
POKKT AIR SDK Developer API Guide

V1.1

Steps to follow (Please follow the Sample_PokktSDK example project):

These instructions are for Flash Builder 4.7, If you are using other version or different tool like flash developer please modify process accordingly.

1. There are two ANEs provided by POKKT, which you need to add in your project properties. First one is PokktASEExtension.ane, which is required for all integrations and second one is GooglePlayService.ane, which is only required if you are not using ANE for Google Play Service in your project already.

2. Basic steps to add the ANE file in your project.

- Right click on your project and select *Properties->ActionScript Build Path (from left pane) ->Native Extensions (Tab from Right pane)->Click on Add ANE->Browse the shared ANE and add it.*
- After adding ANE select *ActionScript Build Packaging* from same *Properties* popup. ActionScript Build Packaging has two options one is Apple iOS and 2nd is Google Android. So select *Google Android->Native Extensions (tab from right side)->Check the check box for added ANE->Apply->ok.*

3. Update your Application Descriptor.

After adding ANE in your project properties extensions will get added in your .xml file (example: Sample_PokktSDK-app.xml). But if due to any reason it doesn't get added then need to add manually.

```
<extensions>
    <extensionID>com.pokkt.pokktsdk</extensionID>
    <extensionID>com.pokkt.extension</extensionID>
</extensions>
```

Now need to add **Permissions**, **Activities**, and **receivers**. So update below code in your xml file.

And also need to change **security_key** and **application_id** in below code according to your App. Also change the **integration type**. Possible values are 0 for INTEGRATION_TYPE_ALL, 1 for INTEGRATION_TYPE_OFFERWALL and 2 for INTEGRATION_TYPE_VIDEO

Permission:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Activity and Receivers:

Notes: RED color text will be in one line.

```
<application>
    <activity
        android:name="com.bigflix.BigFlixMovies.VideoNextActivity"
        android:configChanges="navigation|orientation|screenLayout|uiMode|
            screenSize|smallestScreenSize"
        android:label="@string/app_name" />
    <activity
```

```

        android:name="com.app.pokktsdk.PokktManager"
        android:configChanges="keyboard|keyboardHidden|navigation|orientation|
        screenLayout|uiMode|screenSize"
        android:label="@string/app_name"
        android:windowSoftInputMode="adjustPan" />

<activity
    android:name="com.app.pokktsdk.PlayVideoCampaignActivity"
    android:configChanges="keyboard|keyboardHidden|navigation|orientation|
    screenLayout|uiMode|screenSize"
    android:label="@string/app_name"
    android:screenOrientation="landscape"
    android:windowSoftInputMode="adjustPan" >
</activity>

<receiver android:name="com.app.pokktsdk.AppInstallBroadcastReceiver" >
    <intent-filter android:priority="1000" >
        <action android:name="android.intent.action.PACKAGE_INSTALL" />
        <action android:name="android.intent.action.PACKAGE_ADDED" />
        <data android:scheme="package" />
    </intent-filter>
</receiver>

<meta-data
    android:name="security_key"
    android:value="iJ021Jss0M" /> //Change this Security Key.
<meta-data
    android:name="application_id"
    android:value="a2717a45b835b5e9f50284a38d62a74e" /> //Change this
                                                    Application ID.
<meta-data
    android:name="U_ID"
    android:value="123456" /> //Change this U_ID.
<meta-data
    android:name="integration_type"
    android:value="0" /> //Change this value: 0:both, 1:Offerwall, 2:Video
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
<meta-data
    android:name="implementation_class"
    android:value="com.pokkt.pokktsdk.OfferWallEventListenerImpl" />

</application>

```

4. Add below code in your main class. All these are events which has own event handler.

//-----

Create the instance of PokktASEExtension. And call the initPokkt function which will create the extension of PokktASEExtension.

```

_pokktExtension = new PokktASEExtension();
_pokktExtension.initPokkt();
_pokktExtension.performOperation(PokktOperation.GET_PENDING_COINS);

```

Add the event listener for all response, so in handler do the job according to your need. Please follow the sample project for all handler.

```

        _pokktExtension.addListener(PokktEvent.DOWNLOAD_COMPLETED_EVENTS,
onDownloadCompleted);
        _pokktExtension.addListener(PokktEvent.DOWNLOAD_FAILED_EVENTS, onDownloadFailed);
        _pokktExtension.addListener(PokktEvent.COIN_RESPONSE_EVENTS, onCoinResponseSuccess);
        _pokktExtension.addListener(PokktEvent.COIN_RESPONSE_FAILED_EVENTS,
onCoinResponseFailed);
        _pokktExtension.addListener(PokktEvent.COIN_RESPONSE_WITH_TRANS_ID_EVENTS,
onCoinResponseWithTransID);
        _pokktExtension.addListener(PokktEvent.VIDEO_CLOSED_EVENTS, onVideoClosed);
        _pokktExtension.addListener(PokktEvent.VIDEO_COMPLETED_EVENTS, onVideoCompleted);
        _pokktExtension.addListener(PokktEvent.VIDEO_SKIPPED_EVENTS, onVideoSkipped);
        _pokktExtension.addListener(PokktEvent.VIDEO_DISPLAYED_EVENTS, onVideoDisplayed);
        _pokktExtension.addListener(PokktEvent.VIDEO_GRATIFIED_EVENTS, onVideoGratified);
//-----

