



# POKKT SDK v5.0 Integration Guide for Xamarin (Android)

## Contents:

---

1. Overview
2. Configuration steps
3. Implementation steps
4. Important Points

### 1. Overview

Thank you for choosing **Pokkt SDK Plugin v5.0** for **Xamarin**. Pokkt SDK supports Reward Ad, Non-Reward Ad, Banner Ad and Offerwal campaigns feature. This document contains all the information which is needed to setup the SDK with project. Please follow these steps as per your integration requirement (Reward/Non-Reward/Banner/Offerwal). The current plugin supports mediation for various third party ad-networks. These are:

1. AdColony
2. AppLovin
3. Chartboost
4. Fyber
5. InMobi
6. Supersonic
7. UnityAds
8. Tapjoy
9. Vungle
10. AdMob
11. Facebook

A separate set of documents is provided for each of these, explaining the implementation process.

There is a sample app provided with the SDK. We will be referencing this app during the course of explanation in this document. It is suggested that you should check that app to understand the following

process in detail.

## 2. Configuration steps

All we need is the file provided: **PluginExtension.zip**. This zip file contains three files which are **PokktExtension.dll**, **PokktExtension.Droid.dll** and **res.zip**.

1. Add the **PokktExtension.dll** and **PokktExtension.Droid.dll** to your project's Reference directory.
2. Copy the contents of **res** folder to respective folders in your project.
3. Add following permissions in your **AndroidManifest** xml file, If not already there.

### Android Manifest xml update:

1. Add following permissions in your App xml under android manifest, 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" />
```

2. 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>
```

3. Add the following activity in your **AndroidManifest** for Video Integration.

```

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

```

#### 4. Add the following activity in AndroidManifest for VPAID ads.

```

<activity
    android:name="com.app.pokktsdk.VPAIDActivity"
    android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenLa
yout|uiMode|screenSize|smallestScreenSize"
    android:label="@string/app_name"
    android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified"/>

```

#### 5. Add the following activity in AndroidManifest for interstitial ads.

```

<activity
    android:name="com.app.pokktsdk. PokktInterstitialActivity"
    android:configChanges="keyboard|keyboardHidden|navigation|orientation|screenLa
yout|uiMode|screenSize|smallestScreenSize"
    android:label="@string/app_name"
    android:windowSoftInputMode="stateAlwaysHidden|adjustUnspecified"/>

```

#### 6. 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>

```

#### 7. Add Following meta tag for google play services (Goole play services is required and should be part of your project, if not, please add the GooglePlayService component in project).

```

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

```

8. 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.

```
<meta-data
    android:name="offerwallDelegate
    android:value="<com.pokkt.plugin.common.PokktOfferwallDelegate>" />
```

9. 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"/>
```

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

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

////////////////////////////////////

**Note:** Please do not copy the code points from this PDF file as it may introduce unwanted characters and space in your code. Instead please refer to sample app source code provided with the sample app.

## 3. 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. You need to set extension before calling any method like below. This is **mandatory** to do it.

```
PokktManager.SetNativeExtentions(new AndroidExtension(this));
```

3. In **PokktConfig** you can set ApplicationId and SecurityKey which are must for all type of integrations. Please check the sample app.

```
pokktConfig.ApplicationId = "<Pokkt Application ID>";  
pokktConfig.SecurityKey = "<Pokkt Security Key>";
```

4. Before calling any other methods from the PokktManager, please make sure that you have called the **InitPokkt** with passing PokktConfig object.

```
PokktManager.InitPokkt(pokktConfig);
```

5. If you are doing server to server integration with POKKT you can also set **ThirdPartyUserId** in PokktConfig.

```
pokktConfig.ThirdPartyUserId = "<Third party user Id>";
```

6. Apart from above mentioned parameters you can assign additional ones based on your integration type (please refer to OfferWall and Reward sections below).
7. While in development, please call setDebug method to see Pokkt logs and toast messages. Please make sure to change this to setDebug to false for production build.

```
PokktManager.setDebug(true/false);
```

