v4.1.0 Monday, 21 March 2016

Pokkt Android SDK Integration Guide

Please follow these steps as per your integration requirement(Video/OfferWall/Both).

Configuration Steps

1. Add the PokktSDK_v4.1.0.jar or PokktSDK_v4.1.0.aar to your project.x
2. If you are using pokkt jar then, extract the contents of res.zip folder and copy them to respective folders in your project.
3. Please add android support v4 library in your project for Pokkt In-App Notification.
4. Android MinSDKVersion should be >= 14.
5. Add following permissions in your AndroidManifest, If not already there.

```
<!-- These permissions are mandatory to run Pokkt SDK -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!-- These permissions are strongly recommended and will result in higher performance -->
<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" />
<!-- This permission is optional  but will improve SDK feature-->
<uses-permission android:name="android.permission.WRITE_CALENDAR" />
```

6. Add the following activity in your AndroidManifest for OfferWall Integration

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

7. Add the following activity in your AndroidManifest for Pokkt Interstitial Integration

```
<activity
        android:name="com.app.pokktsdk.InterstitialActivity"
```

```
android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|
smallestScreenSize"

android:label="@string/app_name"

android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified" >

</activity>
```

8. Add the following activity in your AndroidManifest for Video Integration

```
<activity

android:name="com.app.pokktsdk.PlayVideoCampaignActivity"

android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenLayout|uiMode|screenSize|
smallestScreenSize"

android:label="@string/app_name"

android:screenOrientation="landscape"

android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified" >

</activity>
```

9. Add Following Broadcast receiver for OfferWall Integration in AndroidManifest

```
<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>
```

10. Add Following meta tag for google play services (Goole play services is required and should be part of your project, if not, please refer http://developer.android.com/google/play-services/setup.html)

```
<meta-data android:name="com.google.android.gms.version" android:value="@integer/
google_play_services_version" />
```

11. Add a meta data tag for OfferwallCampaignDelegate Implementation class, you will have to implement the OfferwallCampaignDelegate interface in your project to listen for all offerwall related events. The implementation must have a default no args constructor. (refer to sample app for example)

```
<meta-data android:name="offerwallDelegate" android:value="<fully qualified name of your implementation class>" />
```

12. Add following Service and receiver in manifest for google analytics (Optional).

```
<receiver android:name="com.google.android.gms.analytics.AnalyticsReceiver" android:enabled="true">

<intent-filter>
```

```
                <action android:name="com.google.android.gms.analytics.ANALYTICS_DISPATCH" />
            </intent-filter>
        </receiver>
        <service android:name=“com.google.android.gms.analytics.AnalyticsService"
            android:enabled=“true"
            android:exported=“false"/>
```

13. Add following Service in manifest for in-app notification.

```
        <service android:name=“com.app.pokktsdk.notification.NotificationService"
            android:label=“PokktNotificationService”
            android:exported="false"/>
```

Implementation Steps

- ### Common

1. For all invocation of Pokkt SDK developer will make use of methods available in *PokktManager* class.This class only have static methods.
2. Before calling any other methods from the *PokktManager* please make sure that you have called the *initPokkt* already.(This does not apply to few methods namely *startSession* , *endSession and setAdDelegate*).
3. For pokkt initialisation, *PokktConfig* instance is required. *PokktConfig* is plain old java object which will hold all the values required by the SDK which you need to assign.
4. For almost all ad related methods call, *AdConfig* instance is required. *AdConfig* is plain old java object which will hold all the values required by the SDK which you need to assign for getting an Ad.
5. Developer should call ad related method APIs by passing adConfig which must contain screen name and incentive option.
6. In *AdConfig* you can assign *screenName*, *isRewarded, shouldAllowSkip, defaultSkipTime, skipConfirmMessage, backButtonDisabled, shouldAllowMute, shouldSkipConfirm, skipConfirmYesLabel, skipConfirmNoLabel, skipTimerMessage* and *incentiveMessage* . These values can be used to configure the behaviour of ad.
7. In *PokktConfig* you can assign *applicationId*, *securityKey* and *IntegrationType* which are must for all type of integrations.

8. You will have to implement the AdCampaignDelegate interface in your project to listen for all ad related events. Please call *PokktManager.setAdDelegate(adCampaignDelegate)* to register the AdCampaignDelegate implementation with the pokkt SDK.
9. If you are doing server to server integration with pokkt you can also mention *thirdPartyUserId* in *PokktConfig*.
10. Apart from above mentioned parameters you can assign additional ones based on your integration type.(please refer to OfferWall and Video sections below.)
11. While in development please call *PokktManager.setDebug(context,true);* to see pokkt debug logs and toast messages. please make sure to change this to *PokktManager.setDebug(context,false);* for production build.
12. Please call *PokktManager.notifyAppInstall(context)* to send your application installation information to Pokkt.
13. Please call *PokktManager.trackIAP(context, InAppPurchaseDetail)* to send any in-app purchase information to Pokkt.
14. To use google analytics, please set AnalyticsType and Analytics ID in *PokktConfig.*

    setSelectedAnalyticsType(AnalyticsType.GOOGLE_ANALYTICS)
    setGoogleAnalyticsID().
15. To use flurry analytics please set AnalyticsType and Flurry Application Key in *PokktConfig.*

    setSelectedAnalyticsType(AnalyticsType.FLURRY)
    setFlurryApplicationKey() in *PokktConfig*.
16. To use mix panel analytics please set AnalyticsType and Mix PanelProject Token in *PokktConfig.*

    setSelectedAnalyticsType(AnalyticsType.MIXPANEL)
    setMixPanelProjectToken().
17. To use fabrics number analytics please set AnalyticsType and Mix PanelProject Token in *PokktConfig.*

    setSelectedAnalyticsType(AnalyticsType.FABRIC)

- Session

1. Starting with this version Pokkt SDK is adding session tracking for which we have *startSession* and *endSession* methods in *PokktManager.*
2. You should call *startSession* at the start of his application and once only.
3. You should call *endSession* at the end of his application and once only.

