|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **`Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 25-03-2016 | Indrajit Kumar | All | Initial draft |
| 0.2 | 28-04-2016 | Amit Kumar | All | Update interface APIs |
| 0.3 | 10-06-2016 | Indrajit Kumar | Initialization and InApp Purchase API reference | Update interface APIs |
| 0.4 | 29-06-2016 | Indrajit Kumar | InApp Purchase API reference and Cart Icon | Update interface APIs and  Cart icon drawable for vertical proposition use |

InApp Purchase Integration

**DOCUMENT HISTORY**

|  |  |
| --- | --- |
| **Author** | Indrajit and Amit |
| **Approved by** |  |
| **Email Id** | [Indrajit.kumar@philips.com](mailto:Indrajit.kumar@philips.com) and [amit.kumar\_5@philips.com](mailto:amit.kumar_5@philips.com) |

Table of Contents

[1. Introduction 3](#_Toc454981665)

[2. Prerequisites 3](#_Toc454981666)

[3. Library Integration 3](#_Toc454981667)

[3.1 Maven repository Integration 3](#_Toc454981668)

[3.2 Library Integration 4](#_Toc454981669)

[4. INITIALIZATION 6](#_Toc454981670)

[5. InApp Purchase API Reference 7](#_Toc454981671)

[6. Api details 8](#_Toc454981672)

[7. InApp Cart Icon : To Add shopping cart icon for an vertical proposition, 8](#_Toc454981673)

[8. Reference APP link 10](#_Toc454981674)

[9. Notes 10](#_Toc454981676)

# Introduction

This document provides an overview of integration InApp Purchase feature in Android mobile application.

Source Path: [https:// atlas.natlab.research.philips.com/stash/scm/ip/in-app-purchase\_android.git](https://310164421@atlas.natlab.research.philips.com/stash/scm/ip/in-app-purchase_android.git)

# Prerequisites

I. Vertical project is configured for Android Studio

II. Setting->Developer Options->Don’t Keep Activities should be unchecked. [If Developer

Mode is on]

III. Android API version should on 10[Gingerbread]



# Library Integration

## Maven repository Integration

The easiest and preferred way to use these components is using maven. Only we need to add maven repositories in app build gradle:

allprojects {  
 repositories {  
 maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }  
 maven {  
 url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/ext-release-local'  
 }  
 maven {  
 url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-android'  
 }  
 maven {  
 url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-stage-local-android'  
 }  
 flatDir {  
 dirs 'libs'  
 }  
 }  
}

## Library Integration

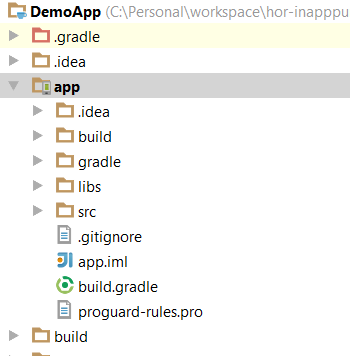
1. Check out the code from above path wherein you can find sample app which depends on InApp Purchase care library’s aar file :

[https:// atlas.natlab.research.philips.com/stash/scm/ip/in-app-purchase\_android.git](https://310164421@atlas.natlab.research.philips.com/stash/scm/ip/in-app-purchase_android.git)

1. InApp Purchase needs other libraries to build which are as below
2. registerationApi

Please make sure all these are added to application with the latest versions along with InAppPurchase library aar file under libs folder.

1. Refer InApp Purchase dependencies along with .aar files in build.gradle as mentioned below :



*Note: Since InApp Purchase Android was built with*

***'com.android.support:appcompat-v7:23+. version***

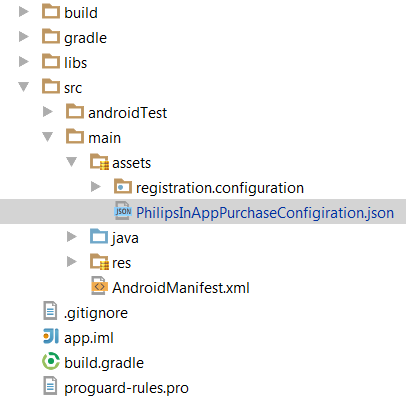
***so vertical app support libs should be greater or equal to mentioned versions***

dependencies {  
 compile(**'com.philips.cdp:iap:2.0.0'**)  
 compile(**'com.philips.cdp:registrationApi:6.0.2'**)

}

1. Paste InApp Purchase related assets in assets folder from reference app:

**Note: Don’t change or rename folder structure**



Inside the PhilipsInAppPurchaseConfiguration.json, app has to configure their host and site such as:

{  
 **"hostport"**: **"tst.pl.shop.philips.com"**,  
 **"propositionid"**: **"Tuscany2016"**}

There are 3 host for providing backend services such as:

1. **Test server: tst.pl.shop.philips.com**
2. **Acceptance server: acc.occ.shop.philips.com**
3. **Production Server: www.occ.shop.philips.com**
4. Clean project



# INITIALIZATION

InApp store selection is based on the combination of language and country.

