v4.0 Wednesday, 16 December 2015

Pokkt Android SDK Integration Guide

Overview

Pokkt Android SDK v4.0 has following extra features:

1. Ad Network mediation support for following networks.
    I. AdColony.
    II. AppLovin.
    III. Chartboost.
    IV. Fuse Powered.
    V. Fyber.
    VI. InMobi.
    VII. MoPub.
    VIII. Supersonic.
    IX. Tapjoy.
    X. Unity3d.
    XI. Vungle
    XII. Vdopia

2. Automatic retry of video caching after a fixed duration, if video caching from all the networks fail. Developer can set this duration. Please call pokktconfig setRetryDuration to set this value.
3. Pokkt videos ads now have pokkt branding text.
4. Pokkt video player has mute/un-mute button now.
5. **Android permissions like write to external storage, access power wake lock and read phone state are made optional but highly recommended. Please update your Manifest accordingly.**
6. Survey can be prefetched so that user need not wait for loading a survey.
7. Timeout applied for video caching for mediation ad networks.

8. Fabric/Numbers analytics support.
9. Minor Bug fixes.
10. **PlayVideoCampaignActivity has updated property in Android Manifest. Please update your Manifest accordingly.**
11. **PokktManager.setDebug api signature is changed. Please change your code accordingly.**
12. Unused resources removed. strings.xml file now not required.

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

Configuration Steps

1. Add the PokktSDK_v4.0.jar to your project's lib directory.
2. Please add android support v4 library in your project for Pokkt In-App Notification.
3. Please add moat.jar to your project's lib directory.
4. Extract the contents of res.zip folder and copy them to respective folders in your project.
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 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" >
</activity>
```

8. 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>
```

9. 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" />
```

10. 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>" />
```