8. Android **MinSDKVersion** should be **>= 14**.
9. To use **Google Analytics**, please set AnalyticsType and Analytics ID in

```
pokktConfig.selectedAnalyticsType = AnalyticsType.GOOGLE_ANALYTICS;  
pokktConfig.googleAnalyticsID = "Id";
```

10. To use **Flurry Analytics** please set AnalyticsType and Flurry Application Key in PokktConfig.

```
pokktConfig.selectedAnalyticsType = AnalyticsType.FLURRY;  
pokktConfig.flurryApplicationKey = "key";
```

11. To use **Mix Panel Analytics** please set AnalyticsType and Mix PanelProject Token in PokktConfig:

```
pokktConfig.selectedAnalyticsType = AnalyticsType.FLURRY;  
pokktConfig.mixPanelProjectToken = "token";
```

12. To use **Fabric Analytics** please set AnalyticsType and Fabric Token in PokktConfig

```
pokktConfig.selectedAnalyticsType = AnalyticsType.FABRIC
```

13. Please call `trackIAP(InAppPurchaseDetail)` to send any in-app purchase information to Pokkt. Like below you can do this.

```
// create object of InAppPurchaseDetails and set properties for purchaseDetails and pass object  
PokktManager.trackIAP (InAppPurchaseDetails);
```

## Session

1. We have option to start session and end session for tracking in PokktManager.

```
PokktManager.startSession();  
PokktManager.endSession();
```

2. You should call `startSession` at the start of his application if you want to use this but this is the optional and call it after setting application id and security key.
3. You should call `endSession` at the end of his application.

## Offerwall

1. In PokktConfig for OfferWall, you can set two additional parameters which are `OfferWallAssetValue` and `closeOnSuccessFlag`.

**OfferWallAssetValue** is only required if you only want to show campaign of certain value on offerwall.

**closeOnSuccessFlag** is required if you wish to close the OfferWall after user has completed one offer. It's default value false.

1. Before calling another method for offerWall in PokktManager, please make sure that you have already called **initPokkt** first.
2. You need to add event listener as below or also please check the OfferwallScreen class in sample app.

```
PokktManager.Dispatcher.PokktInitialisedEvent += PokktInitialisedEvent;  
PokktManager.Dispatcher.CampaignAvailabilityEvent += CampaignAvailabilityEvent  
;  
PokktManager.Dispatcher.CoinResponseEvent += CoinResponseEvent;  
PokktManager.Dispatcher.CoinResponseFailedEvent += CoinResponseFailedEvent;  
PokktManager.Dispatcher.CoinResponseWithTransIdEvent += CoinResponseWithTransIdEvent;  
PokktManager.Dispatcher.OfferwallClosedEvent += OfferwallClosedEvent;
```

3. To show OfferWall you can call **getCoins** method.

```
PokktManager.getCoins(pokktConfig);
```

4. Please register native application to handle for resume and call **getPendingCoins** method so that you can get callback to award points to the user once user returns to application after finishing with OfferWall.

```
PokktManager.getPendingCoins(pokktConfig);
```

