InAppPurchase Integration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 25-03-2016 | Team Phoenix | All | Initial  draft |
| 0.2 | 28-04-2016 | Team Phoenix | All | Update interface APIs |
| 0.3 | 10-06-2016 | Team Phoenix | Initialization and InApp Purchase API reference | Update interface APIs |
| 0.4 | 29-06-2016 | Team Phoenix | InApp Purchase API reference and Cart Icon | Update interface APIs |
| 0.5 | 21/09/2016 | Team Phoenix | All | Update interface APIs |
| 1.0 | 01/12/2016 | Team Phoenix | All | Update interface APIs |
| 1.1 | 26/04/2017 | Team Phoenix | Updated source path | Migrated to TFS |
| 1.2 | 18/05/2017 | Team Phoenix | Initialization | Getting APP VERSION\_NAME |
| 1.3 | 12/09/2017 | Team Phoenix | Initialization | Update interface APIs |

|  |  |
| --- | --- |
| **Author** | Team Phoenix |
| **Approved by** | Rahul (Architect) |
| **Email ID** | DL\_ph\_cdp2\_iap@philips.com |

Contents

[**Introduction** 4](#_Toc492979271)

[**Pre-Requisite** 4](#_Toc492979272)

[**Maven Repository** 4](#_Toc492979273)

[**Library Integration** 4](#_Toc492979274)

[**Configuration** 4](#_Toc492979275)

[**Dependencies** 5](#_Toc492979276)

[**Initialization** 5](#_Toc492979277)

[**IAPLaunchInput** 5](#_Toc492979278)

[Instantiation 5](#_Toc492979279)

[Set Input 5](#_Toc492979280)

[Getter and setter 5](#_Toc492979281)

[IAPFlows 6](#_Toc492979282)

[IAPFlowInput 6](#_Toc492979283)

[IAPListener 6](#_Toc492979284)

[**IAPDependencies** 7](#_Toc492979285)

[Instantiation 7](#_Toc492979286)

[**IAPSettings** 7](#_Toc492979287)

[Instantiation 7](#_Toc492979288)

[**IAPInterface** 7](#_Toc492979289)

[Instantiation 7](#_Toc492979290)

[Initialization 7](#_Toc492979291)

[Exposed API’s 7](#_Toc492979292)

[**Action bar cart icon** 8](#_Toc492979293)

[Layout 8](#_Toc492979294)

[Drawable 9](#_Toc492979295)

# **Introduction**

This document provides an overview of integration of InAppPurchase feature in Android Mobile applications.

Source Path: <http://tfsemea1.ta.philips.com:8080/tfs/TPC_Region24/CDP2/_git/iap-android-in-app-purchase>

# **Pre-Requisite**

* Android Studio 1.5 or higher
* A device running Android version 4.4 or newer

# **Maven Repository**

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**

* InAppPurchase can be integrated by adding the library to the build.gradle file as compile(**‘com.philips.cdp:iap’**)
* Source code path of the library with the sample demo app : <http://tfsemea1.ta.philips.com:8080/tfs/TPC_Region24/CDP2/_git/iap-android-in-app-purchase>

# **Configuration**

Below are the configuration file has to be placed in assets folder of the vertical proposition,

1. AppConfig.json
2. ADBMobileConfig.json

In AppConfig.json, “propositionID” must be placed in the “IAP” object as,

“IAP”:{  
 "propositionid": "Tuscany2016"  
}

# **Dependencies**

* Philips User Registration (8.1.0-SNAPSHOT)
* com.android.support:appcompat-v7:24.2.0

# **Initialization**

**Obtain InApp Component version number from gradle file**

Use the below code for obtaining the library version number from gradle file

String version = com.philips.cdp.di.iap.BuildConfig.VERSION\_NAME;

The InAppPurchase component can be initialized as follows,

IAPLaunchInput **mIAPLaunchInput** = **new** IAPLaunchInput();

IAPDependencies **mIAPDependencies** = **new** IAPDependencies(**new** AppInfra.Builder().build(**this**));

IAPSettings **mIAPSettings** = **new** IAPSettings(**this**);

IAPInterface **mIapInterface** = **new** IAPInterface();

mIapInterface.init(**mIAPDependencies**, **mIAPSettings**);

# **IAPLaunchInput**

This class provides the interface to set the input for the landing screen. Also it has the getter and setter method for the IAPListener. This class contains the enum which represents landing screen which vertical uses to launch the IAP.

