or after using purchasing function by in-app product' ID for purchased parts certification issued by executing a function item processing.
2	Q	Inquiry of popup UI bug.
		After check the 'T cash use' and return by home key, 'T cash use' is a check on the loose is a little bug.
	A	In-app billing library is following the Android Activity lifecycle like switching screen. So provide the proper setting when onPause(), onResume() internal saving switching screen, pop-up state. There are ways to save on the internal variables for T store cash. But we have the issue assigned a variable internal to the memory occupied. So, The library does not stored internally about important information necessary for billing (billing information pop-up product information, etc)
3	Q	Problem about initialization.
		The application happens to die after running the application and turn off the application.
	A	Please check the initialization part. In MainActivity, the case that IAPActivity inheritance can occur if not initialization properly. If don't initiallize the IAPActivity in Main Activity at onCreate(), the task happens to die and execute by android life cycle and occur the nullException error. So, Initialize the IAPLiblnit in the case that IAPActivity inheritance in MainActivity at onCreate().
4	Q	Inquiry about expiration date for checking monthly fixed price product.
		How to check the expiration date for checking monthly fixed price product.
	A	Call sendItemAuth API after finishing monthly fixed price product. If there are valid, it returns expiration date and count through "public void onItemAuthInfo(ItemAuthInfo itemAuth)".

		Ex)
		
		public void onItemAuthInfo(ItemAuthInfo itemAuth) {
		Log.*d("DEBUG","onItemAuthInfo!!");*
		String temp= **new String(itemAuth.pExpireDate);**
		Toast toast = Toast.*makeText(Sample.**this, temp, Toast.LENGTH_SHORT);***
		toast.show();
		byte bEd[] = itemAuth.pExpireDate;
		String str = **new String(bEd);**
		Log.*d("DEBUG","EXpire Datae = " + str);*
		int count = itemAuth.pCount;
		Log.*d("DEBUG","Count = " + str);*
		}

5	Q	How to check the expiration date for checking monthly fixed price product after terminating monthly fixed price product.
		How to check the expiration date when I terminate monthly fixed price product?
	A	When the next month after terminating monthly fixed price product, "pExpireDate" returns expiration date and "pCount" returns 0.
6	Q	Inquiry of TID purchase history data.
		TID purchase history data is not viewed.
	A	To inquiry of TID purchase history data, **Application should call popPurchaseDlg(pID, pName, pTID, pBpinfo) after making "TID" in Application.**
7	Q	In-App-Purchase popup orientation error
		When pop-up turn it back on again after lock the screen on horizontal mode of pop-up, it appears vertical mode.
	A	This problem occurs if the onResume() is not called for the difference in the specific Framework. Delete android:configChanges="orientation" in AndroidManifest.xml. Next, call "Android life cycle" after onCreate().
8	Q	SMS authentication, contact Unity3D library.
		LGU + SMS authentication at the terminal in the Unity 3D library, Intermittently SMS authentication number input does not automatically entered Situation occurs.
	A	You need to check whether another process are internally when you receive an SMS broadcast. (Library-related, such as forced Activity finish () etc)
9	Q	Time delay problem from the [Payment Product] pop-up screen.
		If you use "URLConnection.setDefaultUseCaches(Boolean)" function, pop-up

		exposed speed will be delayed.
	A	If you use "URLConnection.setDefaultUseCaches(Boolean)" function, Network failure occurs. You should to Use URLConnection.setUseCaches(boolean) rather than URLConnection.setDefaultUseCaches(boolean).
10	Q	KidsRock operation occurs errors.
		After KidsLock auth, next process is not occurred. In-App Billing pop-up window will appear.
	A	If you use onActivityResult(int request, int result, Intent data) API , You should add super.onActivityResult(request, result, data)" necessarily.

End of Document