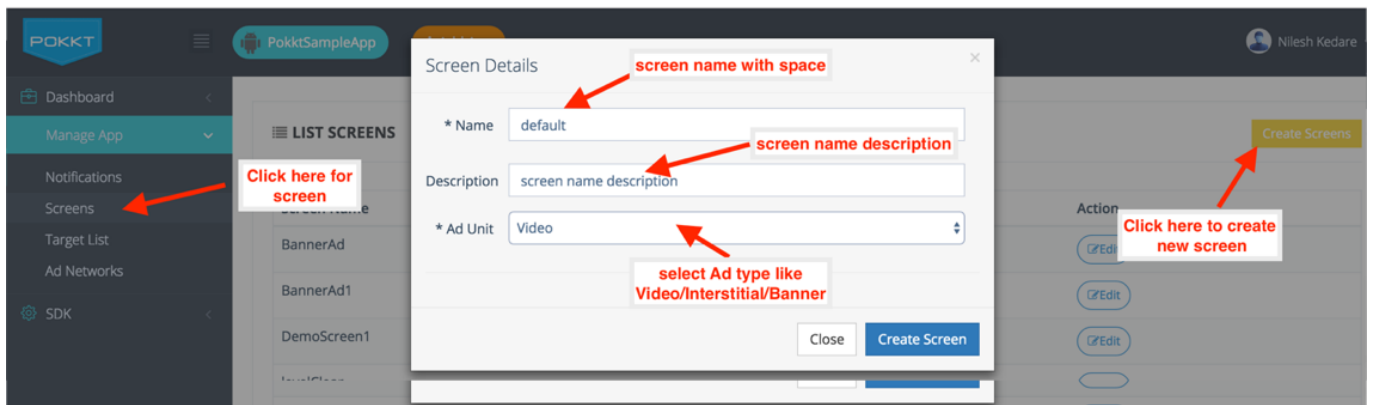
5. You can call **checkOfferWallCampaign** method to check whether the campaigns are available before showing OfferWall screen to the user. You will get a callback for this call.

```
PokktManager.checkOfferWallCampaign(pokktConfig);
```

## AdConfig (used for reward / non-reward ad)

1. In **AdConfig**, you need to provide **screenName**, **isRewarded** and **adFormat**. This screen name will be created on Pokkt dashboard.

**Screen Name:** For screen name, you will have to create on Pokkt dashboard. Please check below screen shot. This will help you to understand how to create new screen name on Pokkt dashboard.



**IsRewarded:** You need to provide ad type. Pass it "false" if you are looking for non-reward ad otherwise "true" for reward ad.

**AdFormat:** you can set VIDEO / BANNER / INTERSTITIAL like below:

```
adConfig->adFormat = VIDEO;  
adConfig->adFormat = BANNER;  
adConfig->adFormat = INTERSTITIAL;
```

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 "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. Ad 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.

## Reward Ad / Non-Reward Ad

Rewarded ad and Non-Rewarded ad can be Video or Interstitial ad.

1. You need to set true/false for rewarded or non-rewarded ad like below:

```
adConfig.isRewarded = true/false;
```

2. Followings are the values that you can set with **AdConfig**:

**ScreenName** (Required): This controls the placement of ads and can be created on Pokkt Dashboard.

**IsRewarded** (Required): Requested ad-type (rewarded/non-rewarded). Ad gratification will happen only for rewarded ads.

**AdFormat** (Required): Requested ad-format. SDK supports Video, Interstitial and Banner ad formats. Default is Video ad format.



**BackButtonDisabled:** Disable 'back' button press while on ad-screen.

**DefaultSkipTime:** If ad-skipping is allowed, this provides the time (in seconds) it will wait before the skip button appears.

**ShouldAllowSkip:** An Ad is skippable or not. If set to 'false', user will be forced to watch the ad till an ad finishes.

**ShouldAllowMute:** Toggles "mute" button on ad-screen.

**ShouldConfirmSkip:** Controls whether to show the skip-confirmation dialog box. If set to 'false', the ad will be silently closed without prompting for confirmation

**SkipConfirmMessage:** The message that will appear on skip-confirmation dialog box.

**SkipConfirmYesLabel:** 'Yes' Label of skip-confirmation dialog box.

**SkipConfirmNoLabel:** 'No' Label of skip-confirmation dialog box.

**SkipTimerMessage:** The message on countdown-timer before the skip button appears. The message must contain a '##'-placeholder to show timer value.

**IncentiveMessage:** If set, the message will be displayed while prompting user to watch the ad for certain time before it can be rewarded.