### Instantiation

IAPLaunchInput **mIAPLaunchInput** = **new** IAPLaunchInput();

### Set Input

setIAPFlow(int landingView, IAPFlowInput iapFlowInput)

setIAPFlow(int landingView, IAPFlowInput iapFlowInput, ArrayList<String> iapIgnoredRetailersList)

**ex:**

IAPFlowInput input = new IAPFlowInput(“HX8071/10”);

mIAPLaunchInput.setIAPFlow(IAPLaunchInput.IAPFlows.IAP\_PRODUCT\_CATALOG\_VIEW, input)

### Getter and setter

IAPListener getIapListener()

setIapListener(IAPListener iapListener)

### IAPFlows

Below are the types of landing views:

public interface IAPFlows {

int IAP\_PRODUCT\_CATALOG\_VIEW = 0; // Navigate the user to the product catalog page

int IAP\_SHOPPING\_CART\_VIEW = 1; // Navigate the user to the shopping cart page

int IAP\_PURCHASE\_HISTORY\_VIEW = 2 ; // Navigate the user to the purchase history page

int IAP\_PRODUCT\_DETAIL\_VIEW = 3; // Navigate the user to the product detail page

int IAP\_BUY\_DIRECT\_VIEW = 4; // Navigate the user to the order summary page

}

### IAPFlowInput

This class takes the input for the landing view via constructor and provides the getter to retrieve the input.

Constructors:

IAPFlowInput(String productCTN)

IAPFlowInput(ArrayList<String> prductCTNs)

Getter methods:

String getProductCTN()

ArrayList<String> getProductCTNs()

### IAPListener

Below are the methods provided by this listener,

onGetCartCount(int count) : Returns the cart count // Count 0 indicates no products (badge icon should be invisible/gone) and -1 indicates Retailer flow(Shopping cart icon should be invisible/gone)

onUpdateCartCount() : Updates the cart count when the product is added to the cart

updateCartIconVisibility(boolean shouldShow) : Returns whether cart icon should be visible or not

onGetCompleteProductList(final ArrayList<String> productList) : Returns the list of product from the server

onSuccess() : Returns success, so that the vertical can do operations like dismissing progress dialog, etc.,

//This will get deprecated later in next PI.

onSuccess(Object data) : Returns success with return data, so that the vertical can do operations like dismissing progress dialog with data back to vertical, etc.,

onFailure(final int errorCode) : Returns error, so that the vertical can handle accordingly

# **IAPDependencies**

This class handles the dependency required for IAP. Currentlt IAP has one dependency i.e AppInfra. So, vertical needs to initialize IAPDependencies and set the app infra object. This app infra object will be used for logging, tagging and configuration.

### Instantiation

IAPDependencies **mIAPDependencies** = **new** IAPDependencies(**new** AppInfra.Builder().build(**this**));

# **IAPSettings**

Currently IAP don’t have any settings to be initialized by vertical. Only instatiation of IAPSettings is required which must be passed while creating IAPInterface class.

### Instantiation

IAPSettings **mIAPSettings** = **new** IAPSettings(**this**);

# **IAPInterface**

This class expose the methods to interact with the IAP.

### Instantiation

IAPInterface **mIapInterface** = **new** IAPInterface();

### Initialization

init(UappDependencies uappDependencies, UappSettings uappSettings)

**ex**: mIapInterface.init(**mIAPDependencies**, **mIAPSettings**);

### Exposed API’s

* isCartVisible(IAPListener iapListener)

**ex**: mIapInterface.isCartVisible(this);

The vertical will be notified with the isCartVisible API of IAPListener to act whether flow should go in retailer flow or Hybris flow. **This is a mandatory API call that needs to be called before calling any other public API**.

* getProductCartCount(IAPListener iapListener)

**ex**: mIapInterface.getProductCartCount(this);

The vertical will be notified with OnGetCartCount API of IAPListener to act accordingly

* getCompleteProductList(IAPListener iapListener)

**ex**: mIapInterface.getCompleteProductList(this);

The vertical will be notified with the onGetCompleteProductList API of IAPListener to act accordingly. This API call should be done before launch API and same CTNs should be passed if launching in Hybris flow else don’t required to make this API call.

* launch(UiLauncher uiLauncher, UappLaunchInput uappLaunchInput)

**ex**:mIapInterface.launch(newActivityLauncher(ActivityLauncher.ActivityOrientation.SCREEN\_ORIENTATION\_PORTRAIT, DEFAULT\_THEME), mIapLaunchInput);

The respective screen will be launched based on the “mLaunchInput” parameter passed.

**Note: “isCartVisible” api is a mandatory method after init api. This method handles the service discovery calls and based on that cart icon visibility will be decided. So, if the user changes the country then this method needs to be called again to decide the visibility of cart icon. So after initialization of InApp Purchase, and before calling any other methods, proposition has to call “isCartVisible” method atleast once, and post country change.**

# **Action bar cart icon**

### Layout

<FrameLayout

android:id="@+id/shopping\_cart\_icon"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_centerVertical="true"

android:layout\_gravity="center\_vertical"

android:clickable="true"

android:paddingLeft="15dp"

android:paddingRight="15dp"

android:visibility="gone">

<ImageView

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:contentDescription="@string/app\_title"

android:paddingRight="12dp"

android:src="@drawable/shopping\_cart\_xml" />

<com.philips.platform.uid.view.widget.NotificationBadge

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

style="@style/NotificationBadge.Small"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="top|end"

android:textColor="@android:color/white"

android:textSize="16sp"

android:textStyle="bold"

android:visibility="visible" />

</FrameLayout>

### Drawable

@drawable/shopping\_cart\_xml

<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"

<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"/>

<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"/>

<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">

</vector>