- OfferWall

1. In *PokktConfig* for OfferWall you can set two additional parameters which are *offerWallAssetValue* and *closeOnSuccessFlag. offerWallAssetValue* is required if you only want to show campaigns of certain value on OfferWall. It has default value as empty. *closeOnSuccessFlag* is required if you wish to close the OfferWall after user has completed one offer. It's default value is false.
2. Before calling another method for offerWall in *PokktManager*, please make sure that you have already called *pokktInit* first.
3. You will need to create *OfferwallCampaignDelegate* implementation class as mentioned in step 8 in configuration steps.
4. To show OfferWall you can call *PokktManager.getCoins(context, pokktConfig);*
5. In the screen or activity where you have button to show offer wall, in that activity onResume you should call *PokktManager.getPendingCoins(context);* so that you get a callback to award points to the user after he has come back to your game after finishing with OfferWall. You will get a callback for this call in your *OfferwallCampaignDelegate* implementation class in method *earnedCoins* or *requestFailed*
6. You can call *PokktManager.checkCampaignAvailable(context, pokktConfig);* to check whether the campaigns are available before showing OfferWall button to user. You will get a callback for this call in your *OfferwallCampaignDelegate* implementation class in method *onOfferwallCampaignCheck.*

AdConfig

1. In *AdConfig* you should set *screenName* and *isRewarded.* This screen name will be created on pokkt dashboard.

2. In *AdConfig* , developer can also set *shouldAllowSkip, defaultSkipTime, skipConfirmMessage, backButtonDisabled, shouldAllowMute, shouldSkipConfirm, skipConfirmYesLabel, skipConfirmNoLabel, skipTimerMessage* and *incentiveMessage* . These values can be used to configure the behaviour of ad.
3. If you want to enable/disable the skip button on video screen please set *shouldAllowSkip* as true/false.The default value for *shouldAllowSkip* is true.
4. If you have enabled skipped button by setting *shouldAllowSkip* as true then you can control after how many seconds the skip button will be visible in video by setting *defaultSkipTime* to appropriate value.Since most videos will be 30 sec or less please set *defaultSkipTime* as 10 or less.You can also give your own skip message by setting *skipConfirmMessage* on *AdConfig*
5. The *screenName* has default value as *default* and can be used by you to give different screen name for different places in your app where you are showing ads. You will control ad targeting based on these screen names which should match exactly with screen names defined in dashboard. ScreenName can not contain white spaces and only special characters allowed are hyphen and underscore.
6. You can choose to show ad with or without incentive to user by setting *isRewarded* as true or false. Video gratification will only happen for incentivised playback.
7. You can disable the back button while video is playing by setting *backButtonDisabled* on *AdConfig*.
8. You can configure the ad skip dialog yes/no labels by setting *skipConfirmYesLabel* and *skipConfirmNoLabel.*
9. You can configure the ad incentive message by setting *incentiveMessage.*
10. You can configure the ad skip timer message by setting *skipTimerMessage.* The message must contain a ## placeholder to show skip time value, which will keep changing as per the time.

Rewarded/Non Rewarded Ad

1. *Y*ou will have to call *PokktManager.cacheAd(context, adConfig);* to start caching ads on device.
2. You will need to create *AdCampaignDelegate* implementation class as mentioned in step 7 in implementation steps.

3. You can call *PokktManager.checkAdAvailability(context,adConfig)* to check if the campaign are available for a particular adConfig before you try to show ad.
4. You can call *PokktManager.showAd(context, adConfig);* to show ad.
5. You will get different callbacks as given in *AdCampaignDelegate* implementation class for ad display.
6. Please reward user only from the *onAdGratified* method in *AdCampaignDelegate* implementation class.

Export Logs

- Developer should call *Logger.exportLog(Activity)* to export the Pokkt SDK logs to some folder.
- This API shows a folder chooser dialog where user can choose a particular folder.
- User can also create a new folder where user wants to export the logs.

Optional Parameters

- *PokktConfig* also has provision for developers to provide extra user data available with them to pokkt.We currently support following data points: *name, age, sex, mobileNo, emailAddress, location, birthday, maritalStatus, facebookId, twitterHandle, education, nationality, employment and maturityRating.*

Mediation Info

- Pokkt SDK now supports 12 ad networks which you can integrate in your application for better monetisation.

- To integrate these networks through Pokkt, please visit the mediation menu on downloads page and download *native mediation zip* and *documentation zip* files.
- Please follow the mediation integration documents shipped for each network.
- In *native mediation zip* we have provided sample application project as an example integrating all these networks.
- You will need to create account on these networks and add the network details in your Pokkt dashboard after login into your account on pokkt website.
- You will also need to do the mapping of Pokkt screens with the corresponding ad networks' placement id/zone id/ad unit etc in the dashboard.
- We recommend passing activity context to PokktManager for caching and showing ads because many ad mediation networks require activity context only to work correctly.

Important Points

- Please do not copy the code points from this pdf as it may introduce unwanted characters and space in your code.instead please refer to sample app source code in pokkt bundle.
- Please also refer to sample app source code for better understanding of implementation.