Integrate Samsung In App Purchase Process Overview

1. Preparation & Request for Commercial Seller
2. Integrate IAP Helper
3. Configuration & Item Processing
4. IAP Helper Programming
5. Samsung IAP Server API
6. Submit App to Galaxy Store

Integrate Samsung In App Purchase SDK - Preparation

SAMSUNG

DETAILED STEPS

1. **Create a project with a unique package name**
   - If package names are not different, app update malfunctions and problems with marketing and promotional support may occur

2. **Add permissions to AndroidManifest.xml**
   - com.samsung.android.iap.permission.BILLING to connect to IAP and enable in-app item registration in Seller Portal.
   - android.permission.INTERNET because IAP uses the internet

3. **Must be an approved Commercial Seller**

4. **Register an app and in-app items in Seller Portal (refer here)**
   - During IAP integration, you may need to test IAP features
   - Samsung IAP needs information about your app and in-app items registered in Seller Portal
   - With the com.samsung.android.iap.permission.BILLING permission added to your app's AndroidManifest.xml file, you can register an incomplete app and one or more in-app items

Refer to Webpage for Further Details	
Samsung In App Purchase SDK Overview	https://developer.samsung.com/iap/overview.html
Item Registration Guide	https://seller.samsungapps.com/guidePopup.as?numcid=0302030000&localeLanguage=en
Samsung In App Purchase SDK	https://developer.samsung.com/iap/release-note.html

Integrate Samsung In App Purchase SDK- Integrate IAP Helper Into Your App

SAMSUNG

DETAILED STEPS

1. **Download IAP 6.0 Helper and sample apps**
   - Add the IAP Helper to your app
   - Click File → New → Import module
   - Select the IAP6Helper folder from the Source Directory, and click Finish
   - From Project Explorer, click Open Module Settings or press F4 → Dependencies tab
   - Select your app module, and click + → Module Dependency from Declared Dependencies
   - Select the IAP6Helper module, and click OK

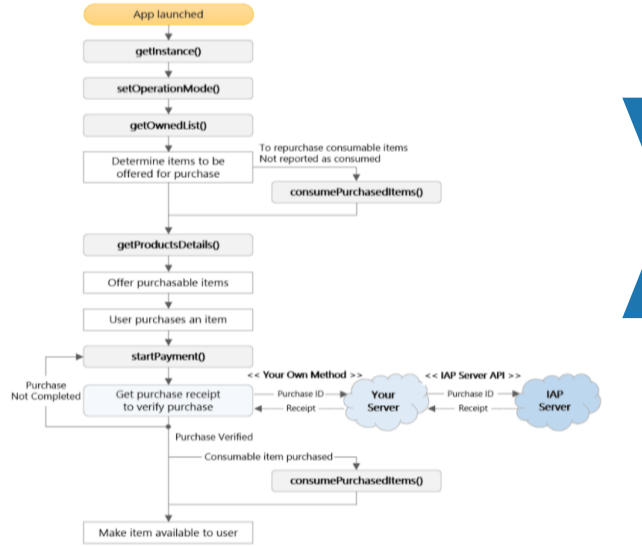
2. **The zipped file contains the SDK and sample projects that uses the IapHelper**

Location	Module	Description
./Libs	IAP6Helper	IAP v6.0 SDK
./Samples	IAP6Sample	Sample Java app using IAP v6.0 SDK
./Samples	IAP6SampleKotlin	Sample Kotlin app using IAP v6.0 SDK

3. **IAP v6.0 is supported since Android API 18 or higher. It does not work properly in lower versions**

Refer to Webpage for Further Details	
Samsung In App Purchase SDK Overview	https://developer.samsung.com/iap/overview.html
Samsung In App Purchase SDK	https://developer.samsung.com/iap/release-note.html
Test Guide	https://developer.samsung.com/iap/iap-test-guide.html

Integrate Samsung In App Purchase SDK - IAP Configuration and In-App Item Processing



DETAILED STEPS

1. Your app code can follow the logic flow to the left to support your in-app items. The scope of each IAP Helper API is straightforward and most integrations follow a similar logic flow, which is linear with a few branches. After item purchases, IAP Server APIs are typically called to verify purchases

2. Even if the user has completed payment for the item purchase, the item might not be available to the user because of a battery or network problem. Make sure you call getOwnedList() whenever launching the application to check if there are consumable items that are not reported as consumed, and if so, consume them immediately by calling consumePurchasedItems(). Then, the user can repurchase the items