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

```
<meta-data android:name="videoDelegate" 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 session related methods namely *startSession* and *endSession*)
3. For almost all methods call, *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. In *PokktConfig* you can assign *applicationId*, *securityKey* and *IntegrationType* which are must for all type of integrations.
5. If you are doing server to server integration with pokkt you can also mention *thirdPartyUserId* in *PokktConfig*.
6. Apart from above mentioned parameters you can assign additional ones based on your integration type.(please refer to OfferWall and Video sections below.)
7. 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.

8. Please call *PokktManager.sendAppInfo(context, pokktConfig)* to send your application installation information to Pokkt.
9. Android MinSDKVersion should be >= 9.
10. In PokktConfig , you can also set the video caching retry duration when all the networks caching fails. Video caching will start again after this duration. Please call *setRetryDuration(duration)* to set this value. This duration is in seconds.
11. To use google analytics, please set AnalyticsType and Analytics ID in *PokktConfig.*

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

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

    setSelectedAnalyticsType(AnalyticsType.MIXPANEL)
    setMixPanelProjectToken().
14. 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. You will need to provide pokktConfig instance for this method with *applicationId*, *securityKey* and *IntegrationType* assigned.
    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.*

Video

1. In *PokktConfig* for Video you can set five additional parameters which are *autoCacheVideo,skipEnabled,defaultSkipTime,screenName* and *incentivised.*
2. *autoCacheVideo* is required if you want to automatically cache video on user device. It has default value as true. if you set it as false then video will not be automatically cached and you will have to call *PokktManager.cacheVideoCampaign(context, pokktConfig);* to start caching videos on device.