```

Showing Video Ad:

If you are using **Sans Auto Cache Video** option then need to pass params like below:

```

_pokktManager.setParams(securityKey:String, applicationID:String, userID:String,
                        integrationType:String, autoCache:String);

```

Here **autoCache will be false** so that caching will not start. And you need to call below method when you want to start caching video.

```

_pokktExtension.performOperation(PokktOperation.START_CACHING_VIDEO);

```

You can use this setParams method for Auto Caching also but there autoCaching will be true.

But need to set this params before calling _pokktExtension.initPokkt();

Or otherwise no need to pass anything.

Here call the performOperation method by passing type of operation you want to do. So we are showing video here so passing PokktOperation.GET_VIDEO.

This method will give you some incentive after completing video.

```

_pokktExtension.performOperation(PokktOperation.GET_VIDEO);

```

This method will not give any incentive after completing video.

```

_pokktExtension.performOperation(PokktOperation.GET_VIDEO_NON_INCENT);

```

This method will not give any incentive after completing video and will take to other Ad activity.

```

_pokktExtension.performOperation(PokktOperation.GET_VIDEO_WITH_INTENT);

```

These all methods will show the Video on screen but video has to be cached locally then only it will show. So when we instantiate the PokktASExtension at that time it start caching video locally. So need to check, video has been cached or not by calling isVideoAvailable method. This will return true if video has been cached otherwise will return false.

```

_pokktExtension.isVideoAvailablePokkt();

```

If user gets incentive then VIDEO_GRATIFIED_EVENTS will be trigger. For videos there are few callback which gets call according to situation. Like if user skip the video then VIDEO_SKIPPED_EVENTS will be fire. If user close the video then VIDEO_CLOSED_EVENTS will be fire. If user complete the video then VIDEO_COMPLETED_EVENTS will be fire.

Also there is option to disable SKIP button from Video Ad. So for this you can call below method with

passing timer value.

```
_pokktExtension.setDefaultSkipTime(10); // Here value 10 is 10 Seconds. So after 10 seconds  
you will see the SKIP button on Video Ad.
```

If due to some reason if video download fails then `DOWNLOAD_FAILED_EVENTS` will be fired otherwise if video download gets succeed then `DOWNLOAD_COMPLETED_EVENTS` will be fired.

Showing OfferWall:

This method will show the OfferWall from where user can download the app and they will get the some incentive but it has to match with given terms and condition.

```
_pokktExtension.performOperation(PokktOperation.GET_FREE_COINS);
```

And if user gets incentive then `COIN_RESPONSE_WITH_TRANS_ID_EVENTS` or `COIN_RESPONSE_EVENTS` will get called.

`COIN_RESPONSE_WITH_TRANS_ID_EVENTS` **will give you Transaction ID and Earned Points.**

`COIN_RESPONSE_EVENTS` **will give you only Earned Points only.**

`COIN_RESPONSE_FAILED_EVENTS` **will be called if there is no pending Earned Points.**

And if you want to check pending earned points then you can check by calling below method.

```
_pokktExtension.performOperation(PokktOperation.GET_PENDING_COINS);
```

Sometimes you will not get coins immediately so you will have to call above method.

And there is option to pass fixed asset value for getting offerwall campaign.

```
_pokktExtension.performOperation(PokktOperation.GET_FIXED_FREE_COINS_FROM_POKKT,  
                                value);
```

And also you can check is there any campaign available or not for the particular user. It may happen if that user has downloaded all app from offerwall.

Important points:

You need to call pending transaction on your app resume. So you can use `NativeApplication` for this. Add the `Event.Activate` with the `NativeApplication`.

```
NativeApplication.nativeApplication.addListener(Event.ACTIVATE, onActivate, false, 0,  
true);
```

And here is the `onActivate` handler for this.

```
protected function onActivate(event:Event):void  
{  
    if(_pokktExtension)  
        _pokktExtension.performOperation(PokktOperation.CHECK_TRANSACTION);  
}
```

For showing Log and showing Toast on screen:

If you want to track log in logcat then you can use `showLogs` method. But before that you will have to enable it in debug mode by calling below method. So call `setDebug` method after initializing `PokktExtension` and then use `showLogs` wherever you want to use.

```
_pokktExtension.setDebug(true);  
_pokktExtension.showLogs("log msg here");
```

If you want to show Toast on screen then you can call below method.

```
_pokktExtension.showToast("Showing toast msg here");
```