Android MPD Client Integration

Tabla de contenido

[How to import MDP into an Android Project 2](#_Toc441666847)

[How to LogIn in the MDP Platform 4](#_Toc441666848)

[Initialize the platform 4](#_Toc441666849)

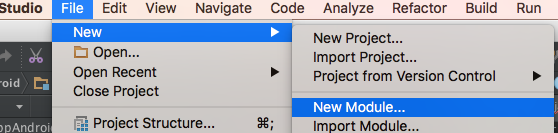
[Login application Activity 4](#_Toc441666850)

[Registry Activity 4](#_Toc441666851)

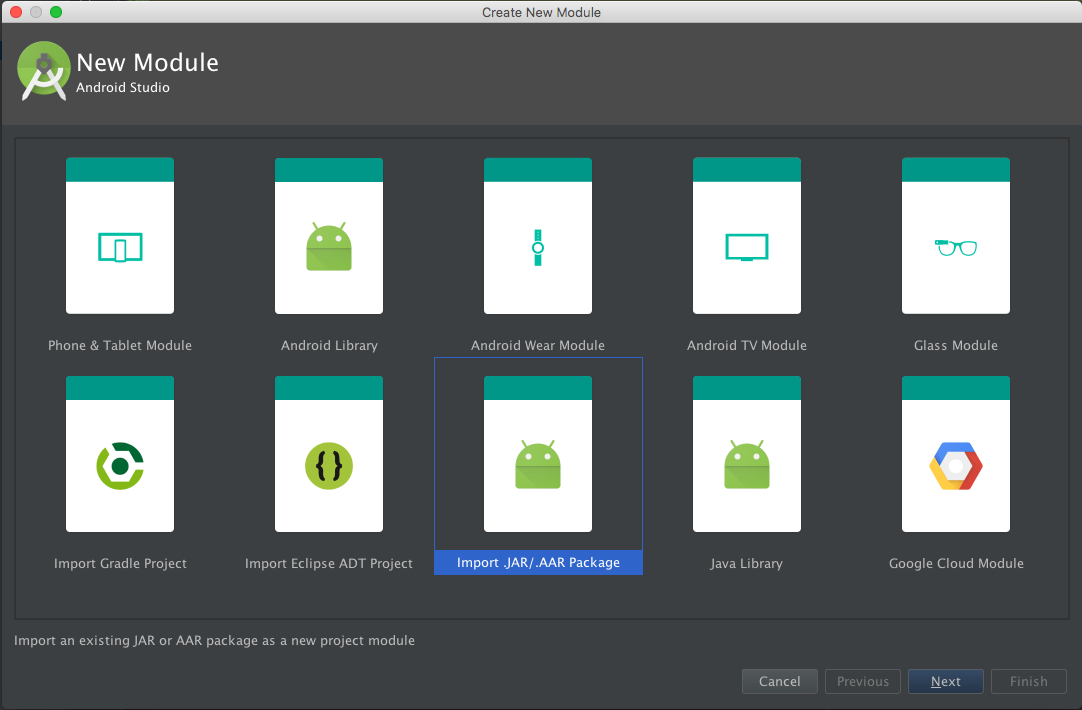
# How to import MDP into an Android Project

In order to use MDP, adal.aar and mdp.aar must be imported in the client project and added as dependencies in the main module’s **build.gradle** file.

