v5.0 Monday, 3 October 2016

Pokkt Android SDK Integration Guide

Please follow these steps as per your integration requirement.

Configuration Steps

1. Add the PokktSDK_v5.0.jar or PokkSDK_v5.0.aar to your project.
2. If you are using pokkt jar then, extract the contents of res.zip folder and copy them to respective folders in your project. Please copy assets folder data to your application assets folder.
3. Please add android support v4 library in your project for Pokkt In-App Notification.
4. Android MinSDKVersion should be >= 11.
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" />
<!-- These permissions are optional  but will improve SDK feature-->
<uses-permission android:name="android.permission.WRITE_CALENDAR" />
 <uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
```

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"/ >
```

7. 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”

android:label=“@string/app_name”

android:screenOrientation=“landscape”

android:windowSoftInputMode=“stateAlwaysHidden|adjustUnspecified"/>
```

8. Add the following activity in AndroidManifest for VPAID ads.

```
<activity

android:name=“com.app.pokktsdk.VPAIDActivity”

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

android:label=“@string/app_name”

android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified"/>
```

9. Add the following activity in AndroidManifest for interstitial ads.

```
<activity

android:name=“com.app.pokktsdk.PokktInterstitialActivity"

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

android:label="@string/app_name"

android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified"/>
```

10. 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>
```

11. 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" />
```

12. 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>" />
```

