

# Advertising SDK Integration Guide (Android)

## Advertising SDK Integration Guide (Android)

### Version 2.4.3

This is Phunware's Android SDK for the MaaS Advertising module. Visit the [Phunware Ads portal](#) for more details and to sign up.

## Requirements

- MaaS Core v3.0.3 or greater
- Use Google Play services to enable Advertising ID support (recommended); installation instructions [here](#)

## Getting Started

- Download [MaaS Advertising](#) and run the included sample app.
- Continue reading below for installation and integration instructions.

## Installation

Usage of Maas Advertising requires the following changes to the build.gradle and AndroidManifest.xml

## Libraries

To use Advertising SDK add the following dependency to your build.gradle

```
compile
('com.phunware.advertising:ads:2.4.3
:release@aar'){
    transitive = true;
}
```

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

```
<uses-permission
android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK
_STATE" />
<uses-permission
android:name="android.permission.READ_PHONE_STA
TE" />

<!-- Optional permissions to enable ad
geotargeting:
<uses-permission
android:name="android.permission.ACCESS_COARSE_
LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LO
CATION" />
-->

<!-- Inside of the application tag: -->
<activity

android:name="com.phunware.advertising.internal
.PwAdActivity"

android:configChanges="keyboard|keyboardHidden|
orientation|screenSize" />
```

## Integration

The primary methods in MaaS Advertising involve displaying the various ad types.

### Native Ad Usage

Native ads are advertisements designed to fit naturally into your app's look and feel. Predefined ad features are provided as a JSON payload which your app consumes in a template that follows your UI's theme.

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

// ...

String zoneId = "YOURNATIVEADZONEID";
PwNativeAd nativeAd =
PwAdvertisingModule.get().getNativeAdForZone(co
ntext, zoneId);
nativeAd.setListener(new
PwNativeAd.PwNativeAdListener() {
    @Override
    public void nativeAdDidLoad(PwNativeAd
nativeAd) {
        try {
            renderUiFromNativeAd(nativeAd);

        } catch (JSONException e) {
            // Log the error and discard this
native ad instance.
        }
    }

    @Override
    public void nativeAdDidFail(PwNativeAd
nativeAd, String errMsg) {
        // The ad failed to load and the errMsg
describes why.
        // Error messages are not intended for
user display.
    }
});

nativeAd.load();

// ...

// ... when native ad data is displayed on
screen:
nativeAd.trackImpression();

// ...

// ... when native ad is clicked:
nativeAd.click(context);
```

```
private void renderUiFromNativeAd(PwNativeAd
nativeAd) throws JSONException {
    JSONObject json = new
JSONObject(nativeAd.getAdData());
    String adtitle = json.optString("adtitle");
    String imageurl =
json.optString("iconurl");
    double stars = json.optDouble("rating");
    String html = json.optString("html");
    String adtext = json.optString("adtext");
    String cta = json.optString("cta");

    // Use the data to build a view item of
your own design.
}
```

## Multiple Ads Request

```

String zoneId = "YOURNATIVEADZONEID";
PwAdRequest request =
PwAdvertisingModule.get().getAdRequestForZone(zoneId);

PwAdvertisingModule.get().getNativeAdLoader();
int numberOfAdsToLoad = 10;
PwAdLoader<PwNativeAd> adLoader =
PwAdvertisingModule.get().getNativeAdLoader();

adLoader.multiLoad(context, request,
numberOfAdsToLoad,
new
PwAdLoader.PwAdLoaderListener<PwNativeAd>() {
    @Override
    public void onSuccess(PwAdLoader adLoader,
List<PwNativeAd> nativeAdsList) {
        for(PwNativeAd nativeAd : nativeAdsList) {
            // Use the native ad to build a view item.
            try {
                renderUiFromNativeAd(nativeAd);
            } catch (JSONException e) {
                // Log the error and discard this native
ad instance.
            }
        }
    }

    @Override
    public void onFail(PwAdLoader adLoader,
String errMsg) {
        // No ads are returned and the
errMsg describes why.
        // Error messages are not intended
for user display.
    }
}
);

```

## Banner Usage

Banners are inline ads that are shown alongside your app's interface.

For XML usage only.

Add this to your layout xml.

```
<!-- Add a banner to your layout xml. -->
<!-- This will cause a 320x50 ad to be created,
which will automatically kick off ad rotation.
-->
<com.phunware.advertising.PwBannerAdView
    android:id="@+id/bannerAd"
    android:layout_width="320dp"
    android:layout_height="50dp"
    zone="YOUR_ZONE_ID" />
```

OR

If Zone is not specified, add this to your layout .xml

```
<!-- Add a banner to your layout xml. -->
<!-- This will cause a 320x50 ad to be created,
which will automatically kick off ad rotation.
-->
<com.phunware.advertising.PwBannerAdView
    android:id="@+id/bannerAd"
    android:layout_width="320dp"
    android:layout_height="50dp" />
```

Add this to your activity:

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

// ...

PwBannerAdView bannerAdView =
(PwBannerAdView)findViewById(R.id.bannerAd);
bannerAdView.startRequestingAdsForZone("YOURBAN
NERZONE_ID");
```

## Interstitial Usage

Interstitial ads are best used at discrete stopping points in your app's flow, such as at the end of a game level or when the player dies.

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

// ...

PwInterstitialAd interstitialAd =
PwAdvertisingModule.get().getInterstitialAdForZone(
this, "YOURINTERSTITIALZONE_ID");
interstitialAd.show();
```

## Video Ads Usage

Video ads are interstitial ads that play

a video. They are best used at discrete stopping points in your app's flow, such as at the end of a game level or when the player dies.

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

// ...

PwVideoInterstitialAd videoAd =
PwAdvertisingModule.get().getVideoInterstitialA
dForZone(this, "YOURVIDEOZONE_ID");
videoAd.show();
```

## Rewarded Video Ads Usage

Rewarded Video ads are interstitial ads that play a video and reward the user after see the video. They are best used on games making the user wants to see an Ad to be rewarded.

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

//...

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

//You can send custom data in 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 exceeds 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) {

    }

    @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(PwRear
dedVideoAd rewardedVideoAd,
TVASTRewardedVideoInfo rewardedVideoInfo) {

}

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

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

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

@Override
public void
onCacheCompleted(PwRewardedVideoAd
rewardedVideoAd, TVASTRewardedVideoInfo

```



```
rewardedVideoInfo) {
    if (rewardedVideoAd != null) {
        rewardedVideoAd.show();
    }
}

@Override
public void
onCacheProgress(PwRewardedVideoAd
rewardedVideoAd, int percentageCompleted) {

}
```

```
});  
rewardedVideoAd.load();
```