

AdMob Pokkt Mediation

Pokkt SDK version used: - 7.5.1

AdMob SDK version used: - 17.2.0

1. **Common Steps for AdMob Pokkt Mediation:** This is common documentation which is useful for all plugin.
2. **Native Integration:** Check the Steps Mentioned for Native.
3. **Xamarin Integration:** Check the Steps Mentioned for Xamarin.
4. **AIR Integration:** Check the Steps Mentioned for AIR.
5. **Marmalade Integration:** Check the Steps Mentioned for Marmalade.
6. **ShiVa Integration:** There is no separate documentation for this. Please follow "Common AdMob Pokkt Mediation" page.
7. **Cocos2d-x Integration:** There is no separate documentation for this. Please follow "Common AdMob Pokkt Mediation" page.
8. **Unity3D Integration:** There is no separate documentation for this. Please follow "Common AdMob Pokkt Mediation" page.
9. **Titanium Integration:** There is no separate documentation for this. Please follow "Common AdMob Pokkt Mediation" page.

Common Steps for AdMob Pokkt Mediation

Prerequisites

- Create account and login at <https://apps.admob.com/>
- Set up app and create ad units in AdMob dashboard

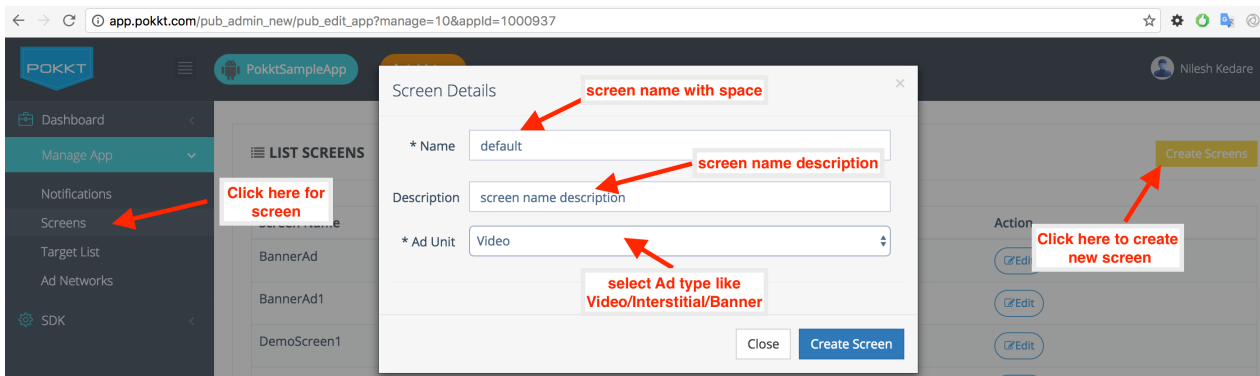
Step 1: Configure Pokkt Dashboard

- Create Account and login at <http://www.pokkt.com>
- Register your app here if not registered.
- To use AdMob Mediation, enter the following details on Pokkt Dashboard:

The screenshot shows the Pokkt Dashboard configuration page for AdMob mediation. The page is divided into three main sections: General, Screens, and Reporting. The General section includes fields for Class Name (com.pokkt.thirdparty.AdMobNetwork), Ad Unit Id, Test Device Id, Amount (1), and ECPM (1). The Screens section lists various ad formats (Banner, getcoin, levelfail, vunglevideo) with their respective Ad Unit Ids and Incents. The Reporting section includes fields for Publisher Id, Client Id, Client Secret, and Refresh Token. A sidebar on the left shows the navigation menu with 'Mediation' highlighted. A red arrow labeled 'Step 1' points to the 'Mediation' menu item. Another red arrow labeled 'Step 2' points to the 'Edit' button in the 'Status' column of the 'Screens' table.

- **Class Name (required):** This is the class name that you define inside your application to set up the network. Classname for the network can be set during SDK integration. If you are using Pokkt's default adaptors for mediation which is given in PokktMediation.jar, then class name will be com.pokkt.thirdparty.AdMobNetwork.
- **Test Device Id (optional):** For testing AdMob and if you want to see test ad for interstitial then please provide test divide Id. You can add multiple device Id with comma (,) separated without any space.

- **ECPM** : Put your expected value of ECPM which will in turn rank this network in comparison to other configured networks.
- **Screens (required)**: Here need to map “AdMob Ad unit ID” with the Pokkt screens. First create screen name in Pokkt dashboard which will reflect here as screen name. Once Pokkt screen is created, provide “Ad unit ID” for that screen. This screen name will map with this Ad unit ID. Please check below screen shot for how to create screen in Pokkt dashboard.



- **Reporting (required)**: Enter publisher id, client id, client secret and Refresh token. You can get it from your AdMob account manager. Please follow <https://developers.google.com/adwords/api/docs/guides/authentication> for details.

Note: For More Detail, Please follow the below link

<https://support.google.com/admob/answer/3052638>

- **Publisher Id (For Reporting)**: This is AdMob publisher Id. Once you create AdMob account, in right top corner you will see this publisher id.
- **Client Id, Client secret and Refresh token (For Reporting)**: Please follow the below process
 1. Open the [Google API Console Credentials page](#).
 2. From the project drop-down, select an existing project or create a new one.
 3. On the Credentials page, select Create credentials, then select OAuth client ID.
 4. Under Application type, choose Web application.
 5. Under Authorized redirect URIs, add a line with: <https://developers.google.com/oauthplayground>
 6. Click Create.
 7. Take note of the client ID and client secret. Same you need to provide here.