13. 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"/>
```

14. 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, please make use of methods available in *PokktManager* class.This class only have static methods.
2. You will have to implement the *AdDelegate* interface in your project to listen for all video ad and interstitial ad related events.
3. Please call *PokktManager.setAdDelegate(adDelegate)* to register the AdCampaignDelegate implementation with the pokkt SDK.
4. You can implement the *BannerDelegate* interface in your project to listen for all banner related events.
5. Please call *PokktManager.setBannerDelegate(bannerDelegate)* to register the BannerDelegate implementation with the pokkt SDK.
6. You can implement *PokktInitDelegate* interface in your project to listen for init related event.
7. Please call *PokktManager.setInitDelegate(PokktInitDelegate)* to register the PokktInitDelegate implementation with the pokkt SDK.
8. 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 , setAdDelegate , setBannerDelegate and setInitDelegate*).
9. 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.
10. In *PokktConfig* you can assign *applicationId*, *securityKey* and *IntegrationType* which are must for all type of integrations.
11. If you are doing server to server integration with pokkt you can also mention *thirdPartyUserId* in *PokktConfig*.
12. 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.
13. You should call ad related method APIs by passing adConfig which must contain screen name and incentive option.

14. In *AdConfig* you can assign *screenName*, *isRewarded, adFormat, shouldAllowSkip, defaultSkipTime, skipConfirmMessage, backButtonDisabled, shouldAllowMute, shouldSkipConfirm, skipConfirmYesLabel, skipConfirmNoLabel, skipTimerMessage* and *incentiveMessage* . These values can be used to configure the behaviour of ad.
15. Apart from above mentioned parameters you can assign additional ones based on your integration type.(please refer to OfferWall and Ad sections below.)
16. 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.
17. Please call *PokktManager.notifyAppInstall(context)* to send your application installation information to Pokkt.
18. Please call *PokktManager.trackIAP(context, InAppPurchaseDetail)* to send any in-app purchase information to Pokkt.

- AdConfig

1. In *AdConfig* you must set *screenName* and *isRewarded.* This screen name will be created on pokkt dashboard.
2. In *AdConfig* , you can also set *adFormat, shouldAllowSkip, defaultSkipTime, skipConfirmMessage, backButtonDisabled, shouldAllowMute, shouldSkipConfirm, skipConfirmYesLabel, skipConfirmNoLabel, skipTimerMessage* and *incentiveMessage* . These values can be used to configure the behaviour of ad.
3. For getting specific type of ad set *adFormat* in adConfig. AdFormat is a type for ad such as 0:video, 1: banner, 3: interstitial , default is 0
4. 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.
5. 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*
6. *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.
7. 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.
8. You can disable the back button while video is playing by setting *backButtonDisabled* on *AdConfig*.
9. You can configure the ad skip dialog yes/no labels by setting *skipConfirmYesLabel* and *skipConfirmNoLabel.*
