Rewarded Visit - Integration Guide (Android)

Rewarded Visit - Integration Guide (Android)

Skip to end of metadataGo to start of metadata
Rewarded Visit Ad uses the features provided by both Mobile Engagement (ME) SDK and Advertising SDK.

To use Rewarded visit Ad in an app both these SDKs needs to be integrated. Please follow the steps below to integrate Rewarded visit Ad in an App

Integrating ME SDK

Step 1: Add the Phunware Maven remote repository

Update Project build.gradle with the following repository

```
allprojects {
    repositories {
        maven {
            url "https://nexus.phunware.com/content/groups/public/"
        }
    }
}
```

Step 2: Add the required dependencies in your app's `build.gradle` file

1. Maas Advertising has a dependency on MaasCore. Include this in the app level gradle
2. To receive Ads include Ads SDK
3. To receive notifications on geo fence breach add Mobile Engagement SDK

```
apply plugin: 'com.android.application'

android {
    ...
}

dependencies {
    ...
    compile ('com.phunware.advertising:ads:2.4.2:release@aar'){
        transitive = true;
    }

    compile 'com.phunware.engagement:mobile-engagement:3.1.0'
    ...
}
```

Step 3 - Add Permissions

1. ME SDK requires Location and Storage permissions
2. Ads SDK requires Internet, Storage and Location permissions.

Update your `AndroidManifest.xml` to include these permissions and activity.

See AndroidManifest.xml for an example manifest file.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>

<!-- Optional permissions to enable ad geotargeting:
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
-->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

<!-- Inside of the application tag: -->
<activity
    android:name="com.phunware.advertising.internal.PwAdActivity"
    android:configChanges="keyboard|keyboardHidden|orientation|screenSize" />
```

Background location notifications currently cannot work with runtime permissions required for apps targeting Android SDK level 23 and higher, so your targetSdkVersion in your build.gradle file must be 22 or lower.

Step 4: Retrieve App ID, Access Key and Signature Key from MaaS Portal

Navigate to Phunware's MaaS portal to find your App ID, Access Key and Signature key.

GCM setup: https://developers.google.com/cloud-messaging/android/client

Part of the GCM setup is managed through the Mobile Engagement SDK including:

- Add the play-services-gcm to your application gradle
- changes required to the AndroidManifest

Step 5: Add Phunware key resources to strings.xml for App Id, Access Key, Signature Key

Add the keys obtained in step 3 to strings.xml
strings.xml

```
<string name="app_id">APPID</string>
<string name="access_key">ACCESSKEY</string>
<string name="sig_key">SIGKEY</string>
<string name="encrypt_key"></string>
```

Step 6: Add Phunware keys for App Id, Access Key, and Signature Key to Manifest

Add the keys obtained in step 4 to Manifest.

```
<meta-data android:name="com.phunware.APPLICATION_ID" android:value="@string/app_id" />
<meta-data android:name="com.phunware.ACCESS_KEY" android:value="@string/access_key" />
<meta-data android:name="com.phunware.SIGNATURE_KEY" android:value="@string/signature_key" />
<meta-data android:name="com.phunware.ENCRYPTION_KEY" android:value="@string/encrypt
```

Step 7: Configure the Mobile Engagement SDK with your environment.

You should only initialize the Mobile Engagement SDK once, after you initialize PwCoreSession.

Once initialization is complete, users will be automatically notified with messages when they visit the promoted location.

```
public class MyApplication extends Application {

  @Override
  public void onCreate() {
    super.onCreate();
    //initialize PwCoreSession
    PwCoreSession.getInstance().registerKeys(this);

    //initialize LocationMessaging
    new Engagement.Builder(this)
            .appId(/*your app ID from the MaaS portal, as a long*/)
            .build();

    //start location manager to receive location based events
    Engagement.locationManager().start();
  }

}
```

Step 8: Designate an Activity to launch from geo fence notifications

This is the Activity that will be launched when a user visits the location promoted by rewarded visit Ad.The intent which launches your activity from this notification will have the following extras:

```
url        - which points to the digital reward which is usually a digital coupon
currencyId - The currency in which in-app reward will be issued to the user. This is the second
reward for visiting the location.
amount     - The amount of in-app rewards issued to the customer (Secondary reward)
```

```
<activity
    android:name=".RewardActivity" >
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="engagement/message" />
    </intent-filter>
</activity>
```

Getting the reward details from the message