1. Create IAPSettings :

Parameters: language (ex: two chars long, “en”)

Country (ex: two chars long, “US”, “UK”)

themeIndex: int constant pointing to UIKIt theme resource

(ex:R.style.Theme\_Philips\_DarkPurple\_WhiteBackground)

1. Create instance of IAPHandler class as below and call initialization by init.

Parameters: Context, IAPSettings

Example initialization for en-US

IAPHandler handler = new IAPHandler.*init*(**this**, **new** IAPSettings(**"US"**, **"en"**, R.style.Theme\_Philips\_DarkPurple\_WhiteBackground);

1. To launch as Fragment, vertical proposition has to inflate IAP action bar by creating instance of IAPActionLayout and add IAP custom action bar to vertical custom action bar LinearLayout.

**mCustomView** = **layout**.getCustomView(**this**);  
((ViewGroup) findViewById(R.id.***ll\_custom\_action***)).addView(**mCustomView**);  
ViewGroup mBackButton = (ViewGroup) **mCustomView**.findViewById(com.philips.cdp.di.iap.R.id.***iap\_header\_back\_button***);  
mBackButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(**final** View v) {  
 onBackPressed();  
 }  
});

Once IAPHandler instance initialized, refer below available APIs.

# InApp Purchase API Reference

These are the InApp Purchase APIs.

**Class Overview**

This class having the inApp purchase methods to access the cart.

**Summary**

Inherited Constants

|  |  |  |  |
| --- | --- | --- | --- |
| **Return type** | **Error Constants** | **Error Code** | **Error Descriptions** |
| int | IAPConstant.IAP\_ERROR\_NO\_CONNECTION | 2 | No Connection |
| int | IAPConstant.ERROR\_CONNECTION\_TIME\_OUT | 3 | Connection Timeout |
| int | IAPConstant.IAP.ERROR\_AUTHENTICSTION\_FAILURE | 4 | Authentication failure |
| int | IAPConstant.IAP\_ERROR\_INSUFFICIENT\_STOCK\_ERROR | 6 | Product out of stock |

Public Methods

|  |  |  |
| --- | --- | --- |
| View | getCustonView(Context context) | It will set IAP custom action bar on fragment |
| void | setBackgroundDrawable(Drawable drawable) | It will set the background of IAP custom action bar. |
| Void | setBackButtonDrawable(Drawable drawable) | It will set the back button drawable on IAP action bar. |
| boolean | onHWBackPressed() | It will return true when IAP handled their implementation else return false |
| void | setLaunchAsFragment(boolean launchAsFragment) | Set true if launching in fragment else set false to launch as Activity |
| void | setUseLocalData(boolean useLocalData) | Set true for only rertailer feature from IAP |
| void | setFragmentProperties(FragmentManager fragmentManger, int containerId) | Set container id where IAP fragments will get inflates. |
| IAPHandler | IAPHandler init(Context context, IAPSettings config) | First api call for initializing IAPHandler with language, country and themes. |
| void | **launchIAP(int** landingView, String ctnNumber, IAPHandlerListener listener) | Launching IAP component |
| void | getProductCartCount(IAPHandlerListner pIAPHandlerListner)) | It will give the count of products from cart |
| void | launchCategorizedCatalog(ArrayList<String> productList) | It will launch the given product list in IAP |
| void | getCompleteProductList(IAPHandlerProductListListener  pIAPHandlerProductListListener) | It will give complete product list after retrieving from Hybris server |

Inherited Methods

|  |  |  |
| --- | --- | --- |
| void | onSuccess(int count) | getProductcount will give actual count on onSuccess call and for others, it will return count 0. |
| void | onSuccess(ArrayList<String> pProductList) | getCompleteProductList will give complete product list on onSuccess call |
| void | onFailure(int errorCode) | It will return int constants for error |

# Api details

1. launchIAP(int landingView, String ctnNumber, IAPHandlerListener listener)

landingView: Constant value of IAPConstatns.IAPLandginViews

IAP\_PRODUCT\_CATALOG\_VIEW

IAP\_SHOPPING\_CART\_VIEW

IAP\_PRODUCT\_DETAIL\_VIEW

IAP\_PURCHASE\_HISTORY\_VIEW

ctnNumber: Product ctnNumber

If ctnNumber is empty, product catalogue is launched, else the product is added in cart (if not already present) and shopping cart is launched.