10. You can configure the ad incentive message by setting *incentiveMessage.*
11. 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.

- Analytics

1. To use google analytics, please set AnalyticsType and Analytics ID in *PokktConfig.*

   setSelectedAnalyticsType(AnalyticsType.GOOGLE_ANALYTICS)
   setGoogleAnalyticsID().

2. To use flurry analytics please set AnalyticsType and Flurry Application Key in *PokktConfig.*

   setSelectedAnalyticsType(AnalyticsType.FLURRY)
   setFlurryApplicationKey() in *PokktConfig*.

3. To use mix panel analytics please set AnalyticsType and Mix PanelProject Token in *PokktConfig.*

   setSelectedAnalyticsType(AnalyticsType.MIXPANEL)
   setMixPanelProjectToken().

4. To use fabrics number analytics please set AnalyticsType and Mix PanelProject Token in *PokktConfig.*

   setSelectedAnalyticsType(AnalyticsType.FABRIC)

- Session

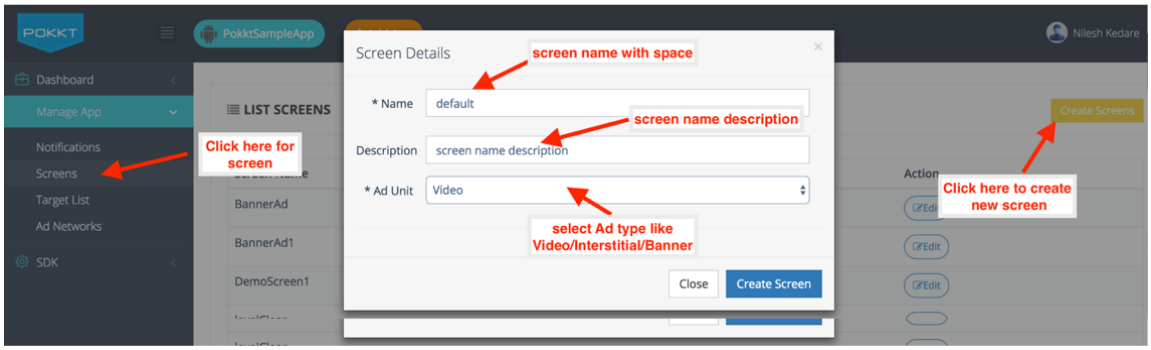
1. Starting with this version Pokkt SDK is adding session tracking for which we have *startSession* and *endSession* methods in *PokktManager.*
2. You can call *startSession* at the start of his application and once only.
3. You can call *endSession* at the end of his application and once only.

- Ad (Video and Interstitial)

1. You will need to create ***AdDelegate*** implementation class as mentioned in step 3 in implementation steps.
2. *Y*ou will have to call ***PokktManager.cacheAd(context, adConfig)****;* to start caching ads on device.
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 *AdDelegate* implementation class for ad display.
6. Please reward user only from the *onAdGratified* method in *AdDelegate* implementation class.

- Banner

1. You will need to create ***BannerDelegate*** implementation class as mentioned in step 4 in implementation steps.
2. You will have to call ***PokktManager.loadBanner(context, screenName, container);*** to show banner on device.
3. You can call ***PokktManager.showAd(context, adConfig);*** to show ad.
4. You will get different callbacks as given in *BannerDelegate* implementation class for banner display. BannerDelegate has two delegate methods which you need to implement in your class.*onBannerLoaded(String screenName);* will get triggered when banner gets loaded for that screen name. *onBannerLoadFailed(String screenName, String errorMsg);* will get triggered when banner load failed for that screen name.
5. ***Screen name*** : For screen name, you will have to create in Pokkt dashboard. Please check below screen. This will help you to understand how to create new scree name in Pokkt dashboard.



6. Once you come out from Activity then you need to call destroy method ***PokktManager.destroyBanner(container)***. This is mandatory to do. Pass same container object for that you have loaded.
7. You need to resume banner once onResume() method get triggered by calling **PokktManager.onResumeBanner(activity, ScreenName).** This is mandatory to do. You need to pass activity context where you are showing current banner. You need to pass screen name for that you have called loadBanner last time.