```
if (getIntent().getAction() == Intent.ACTION_VIEW) {

    Message intentMessage = getIntent().getParcelableExtra(MessageManager.EXTRA_MESSAGE);
    Engagement.analytics().trackCampaignAppLaunched(intentMessage.campaignId(),
            intentMessage.campaignType());

    boolean hasPromo = getIntent()
            .getBooleanExtra(MessageManager.EXTRA_HAS_EXTRAS, false);
    if (hasPromo) {
        final long messageId = intentMessage.campaignId();
        Engagement.messageManager().getMessage(messageId, new Callback<Message>() {
            @Override
            public void onSuccess(Message data) {

                String rewardUrl = null;
                String currencyId = null;
                String amount = null;

                List<MessageMetadata> metadataList = data.metadata();

                for (MessageMetadata tempMData : metadataList) {
                    String tempKey = tempMData.key();
                    String tempValue = tempMData.value();

                    if (tempKey.equalsIgnoreCase(METADATA_KEY_URL)) {
                        rewardUrl = tempValue;
                    }
                    if (tempKey.equalsIgnoreCase(METADATA_KEY_CURRENCY_ID)) {
                        currencyId = tempValue;
                    }
                    if (tempKey.equalsIgnoreCase(METADATA_KEY_AMOUNT)) {
                        amount = tempValue;
                    }
                }
                //show the in-app reward
                if (currencyId != null && amount != null) {
                    showRewardDialog(currencyId, amount);
                }

                //display the second reward (digital coupon)
                showReward(rewardUrl);

            }

            @Override
            public void onFailed(Throwable e) {
                Log.e(TAG, "Failed to get message for id: " + messageId, e);
            }
        });
    }
}
```

For more details on integrating ME SDK ME SDK Integration

Integrating Ads SDK

Step 1 - Add Dependencies

This has already been done in Step 2 of ME SDK Integration

Step 2 - Add Permissions for Ads SDK

Done in Step 3 of ME SDK Integration

Step 3 - Request Rewarded Visit Ad in your app

Create Rewarded Video Ad with Rewarded Visit support in Ad portal. Use that zone ID to request a Rewarded Video Ad from your app.

```
import com.phunware.advertising.*;

//...

PwRewardedVideoAd rewardedVideoAd = PwRewardedVideoAd.getInstance(this, "YOUR_REWARDED_VISIT_ZONE_ID")
;
rewardedVideoAd.setUserId("YOUR_LOCAL_PLAYER_ID"); //This is required.

//You can send custom data on a HashMap
HashMap<String, String> customData = new HashMap<>();
        customData.put("Data 1", "value 1");
        customData.put("Data 2", "value 2");
        //Note: this custom data is converted to JSON, and has a limit of 255 characters, if this
exceed the 255 limit the SDK will delete the necessary keys of data to reach the limit.
        mRewardedVideoAd.setCustomData(customData);

//Setting listeners.
rewardedVideoAd.setListener(new PwRewardedVideoAd.PwRewardedVideoAdListener() {
            @Override
            public void rewardedVideoDidLoad(PwRewardedVideoAd rewardedVideoAd, TVASTRewardedVideoInfo
rewardedVideoInfo) {
                rewardedVideoAd.show();
            }

            @Override
            public void rewardedVideoDidClose(PwRewardedVideoAd rewardedVideoAd, TVASTRewardedVideoInfo
rewardedVideoInfo) {
                Log.d("TAG", "rewardedVideoDidClose");
            }

            @Override
            public void rewardedVideoDidFail(PwRewardedVideoAd rewardedVideoAd, String error,
TVASTRewardedVideoInfo rewardedVideoInfo) {
                //If rewarded video doesn't have remaining views, you can check the error code if this
exist.
                if(rewardedVideoInfo.getError() == 557){
                    Toast.makeText("getContext()", "You don't have remaining views",
Toast.SHORT).show();
                }
            }

            @Override
            public void rewardedVideoActionWillLeaveApplication(PwRewardedVideoAd rewardedVideoAd,
TVASTRewardedVideoInfo rewardedVideoInfo) {

            }

            @Override
            public void rewardedVideoDidEndPlaybackSuccessfully(PwRewardedVideoAd rewardedVideoAd,
RVSuccessInfo rewardedVideoSuccessInfo, TVASTRewardedVideoInfo rewardedVideoInfo) {

                Log.d("REWARD:", rewardedVideoSuccessInfo.getCurrencyId());
                Log.d("AMOUNT:", String.valueOf(rewardedVideoSuccessInfo.getAmount()));

                //Remaining views after video completes.
                Log.d("REMAINING VIEWS:",
String.valueOf(rewardedVideoSuccessInfo.getRemainingViews()));
            }
        });

rewardedVideoAd.load();
```

At the end of the video, user will be shown an end card with Rewarded Visit details, like the location where promotion is going on.

Step 4 - Receiving the reward

When the user visits the location that was shown in the Rewarded visit Ad, he will get a notification with the reward that was earlier set up in Step 8 of ME SDK integration