3. If you want to enable/disable the skip button on video screen please set *skipEnabled* as true/false.The default value for *skipEnabled* is false.
4. If you have enabled skipped button by setting *skipEnabled* 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 *customSkipMessage* on *PokktConfig*

5. *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 video 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 video with or without incentive to user by setting *incentivised* 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 *PokktConfig*.
8. You will need to create *VideoCampaignDelegate* implementation class as mentioned in step 9 in configuration steps.
9. You can call *PokktManager.isVideoAvailable()* to check if the campaign are available before you try to play video.
10. You can call *PokktManager.playVideoCampaign(context, pokktConfig);* to play video.
11. You will get different callbacks as given in *VideoCampaignDelegate* implementation class for video playback.
12. Please reward user only from the *onVideoGratified* method in *VideoCampaignDelegate* implementation class.

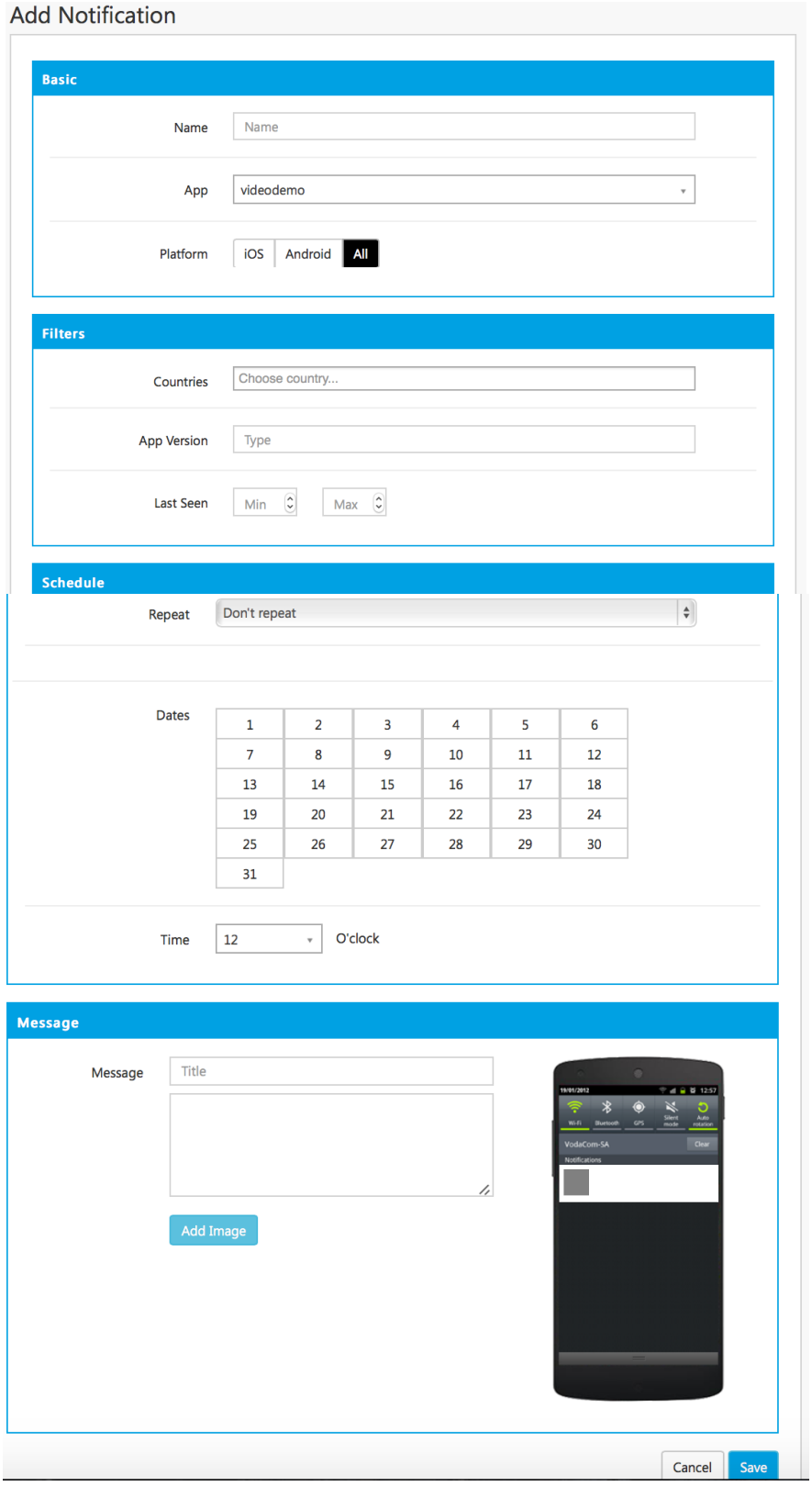
Export Logs

1. Developer should call *Logger.exportLog(context)* 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.

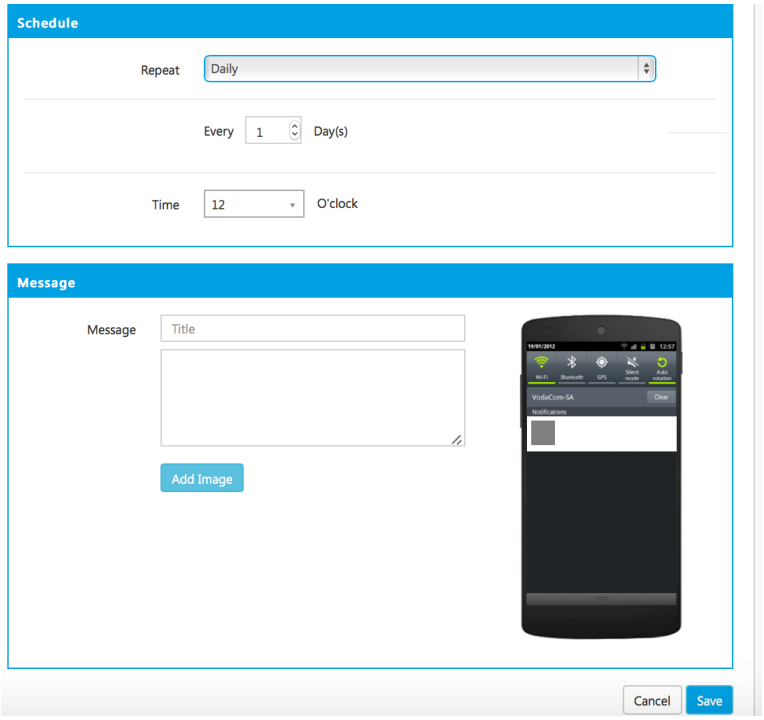
In-App Notifications

Developer can add In-App notifications in their dashboard.

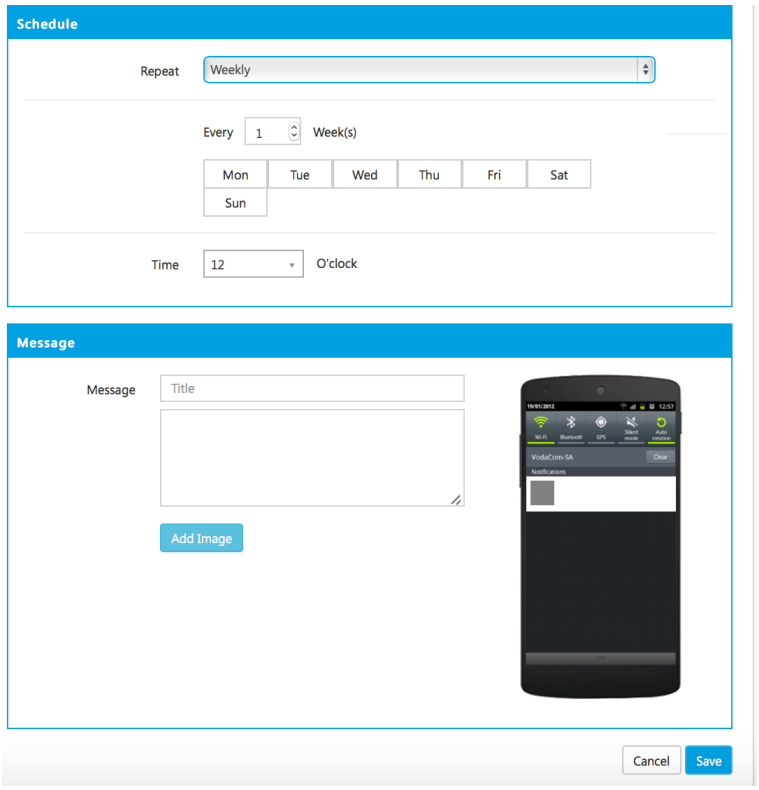
Repeat schedule can be daily, weekly monthly.



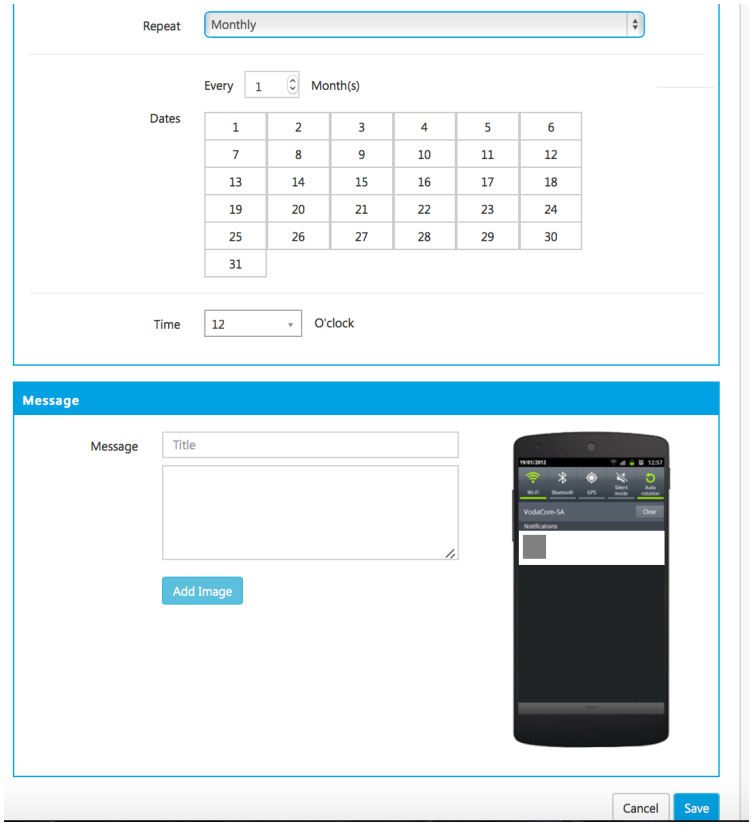
Daily Repeat can have options like frequency of repeat and time in hours of notification.



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 developer can integrate in their application for a better monetisation and better user engagement.

Developer 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.

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

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.*

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.