3. You will have to call cache ad to start caching ads on device.

```
PokktManager.cacheAd(adConfig)
```

4. You will need to register event for getting callback for Ad related callback like below and also please check **VideoActivity.cs** class given **sample app**.

```
PokktManager.Dispatcher.PokktInitialisedEvent += PokktInitialised;  
PokktManager.Dispatcher.AdCachingCompletedEvent += AdCachingCompleted;  
PokktManager.Dispatcher.AdCachingFailedEvent += AdCachingFailed;  
PokktManager.Dispatcher.AdAvailabilityEvent += AdAvailability;  
PokktManager.Dispatcher.AdDisplayedEvent += AdDisplayed;  
PokktManager.Dispatcher.AdCompletedEvent += AdCompleted;  
PokktManager.Dispatcher.AdClosedEvent += AdClosed;  
PokktManager.Dispatcher.AdSkippedEvent += AdSkipped;  
PokktManager.Dispatcher.AdGratifiedEvent += AdGratified;
```

5. You can call checkAdAvailability to check if the campaign are available for a particular adConfig before you try to show ad.

```
PokktManager.checkAdAvailability(adConfig)
```

6. You can call showAd method to show ad.

```
PokktManager.showAd(adConfig);
```

7. Please reward user only when AdGratified event gets triggered.

## Banner Ad

Pokkt SDK allows to show banner ad on your screen. You can set any custom size or any position for banner. There are few fixed position already given in BannerPosition class. But you can customise that also.

1. **Load Banner:** Use loadBanner to show banner ad like below:

```
PokktManager.LoadBanner(<ScreenName>, (int)BannerPosition.TopCenter, <activity>);
```

There is predefined positions are already given which can be used but there is separate method if you want to customise banner ad.

Predefined position: TOP\_LEFT, TOP\_CENTER, TOP\_RIGHT, MIDDLE\_LEFT, MIDDLE\_CENTER, MIDDLE\_RIGHT, BOTTOM\_LEFT, BOTTOM\_CENTER, BOTTOM\_RIGHT

2. **Remove Banner:** Call removeBanner method to remove banner ad from screen like below:

```
PokktManager.removeBanner(ScreenName);
```

3. **Auto Refresh Banner:** Use setBannerAutoRefresh method to disable or enable auto refresh. Default it is true and it will refresh automatically based on given time on Pokkt dashboard for particular screen name.

```
PokktManager.setBannerAutoRefresh(false/true);
```

4. **Banner Position:** Use BannerPosition class properties for banner position.

5. **Custom banner:** There is also option given to customise banner size and position by using below method:

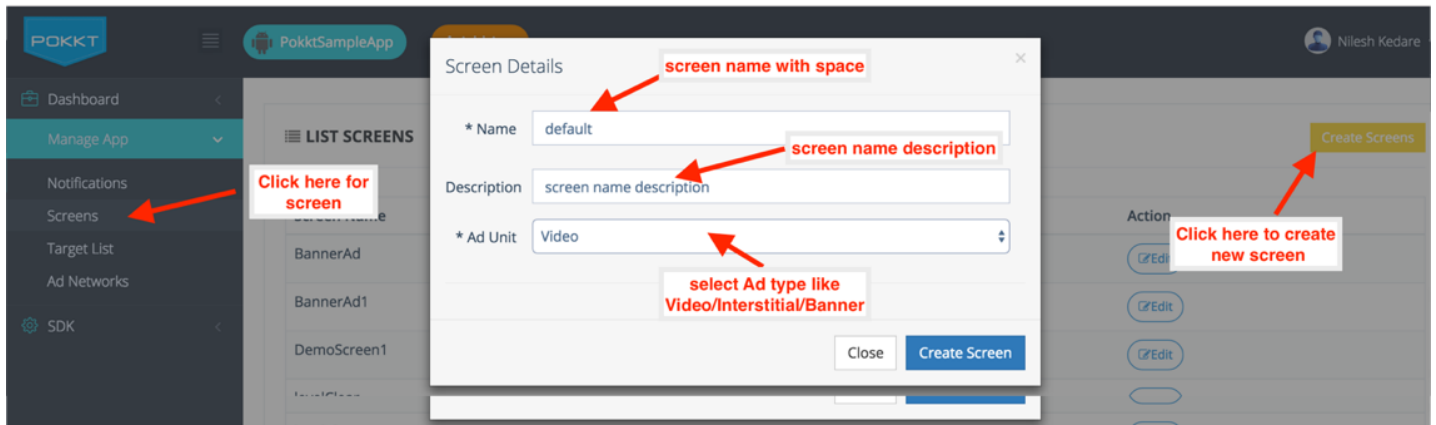
```
PokktManager.loadBannerWithRect(ScreenName, width, height, x, y);
```