# InApp Cart Icon : To Add shopping cart icon for an vertical proposition,

<FrameLayout

        android:id="@+id/cart\_container"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_alignParentEnd="true"

        android:layout\_centerVertical="true"

        android:layout\_gravity="center\_vertical"

        android:paddingLeft="15dp"

        android:paddingRight="15dp"

        android:visibility="gone">

        <ImageView

            android:id="@+id/cart\_icon"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_gravity="center\_vertical"

            android:layout\_marginLeft="10dp"

            android:paddingRight="@dimen/network\_margin\_left\_right"

            android:visibility="gone" />

        <com.philips.cdp.uikit.customviews.BadgeView

            android:id="@+id/item\_count"

            style="@style/NotificationLabel.Small"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_gravity="top|left"

            android:layout\_marginBottom="15dp"

            android:visibility="gone" />

</FrameLayout>

mCartIcon = (ImageView) mMainLayout.findViewById(R.id.cart\_icon);

 mCartIconDrawable = VectorDrawable.create(context, **R.drawable.iap\_shopping\_cart**); //Vector Drawable

mCartIcon.setBackground(mCartIconDrawable);

**Vector Drawable (R.drawable.iap\_shopping\_cart)**

    <vector

    xmlns:android="<http://schemas.android.com/apk/res/android>"

    xmlns:app="<http://schemas.android.com/apk/res-auto>"

    android:width="20dp"

    android:height="18dp"

    android:viewportHeight="370.7"

    android:viewportWidth="425.2"

    app:uikit\_vc\_viewportHeight="370.7"

    app:uikit\_vc\_viewportWidth="425.2">

    <path

        android:fillColor="#fff"

        android:pathData="M267.4,308.8c-19.6,-1.9 -36,14.4 -34,34c1.4,14.5 13.2,26.3 27.7,27.7c19.6,1.9 36,-14.4 34,-34C293.7,322 281.9,310.2 267.4,308.8z"

        app:uikit\_vc\_fillColor="#fff"

        app:uikit\_vc\_pathData="M267.4,308.8c-19.6,-1.9 -36,14.4 -34,34c1.4,14.5 13.2,26.3 27.7,27.7c19.6,1.9 36,-14.4 34,-34C293.7,322 281.9,310.2 267.4,308.8z"/>

    <path

        android:fillColor="#fff"

        android:pathData="M115.6,308.7c-19.6,-1.9 -36,14.4 -34,34c1.4,14.5 13.2,26.3 27.7,27.7c19.6,1.9 36,-14.4 34,-34C141.9,321.9 130.1,310.2 115.6,308.7z"

        app:uikit\_vc\_fillColor="#fff"

        app:uikit\_vc\_pathData="M115.6,308.7c-19.6,-1.9 -36,14.4 -34,34c1.4,14.5 13.2,26.3 27.7,27.7c19.6,1.9 36,-14.4 34,-34C141.9,321.9 130.1,310.2 115.6,308.7z"/>

    <path

        android:fillColor="#fff"

        android:pathData="M351.5,0c-25.2,0 -41.7,10.3 -46.6,37.2l-1.9,12l-289.3,0c-9.6,0 -16.2,9.6 -12.8,18.5l21.9,58c22.9,59.3 99.2,82.6 163.7,82.6H278c-4.1,25.7 -23.8,44.3 -46.9,44.3l-149.8,0l0,25.9c0,0 110.5,0 151.8,0c41.4,0 77.8,-32.3 84.7,-76.9l24.4,-155.4c1.6,-11.8 5.5,-14.8 16.5,-15.6l66.6,-4.9V0H351.5z"

        app:uikit\_vc\_fillColor="#fff"

        app:uikit\_vc\_pathData="M351.5,0c-25.2,0 -41.7,10.3 -46.6,37.2l-1.9,12l-289.3,0c-9.6,0 -16.2,9.6 -12.8,18.5l21.9,58c22.9,59.3 99.2,82.6 163.7,82.6H278c-4.1,25.7 -23.8,44.3 -46.9,44.3l-149.8,0l0,25.9c0,0 110.5,0 151.8,0c41.4,0 77.8,-32.3 84.7,-76.9l24.4,-155.4c1.6,-11.8 5.5,-14.8 16.5,-15.6l66.6,-4.9V0H351.5z"/>

    </vector>

# Reference APP link

# <https://atlas.natlab.research.philips.com/stash/scm/ip/in-app-purchase_android.git>

# Notes

1. Registration is developed as separate library project. App is expected to invoke registration library on click of registration button inside consumer care support page.
2. Library should be initialized as per document or sample app otherwise library would throw runtime exception.
3. Please refer sample application for more details
4. All dependencies can be referred as it is.
5. Configuration can be followed [all sample configurations are provided in sample]