Refer to Webpage for Further Details	
Samsung In App Purchase SDK Overview	https://developer.samsung.com/iap/overview.html
IAP Helper API	https://developer.samsung.com/iap/programming-guide/iap-helper-programming.html
IAP Servers API	https://developer.samsung.com/iap/programming-guide/samsung-iap-server-api.html
Samsung In App Purchase SDK	https://developer.samsung.com/iap/release-note.html

Integrate Samsung In App Purchase SDK - IAP Helper Programming

SAMSUNG

DETAILED STEPS

1. **Instantiate Samsung In-App Purchase - Before your app can make IAP requests, it must call getInstance() to create a singleton instance of IapHelper**

2. **Set the IAP operation mode**
   - IAP supports three operational modes. One is for enabling billing for item purchases and the other two are for testing IAP functions without billing app users for item purchases
   - If setOperationMode() is not called, operation mode is set to OPERATION_MODE_PRODUCTION by default
   - Ensure the operation mode is set to OPERATION_MODE_PRODUCTION before submitting for Beta test or normal publication

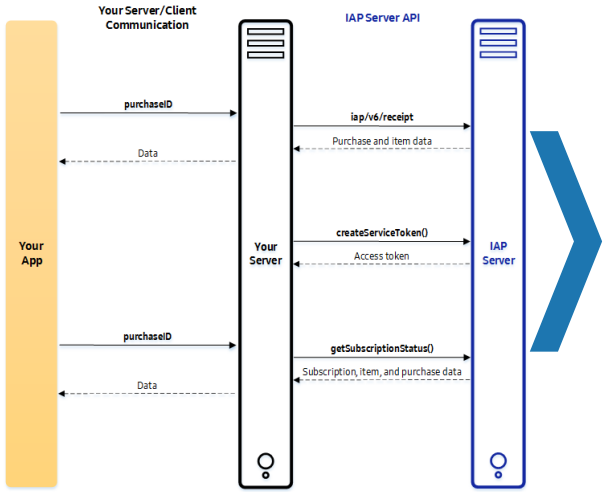
3. **Get in-app item details**
   - getProductsDetails() returns information for one, more, or all in-app items registered to the app
   - Returns item data and processing results specified by the OnGetProductsDetailsListener interface

4. **Purchase an in-app item**
   - startPayment() initiates the purchase and payment transaction of the specified in-app item and can notify the end user if the purchase succeeded or failed. Returns the item data and transaction results and data specified in the OnPaymentListener interface
   - You can specify a passThroughParam parameter value to enhance purchase security. During purchases with passThroughParam values created and passed by an IAP-integrated application are returned in the responses

5. **Acknowledge a purchased consumable item**
   - consumePurchasedItems() reports one or more purchased consumable items as consumed, which makes the items available for another purchase. The app user may or may not have used the items. Returns item data and processing results specified by the OnConsumePurchasedItemsListener interface

Refer to Webpage for Further Details	
Samsung In App Purchase SDK Overview	https://developer.samsung.com/iap/overview.html
IAP Helper API	https://developer.samsung.com/iap/programming-guide/iap-helper-programming.html
IAP Servers API	https://developer.samsung.com/iap/programming-guide/samsung-iap-server-api.html

Integrate Samsung In App Purchase SDK - Samsung IAP Server API



DETAILED STEPS

1. **Verify a purchase**
   - iap/v6/receipt enables your server and client app to verify that a specified in-app item purchase and payment transaction was successfully completed
   - The API returns a JSON object with a successful status and details about a successful transaction and the item or with a failure status
   - This API can help to prevent malicious purchases and ensure that purchase and payment transactions were successful when the client app experiences network interruptions after an item purchase and payment transaction

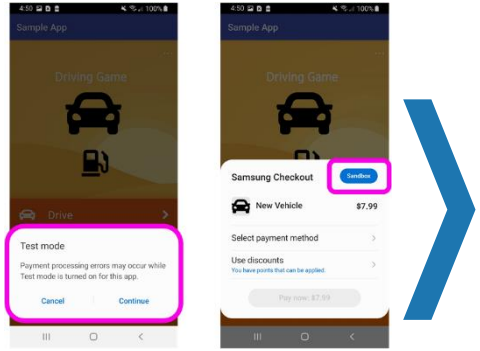
2. **Create a service token**
   - createServiceToken generates and returns access token value that your server must use to authenticate getSubscriptionStatus SOAP requests. Each token is valid for 30 days

3. **Check subscription status**
   - getSubscriptionStatus gets subscription status, item information, and purchase information of a specified Auto Recurring Subscription (ARS) item that was purchased previously