8. Banner auto refreshes on given time interval. This interval you give when you create screen name for banner in Pokkt dashboard. This interval varies in-between 30-120 seconds. If you want to disable auto refresh from code then ***PokktManager.setBannerAutoRefresh(false);*** should be called. Default value for auto refresh is "true" but if you want to disable then pass “false”.

9. ***Banner Ad Size***: Banner size can be any size. It is not fixed. It depends on what size ad you are looking for. So size you can define in your layout xml file and also you need to create screen name for that particular size in Pokkt dashboard. For creating screen name in Pokkt dashboard, we have already explained above. Please check that. Banner size must match both places (layout xml file and screen name in pokkt dashboard).
10. Container : For container, you need to pass container view group object like Frame Layout / Relative Layout/ Linear Layout object from your layout file to ***PokktManager.loadBanner(this, default, container)*** method.

- 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.*

- Export Logs

    1. You can call *Logger.exportLog(Activity)* to export the Pokkt SDK logs to some folder.
    2. This API shows a folder chooser dialog where user can choose a particular folder.
    3. User can also create a new folder where user wants to export the logs.
    4. You can also call *Logger.exportLogToCloud(Activity)* to export the Pokkt SDK logs to cloud or mail.

- Optional Parameters

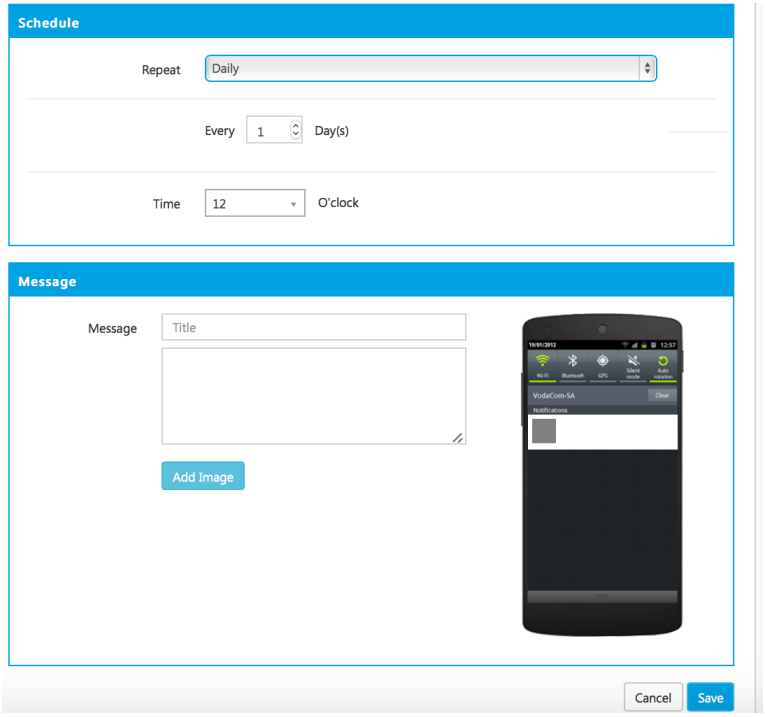
    - *PokktConfig* also has provision for developers to provide extra user data available with them to pokkt.We currently support following data points: *name, age, sex, mobileNo, emailAddress, location, birthday, maritalStatus, facebookId, twitterHandle, education, nationality, employment and maturityRating.*

In-App Notifications

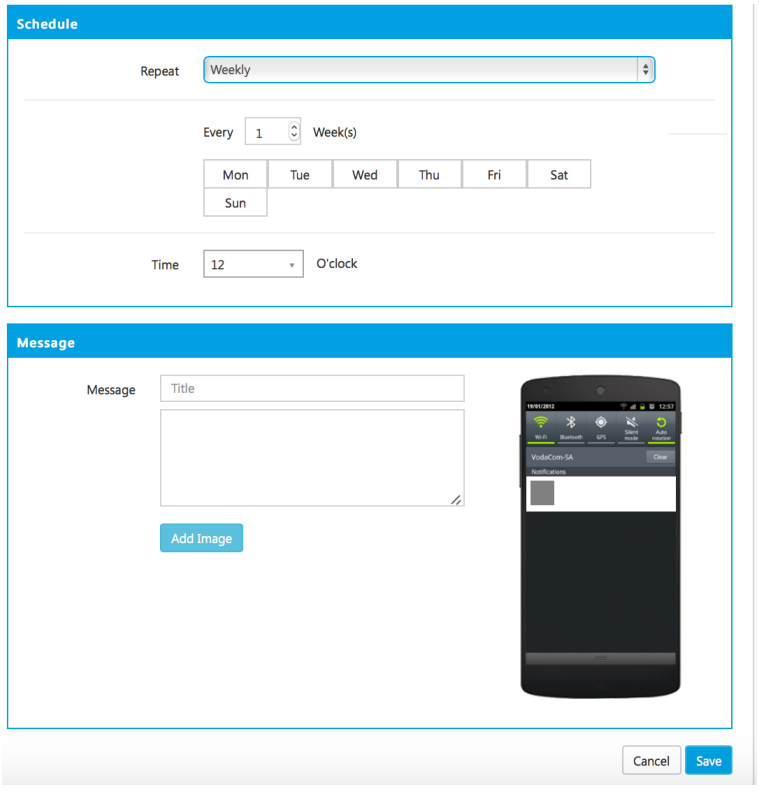
You can add In-App notifications in your dashboard.

Repeat schedule can be daily, weekly monthly.

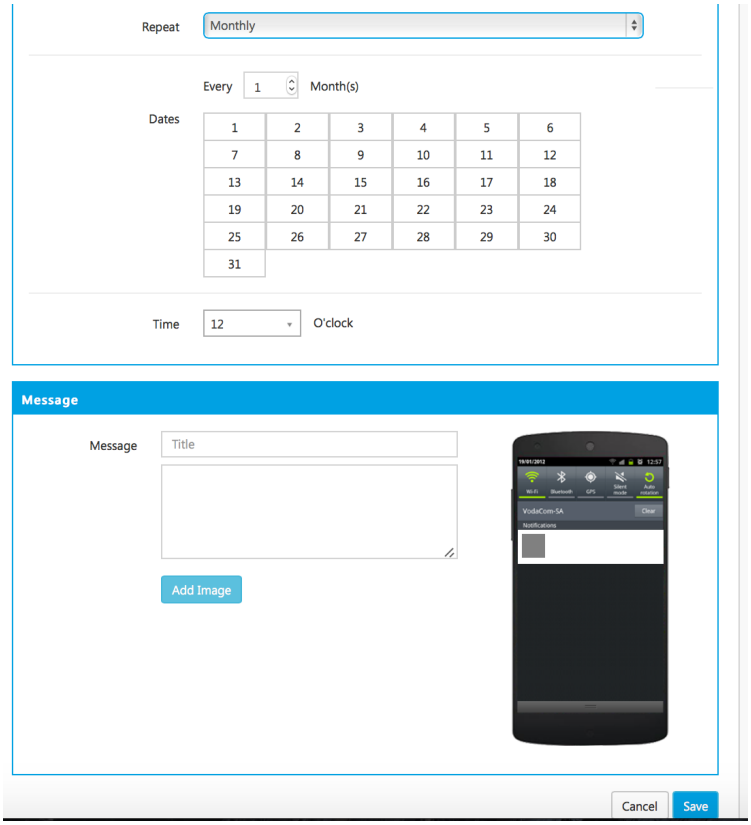
Daily Repeat can have options like frequency of repeat and time in hours of notification.



Weekly repeat can have options like frequency of repeat in weeks, days of repeat and time in hours of notification.



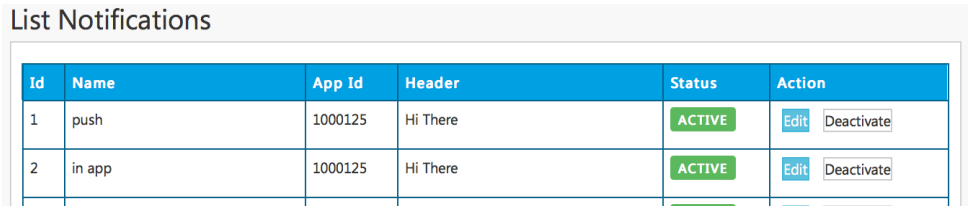
Monthly repeat can have options like frequency of repeat in months dates of repeat and time in hours for notification.



For don't repeat case, there are options like dates and time in hour for notification.

The notifications are listed and can be edited.

Notifications can also be deactivated/activated.



Mediation

Pokkt SDK now supports many ad networks which you can integrate in you application for a better monetisation and better user engagement.

You can integrate the Pokkt Supported Mediation networks as mentioned above to fetch the ads from these networks.

To integrate these networks through Pokkt, please follow the mediation integration documents shipped with the release.

Sample code is also provided in release for each and every ad network.

You need to create account on these networks and add the network details in Pokkt dashboard.

You must do the mapping of Pokkt screen name with the corresponding ad placement or ad id of the mediated network.

We recommend using activity context for caching and showing ads, because many Ad Mediation networks expect activity to for their working.

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.