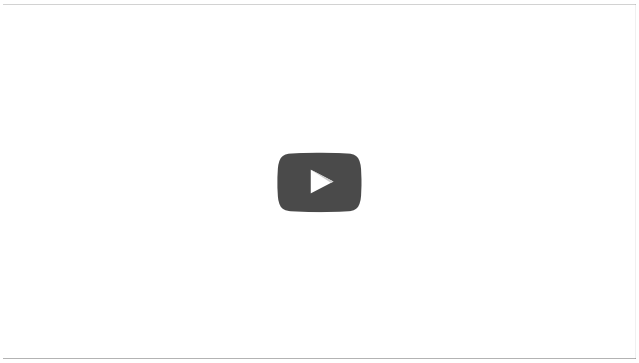
Interstitial Ads



Interstitial ads are full-screen ads that cover the interface of their host app. They're typically displayed at natural transition points in the flow of an app, such as between activities or during the pause between levels in a game. When an app shows an interstitial ad, the user has the choice to either tap on the ad and continue to its destination or close it and return to the app.

This guide explains how to integrate interstitial ads into an Android app.

Prerequisite

- Import the Google Mobile Ads SDK, either by itself (https://developers.google.com/admob/android/quick-start) or as part of Firebase (//firebase.google.com/docs/admob/android/quick-start).

Create an interstitial ad object

Interstitial ads are requested and shown by **`InterstitialAd`** (https://developers.google.com/android/reference/com/google/android/gms/ads/InterstitialAd) objects. The first step is instantiating **`InterstitialAd`** and setting its ad unit ID. This is done in the **`onCreate()`** method of an Activity:

```
public class MainActivity extends Activity {

    private InterstitialAd mInterstitialAd;

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mInterstitialAd = new InterstitialAd(this);
        mInterstitialAd.setAdUnitId("ca-app-pub-3940256099942544/1033173712");
    }
}
```

A single `InterstitialAd` object can be used to request and display multiple interstitial ads over the course of an activity's lifespan, so you only need to construct it once.

Always test with test ads

The sample code above contains an ad unit ID and you're free to request ads with it. It's been specially configured to return test ads rather than production ads for every request, which makes it safe to use.

However, once you register an app in the AdMob UI and create your own ad unit IDs for use in your app, you'll need to explicitly configure your device as a test device (https://developers.google.com/admob/android/test-ads#enable_test_devices) when you're developing. This is **extremely important**.

Testing with real ads (even if you never tap on them) is against AdMob policy and can cause your account to be suspended. See Test Ads (https://developers.google.com/admob/android/test-ads) for information on how you can make sure you always get test ads when developing.

Load an ad

To load an interstitial ad, call the `InterstitialAd` object's `loadAd()` (https://developers.google.com/android/reference/com/google/android/gms/ads/AdView#loadAd(com.google.android.gms.ads.AdRequest)) method. This method accepts an `AdRequest` (https://developers.google.com/android/reference/com/google/android/gms/ads/AdRequest) object as its single parameter:

```
public class MainActivity extends Activity {
```

```
    private InterstitialAd mInterstitialAd;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mInterstitialAd = new InterstitialAd(this);
        mInterstitialAd.setAdUnitId("ca-app-pub-3940256099942544/1033173712");
        mInterstitialAd.loadAd(new AdRequest.Builder().build());
    }
}
```

Show the ad

Interstitial ads should be displayed during natural pauses in the flow of an app. Between levels of a game is a good example, or after the user completes a task. To show an interstitial, use the **`isLoaded()`** (https://developers.google.com/android/reference/com/google/android/gms/ads/InterstitialAd#isLoaded()) method to verify that it's done loading, then call **`show()`** (https://developers.google.com/android/reference/com/google/android/gms/ads/InterstitialAd#show()). The interstitial ad from the previous code example could be shown in a button's **`OnClickListener`** like this:

```
...
mMyButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (mInterstitialAd.isLoaded()) {
            mInterstitialAd.show();
        } else {
            Log.d("TAG", "The interstitial wasn't loaded yet.");
        }
    }
});
...
```

Ad events

To further customize the behavior of your ad, you can hook onto a number of events in the ad's lifecycle: loading, opening, closing, and so on. You can listen for these events through the `AdListener` (https://developers.google.com/android/reference/com/google/android/gms/ads/AdListener) class.

To use an `AdListener` (https://developers.google.com/android/reference/com/google/android/gms/ads/AdListener) with an `InterstitialAd` (https://developers.google.com/android/reference/com/google/android/gms/ads/InterstitialAd) object, simply call the `setAdListener()` (https://developers.google.com/android/reference/com/google/android/gms/ads/InterstitialAd#setAdListener(com.google.android.gms.ads.AdListener)) method:

```
mInterstitialAd.setAdListener(new AdListener() {
    @Override
    public void onAdLoaded() {
        // Code to be executed when an ad finishes loading.
        Log.i("Ads", "onAdLoaded");
    }

    @Override
    public void onAdFailedToLoad(int errorCode) {
        // Code to be executed when an ad request fails.
        Log.i("Ads", "onAdFailedToLoad");
    }

    @Override
    public void onAdOpened() {
        // Code to be executed when the ad is displayed.
        Log.i("Ads", "onAdOpened");
    }

    @Override
    public void onAdLeftApplication() {
        // Code to be executed when the user has left the app.
        Log.i("Ads", "onAdLeftApplication");
    }

    @Override
    public void onAdClosed() {
        // Code to be executed when when the interstitial ad is closed.
        Log.i("Ads", "onAdClosed");
```

```
    }
});
```

Each of the overridable methods in **AdListener** (https://developers.google.com/android/reference/com/google/android/gms/ads/AdListener) corresponds to an event in the lifecycle of an ad.