Refer to Webpage for Further Details	
Samsung In App Purchase SDK Overview	https://developer.samsung.com/iap/overview.html
IAP Helper API	https://developer.samsung.com/iap/programming-guide/iap-helper-programming.html
IAP Servers API	https://developer.samsung.com/iap/programming-guide/samsung-iap-server-api.html
Samsung In App Purchase SDK	https://developer.samsung.com/iap/release-note.html

Integrate Samsung In App Purchase SDK - Test Guide

SAMSUNG



DETAILED STEPS

1. If OPERATION_MODE_TEST is set, the pop-up is displayed and Sandbox is shown at the top right of the payment window to indicate that the app user will not be billed for item purchases

2. Test your game in IAP operation modes (defined in source code)
   - Testers must be registered as a License Tester in the seller's Seller Portal profile.
   - Production mode for commercial use with real transaction
   - Test mode for success and failure (fake transaction)

3. Test your game in Beta - for more information, refer here
   - Test real IAP transaction of your items
   - Register tester's Samsung Accounts to distribute the beta release
   - Whitelist the tester's accounts as Valid License Tester to avoid getting charged

4. Be sure to switch from test mode to production mode when submitting application for

Refer to Webpage for Further Details	
Samsung In App Purchase SDK Overview	https://developer.samsung.com/iap/overview.html
Test Guide	https://developer.samsung.com/iap/iap-test-guide.html
Beta Testing	https://seller.samsungapps.com/guidePopup.as?numcid=0201090000&localeLanguage=en

Integrate Samsung In App Purchase SDK - Submit the App to Galaxy Store

SAMSUNG

DETAILED STEPS

1. **Check the operation mode**
   - After IAP integration, you must check the operation mode before submitting the app. If you submit the app with OPERATION_MODE_TEST, the users will get an error message if they try to purchase an item. So, before beta release or normal publication, confirm that the operation mode is OPERATION_MODE_PRODUCTION

2. **Beta test**
   - Before submitting your app and its in-app items for review testing and normal publication in the Galaxy Store, you can beta release a test version of the app and one or more of its items to test IAP integration

3. **Submit the app**
   - When you have created an app version that is ready for review testing and normal publication in Galaxy Store, click Submit

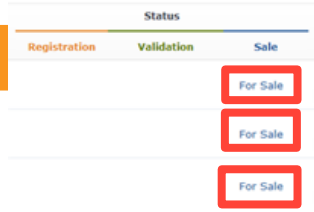
Refer to Webpage for Further Details	
Samsung In App Purchase SDK Overview	https://developer.samsung.com/iap/overview.html
Samsung In App Purchase SDK	https://developer.samsung.com/iap/release-note.html
App Registration Guide	https://seller.samsungapps.com/guidePopup.as?numcid=0201010000&localeLanguage=en

Application Validation Steps

SAMSUNG

PRE REVIEW (~1 Business Day) → **FOR SALE** → **POST DEVICE TEST (~2-4 Business Day)**







- **Examines:**
  - Security issue
  - Contents
  - Basic function test[1]

- **If android application passes this step, it will be live in Galaxy Store right away[2]**
- **If there is no reason for suspension or removal, the "For Sale" status is granted**

- **Examines application functions**
- **If application is found to have additional function issue, it may be suspended or device groups in problem can be removed**

QUICK PASS[3] (~15 minutes) → **FOR SALE** → **POST DEVICE TEST (~2-4 Business Day)**

1 Such as installation, execution, uninstallation and etc.
2 All non android app (watch, theme, and stickers) must pass device test to be alive in Galaxy Store
3 Contact your Business Developer Partner to be added to Quick Pass

Samsung IAP Unity and Unreal Plugin

SAMSUNG

GENERAL INFORMATION

- **Unity Plugin**
  - The Samsung IAP Unity plugin comes with all the content necessary for your integration.
  - Plugins folder contains all Samsung IAP related scripts and libraries.
  - Menu Scene is the Unity scene that demonstrates all the functionality of the Samsung IAP Unity plugin.
  - https://developer.samsung.com/iap/samsung-iap-unity-plugin.html#

- **Unreal Plugin**
  - The Samsung IAP Unreal plugin comes with all the content necessary for your integration.
  - The guide explains how to use the Samsung In-App Purchase (IAP) functionality by integrating the Samsung IAP Unreal Engine Plugin.
  - https://developer.samsung.com/iap/samsung-iap-unreal-plugin.html#

Refer to Webpage for Further Details	
Unity Plugin	https://developer.samsung.com/iap/samsung-iap-unity-plugin.html
Unreal Plugin	https://developer.samsung.com/iap/samsung-iap-unreal-plugin.html