6. **Banner Event:** Register event to get banner related callback like below:

```
//Register event
PokktManager.Dispatcher.BannerLoadedEvent += onBannerLoaded;
PokktManager.Dispatcher.BannerLoadFailedEvent += onBannerLoadFailed;

// Handler method
public void onBannerLoaded(String screenName) { }
public void onBannerLoadFailed(String screenName, String errorMessage) { }
```

7. **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 screen name in Pokkt dashboard.



## Mediation Info

1. Pokkt SDK supports 10 ad networks which you can integrate in your application for better monetization.
2. To integrate these networks through Pokkt, please visit the mediation menu on downloads page and download xamarin mediation zip and documentation zip files.
3. Please follow the mediation integration documents shipped for each network.
4. 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.
5. You will also need to do the mapping of Pokkt screens with the corresponding ad networks' **PlacementID / ZoneID / AdUnit** etc in the dashboard.

## Export Logs

1. Developer should call **exportLog** to export the Pokkt SDK logs to folder of your choice.

```
PokktManager.exportLog( )
```

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

## 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, setEducation, setNationality, setEmployment and setMaturityRating.*

## In-App Notifications

Developer can add In-App notifications in their dashboard.

### Add Notification

Basic

Name

App

videodemo

Platform

iOS

Android

All

Filters

Countries

App Version

Last Seen

Min

Max

Schedule

Repeat

Don't repeat

Repeat

Don't repeat

Dates

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					

Time

12

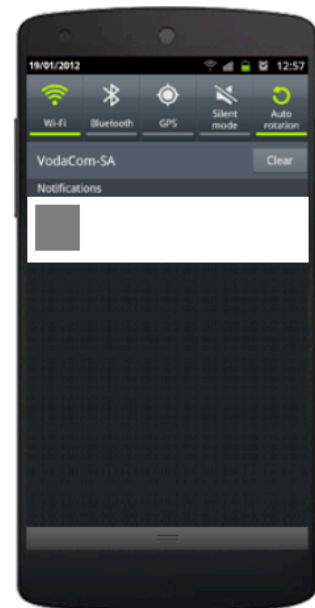
O'clock

## Message

Message

Title

Add Image



Cancel

Save

Repeat schedule can be daily, weekly monthly. Daily Repeat can have options like frequency of repeat and time in hours of notification.

Schedule

Repeat

Daily

Every

1

Day(s)

Time

12


O'clock

Message

Message

Title

Add Image



Cancel

Save

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

## Schedule

Repeat

Weekly

Every 1 Week(s)

Mon

Tue

Wed

Thu

Fri

Sat

Sun

Time

12

O'clock

## Message

Message

Title

Add Image



Cancel

Save

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

Repeat

Monthly

Every

1

Month(s)

Dates

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31					

Time

12

O'clock

## Message

Message

Title

Add Image



Cancel

Save

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.



## List Notifications

Id	Name	App Id	Header	Status	Action
1	push	1000125	Hi There	ACTIVE	<a href="#">Edit</a> <a href="#">Deactivate</a>
2	in app	1000125	Hi There	ACTIVE	<a href="#">Edit</a> <a href="#">Deactivate</a>

## 4. 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.