Overridable methods

onAdLoaded()	The onAdLoaded() (https://developers.google.com/android/reference/com/g executed when an ad has finished loading.
onAdFailedToLoad()	The onAdFailedToLoad() (https://developers.google.com/android/reference/com/google/android/gms/a **errorCode** parameter that indicates what type of failure occurred. The possible (https://developers.google.com/android/reference/com/google/android/gms/a • **ERROR_CODE_INTERNAL_ERROR** - Something happened internally; for instar • **ERROR_CODE_INVALID_REQUEST** - The ad request was invalid; for instance • **ERROR_CODE_NETWORK_ERROR** - The ad request was unsuccessful due to n • **ERROR_CODE_NO_FILL** - The ad request was successful, but no ad was retu
onAdOpened()	This method is invoked when the ad is displayed, covering the device's screen.
onAdLeftApplication()	This method is invoked when a user click opens another app (such as the Googl
onAdClosed()	This method is invoked when when the interstitial ad is closed due to the user ta paused its audio output or game loop, this is a great place to resume it.

Using an AdListener to reload

The **AdListener** (https://developers.google.com/android/reference/com/google/android/gms/ads/AdListener) class's **onAdClosed()** (https://developers.google.com/android/reference/com/google/android/gms/ads/AdListener#onAdClosed()) method is a handy place to load a new interstitial after displaying the previous one:

```
...
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mInterstitialAd = new InterstitialAd(this);
    mInterstitialAd.setAdUnitId("ca-app-pub-3940256099942544/1033173712");
    mInterstitialAd.loadAd(new AdRequest.Builder().build());

    mInterstitialAd.setAdListener(new AdListener() {
        @Override
        public void onAdClosed() {
            // Load the next interstitial.
            mInterstitialAd.loadAd(new AdRequest.Builder().build());
        }

    });
}
...
```

Some best practices

Consider whether interstitial ads are the right type of ad for your app.

Interstitial ads work best in apps with natural transition points. The conclusion of a task within an app, like sharing an image or completing a game level, creates such a point. Because the user is expecting a break in the action, it's easy to present an interstitial ad without disrupting their experience. Make sure you consider at which points in your app's workflow you'll display interstitial ads and how the user is likely to respond.

Remember to pause the action when displaying an interstitial ad.

There are a number of different types of interstitial ads: text, image, video, and more. It's important to make sure that when your app displays an interstitial ad, it also suspends its use of some resources to allow the ad to take advantage of them. For example, when you make the call to display an interstitial ad, be sure to pause any audio output being produced by your app. You can resume playing sounds in the `onAdClosed()` (https://developers.google.com/android/reference/com/google/android/gms/ads/AdListener#onAdClosed())
event handler, which will be invoked when the user has finished interacting with the ad. In addition, consider temporarily halting any intense computation tasks (such as a game

loop) while the ad is being displayed. This will make sure the user doesn't experience slow or unresponsive graphics or stuttered video.

Allow for adequate loading time.

Just as it's important to make sure you display interstitial ads at an appropriate time, it's also important to make sure the user doesn't have to wait for them to load. Loading the ad in advance by calling `loadAd()` (https://developers.google.com/android/reference/com/google/android/gms/ads/AdView#loadAd(com.google.android.gms.ads.AdRequest)) before you intend to call `show()` can ensure that your app has a fully loaded interstitial ad at the ready when the time comes to display one.

Don't flood the user with ads.

While increasing the frequency of interstitial ads in your app might seem like a great way to increase revenue, it can also degrade the user experience and lower clickthrough rates. Make sure that users aren't so frequently interrupted that they're no longer able to enjoy the use of your app.

Additional resources

Samples

- Interstitial sample app (//github.com/googleads/googleads-mobile-android-examples/tree/master/admob/InterstitialExample) on GitHub

Mobile Ads Garage video tutorials

- Interstitial Implementation (//www.youtube.com/watch?v=9gER3z_xWkQ&list=PLOU2XLYxmsIKX0pUJV3uqp6N3NeHwHh0c&index=4)
- Interstitial Best Practices (//www.youtube.com/watch?v=r2RgFD3Apyo&index=5&list=PLOU2XLYxmsIKX0pUJV3uqp6N3NeHwHh0c)

Next steps

- If you haven't already, create your own interstitial ad unit in the AdMob UI (//apps.admob.com).
- Learn about ad targeting (https://developers.google.com/admob/android/targeting) and interstitial ad guidance (//support.google.com/admob/answer/6066980).
- Try another ad format:
  - Banner (https://developers.google.com/admob/android/banner)
  - Rewarded Video (https://developers.google.com/admob/android/rewarded-video)
  - Native (https://developers.google.com/admob/android/native)

Last updated July 14, 2017.