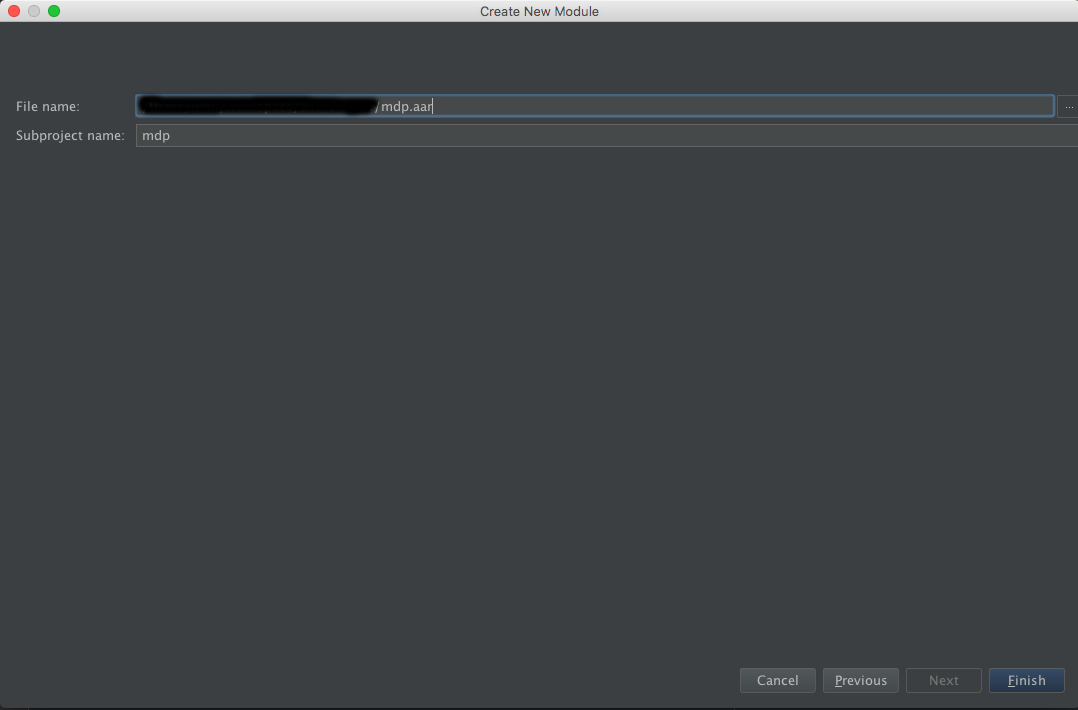
To import the modules, go to File->New->New Module.



In the flyout select **Import .JAR/.AAR Package**.



And select the .aar file desired



It will be necessary to include **Microsoft Application Insights SDK for Android** and **Google’s GSON** as dependencies too by adding

*compile 'com.microsoft.azure:applicationinsights-android:1.0-beta.4'*

*compile 'com.google.code.gson:gson:2.3.1'*

in the **build.gradle** file.

The final **build.gradle** file should look like this:

*compile project(':mdp’)*

*compile project(':adal')*

*compile 'com.microsoft.azure:applicationinsights-android:1.0-beta.4'*

*compile 'com.google.code.gson:gson:2.3.1'*

You should add inside the android block the packagingOptions as follow:

packagingOptions{  
 exclude 'META-INF/DEPENDENCIES'  
 exclude 'META-INF/NOTICE'  
 exclude 'META-INF/LICENSE'  
}

# How to LogIn in the MDP Platform

## Initialize the platform

First of all, you have to init the platform once the application has started. To do so you have to call to the method:

public static void init(Context ctx, String env, String clientId)

Of the DigitalPlatformClient class. This should be done in the onCreate method of the application.

## Login application Activity

The next step is login into the application, this is made in the launcher Activity of your application (or when it requires login into the platform), through the AuthHandler with a call to:

public void initialAccess(Activity activity, Class registryActivity, AuthListener listener)

The three parameters are:

* Activity activity: the activity from which you make the call.
* Class registryActivity: should be an Activity that you create in which you ask the user if it’s already registered in the platform or not.
* AuthListener listener: an implementation of the AuthListener. If the method onResponse is called then the login was successful and you have the token. If not the login was not successful.

This Activity in which you perform the login should override the Activity.onActivityResult method to receive the login response:

//Receive auth login page response  
if (DigitalPlatformClient.*getInstance*().getAuthHandler() != null) {  
 DigitalPlatformClient.*getInstance*().getAuthHandler().onActivityResult(this, requestCode, resultCode, data);  
}

## Registry Activity

The second parameter of the initialAccess method is this registryActivity, which as we mention before, should ask the user if it is already registered into the platform or not.

The activity has to have an UI so the user can select the appropriate option. If the user desire is to register into the platform, then we should finalize the activity as follows:

Intent i = getIntent();  
i.putExtra(AuthHandler.*SIGN\_OPTION*, AuthHandler.*SIGN\_UP\_OPTION*);  
setResult(*RESULT\_OK*, i);  
finish();

If the user is already registered and wants to sign in, then:

Intent i = getIntent();  
i.putExtra(AuthHandler.*SIGN\_OPTION*, AuthHandler.*SIGN\_IN\_OPTION*);  
setResult(*RESULT\_OK*, i);  
finish();