Refresh Token:





1. Go to the OAuth2 Playground via [this link](#), which should pre-populate some key values for you.
2. Click the gear icon in the upper right corner and check the box labeled Use your own OAuth credentials (if it isn't already checked).
3. Enter the OAuth2 client ID and OAuth2 client secret you obtained above.

4. In the section labeled Step 1 - Select & authorize APIs, enter the following URL in the text box at the bottom, if it's not already there, then click Authorize APIs:
5. <https://www.googleapis.com/auth/adsense.readonly>





OAuth 2.0 Playground
✕

▼ **Step 1** Select & authorize APIs

Select the **scope** for the APIs you would like to access or input your own OAuth scopes below. Then click the **"Authorize APIs"** button.

- ▶  Ad Exchange Buyer API v1.4
- ▶  Ad Exchange Seller API v2.0
- ▶  AdSense Host API v4.1
- ▼  AdSense Management API v1.4

https://www.googleapis.com/auth/adsense

✓ *https://www.googleapis.com/auth/adsense.readonly*
- ▶  Admin Reports API reports_v1
- ▶  BigQuery API v2
- ▶  Blogger API v3
- ▶  Books API v1

Input your own scopes

Authorize APIs

6. If prompted, log in to the account to which you want to grant access and authorization. Otherwise, confirm that the current Google user in the top right corner is the AdWords or manager account for whom you want to obtain credentials.
7. A prompt appears indicating your app would like to Manage your AdWords Campaigns. Click Accept to continue.
8. In the tab labeled Step 2 - Exchange authorization code for tokens, you should now see an Authorization code. Click Exchange authorization code for tokens

OAuth 2.0 Playground

×

▶ Step 1 Select & authorize APIs

▼ Step 2 Exchange authorization code for tokens

Once you got the Authorization Code from Step 1 click the **Exchange authorization code for tokens** button, you will get a refresh and an access token which is required to access OAuth protected resources.

Authorization code: 4/F

Exchange authorization code for tokens

Refresh token: Refresh token

Access token: Access token Refresh access token

☐ Auto-refresh the token before it expires.

Note: The OAuth Playground does not store refresh tokens, but as refresh tokens never expire, user should go to their Google Account [Authorized Access](#) page if they would like to manually revoke them.

9. If all goes well, you should see the Refresh token and Access token filled in for you (you may have to re-expand Step 2 - Exchange authorization code for tokens to see these values).

OAuth 2.0 Playground

✕

▶ **Step 1** Select & authorize APIs

▼ **Step 2** Exchange authorization code for tokens

Once you got the Authorization Code from Step 1 click the **Exchange authorization code for tokens** button, you will get a refresh and an access token which is required to access OAuth protected resources.

Authorization code:

Exchange authorization code for tokens

Refresh token: 1/v4

Access token: Refresh access token

☐ Auto-refresh the token before it expires.

The access token will expire in **3589** seconds.

Note: The OAuth Playground does not store refresh tokens, but as refresh tokens never expire, user should go to their Google Account [Authorized Access](#) page if they would like to manually revoke them.

10. Copy the Refresh token into the configuration file for your client library of choice, along with the client ID and client secret. See the instructions above to set configuration options for your client library of choice.

Notes:

1. If you got your client ID and secret only a few minutes ago, and you get an error such as **Error: redirect_uri_mismatch**, it's possible the changes you made haven't yet propagated. Click the back button in your browser, wait a few minutes, then try clicking **Authorize APIs** again.
2. You need to enable [AdSense API](#) so please go to AdSense Management API and enable this. For more details about AdMob reporting, please see [AdMob reporting documentation](#):

Step 2: Configuration in Application (Native)

- Download the mediation project from POKKT and it is suggested to use the same class file **AdMobNetwork** (AdMobNetwork.java, AdMobVideoAdManager.java, AdMobBannerManager.java and AdMobInterstitialManager.java) in your project. You have the option to make changes to these classes, in-fact you can create your own custom adaptor for AdMob using our protocol, just make sure to mention the exact name of class in the “class name” box, refer the [Step 1](#).
- Add “**Google play services ads framework**” to your project. Make sure to update your project's dependencies accordingly.
- Add the necessary permissions for AdMob as suggested in AdMob integration document to your application manifest.
- Add the recommended **activities** and **meta data** to your **AndroidManifest.xml** as suggested in AdMob integration guide.

Xamarin Android

Follow below steps to integrate Mediation in Xamarin android project:

- Add downloaded file **PokktMediation.dll** in your project under reference section.
- You need to download AdMob extension for Xamarin.
- Follow above all process to complete integration.

AIR Android

Follow below steps to integrate Mediation in AIR android project:

- Add downloaded file **PokktMediation.ane** in your project.
- You need AdMob so please google to download AdMob .ane file and use that.
- Google play service .ane file also required and few of the AdNetwork .ane comes with google play service .ane so be careful with this. We are giving PokktMediation.ane separate GooglePlayService.ANE separate.
- Follow above all process to complete integration and also check our given example specially app.xml file.

Marmalade Android

Follow below steps to integrate Mediation in Marmalade android project:

- Check downloaded file which has one mediation project “***PokktMediationExtension***” and this need to be added in your project as subproject in .mkb file.
- Marmalade is providing AdMob extension “s3eGoogleAdMob” so add this as subproject in .mkb file.
- Follow above all process to complete integration and also check our given example.