

Android SDK

RapidoReach Android Integration Guide

Get Your API Key

Sign-up for a new developer account and create a new Android app [here](#) and copy your API Key.

Download the SDK

Download the latest version of the Android SDK [here](#). Add the RapidoReach-1.0.0.aar file to your projects "libs" folder.

Update your module's build.gradle file

Include the rapidoreach.aar, Google Play Services and androidx.appcompat in your build.gradle file. A Google Advertising ID helps us serve offers and surveys so we recommend adding the `Google Play Services SDK` to your project. Be sure your `minSdkVersion` is set to at least 16.

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

android {
    ...
    defaultConfig {
        ...
        minSdkVersion 16
        ...
    }
}
```

```
    ...
}

dependencies {
    ...
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.gms:play-services-ads:19.2.0'
    implementation (name: 'RapidoReach-1.0.0', ext: 'aar')
}
```

Update your projects's build.gradle file

Ensure that your project has access to the 'libs' file by including the following in your project level build.gradle file.

```
buildscript...

allprojects {
    repositories {
        jcenter()
        flatDir {
            dirs 'libs'
        }
    }
}
```

Proguard

If you use proguard in your app. Be sure to add this line to your rules file:

```
-keep class rapidoreach.com.** { *; }
```

Import SDK in your android activity (MainActivity.java)

```
import com.rapido reach.rapido reach sdk.RapidoReach;
import com.rapido reach.rapido reach sdk.RapidoReachRewardListener;
import com.rapido reach.rapido reach sdk.RapidoReachSurveyAvailableListener;
import com.rapido reach.rapido reach sdk.RapidoReachSurveyListener;
```

Implement interfaces and methods (MainActivity.java)

```
public class MainActivity extends AppCompatActivity implements RapidoReachRewardListener,
RapidoReachSurveyListener, RapidoReachSurveyAvailableListener {
```

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

```
    @Override
    public void onReward(int i) {

    }
```

```
    @Override
    public void rapidoReachSurveyAvailable(boolean b) {

    }
```

```
    @Override
    public void onRewardCenterClosed() {

    }
```

```
    @Override
    public void onRewardCenterOpened() {

    }
```

```
}
```

Initialize RapidoReach

In your activity overwrite the `onCreate()` method and initialize the RapidoReach SDK with the `initWithApiKeyAndUserIdAndActivityContext` call. And implement the RapidoReach `onPause()` and `onResume()` calls. Replace the `YOUR_API_TOKEN` with the actual api key found on your app. Replace `YOUR_USER_ID` with your unique ID for your appuser. If you do not have a unique user ID we recommend just using their Google Advertising ID (`GPS_ID`)

```
//initialize RapidoReach
RapidoReach.initWithApiKeyAndUserIdAndActivityContext(`YOUR_API_TOKEN`, `YOUR_USER_ID`, this);

//customize navigation header
RapidoReach.getInstance().setNavigationBarText("Demo App");
RapidoReach.getInstance().setNavigationBarColor("#211548");
RapidoReach.getInstance().setNavigationBarTextColor("#FFFFFF");

//set reward and survey status listeners
RapidoReach.getInstance().setRapidoReachRewardListener(this);
RapidoReach.getInstance().setRapidoReachSurveyListener(this);
RapidoReach.getInstance().setRapidoReachSurveyAvailableListener(this);
```

Reward Center

Next, in your activity, implement the logic to display the reward center. Call the `showRewardCenter` method when you are ready to send the user into the reward center where they can complete surveys in exchange for your virtual currency. We automatically convert the amount of currency a user gets based on the conversion rate specified in your app.

```
Button btn = (Button) findViewById(R.id.button);
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Log.d(TAG, "Button is clicked");
        if (RapidoReach.getInstance().isSurveyAvailable()) {
            RapidoReach.getInstance().showRewardCenter();
        }
    }
});
```

```
}  
});
```

Reward Callback

To ensure safety and privacy, we notify you of all awards via a server side callback. In the developer dashboard for your App add the server callback that we should call to notify you when a user has completed an offer. Note the user ID pass into the initialize call will be returned to you in the server side callback. More information about setting up the callback can be found in the developer dashboard.

The quantity value will automatically be converted to your virtual currency based on the exchange rate you specified in your app. Currency is always rounded in favor of the app user to improve happiness and engagement.

Client Side Award Callback

For security purposes we always recommend that developers utilize a server side callback, however we also provide APIs for implementing a client side award notification if you lack the server structure or a server altogether or want more real-time award notification. It's important to only award the user once if you use both server and client callbacks (though your users may not be opposed!).

```
public class MyActivity extends Activity implements RapidoReachRewardListener {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        ...  
        // initialize RapidoReach  
        super.onCreate();  
        RapidoReach.initWithApiKeyAndUserIdAndActivityContext("YOUR_API_TOKEN",  
"YOUR_USER_ID", "YOUR_ACTIVITY");  
  
        // set RapidoReach client-side reward listener  
        RapidoReach.getInstance().setRapidoReachRewardListener(this);  
  
    }  
  
    // implement callback for award notification  
    @Override
```

```
public void onReward(int i) {
    Log.d(TAG, "onReward: " + i);
}
}
```

Reward Center Events

You can optionally listen for the `onRewardCenterOpened` and `onRewardCenterClosed` events by implementing the `RapidoReachSurveyListener` interface.

```
public class MyActivity extends Activity implements RapidoReachSurveyListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        // initialize RapidoReach
        super.onCreate();
        RapidoReach.initWithApiKeyAndUserIdAndActivityContext("YOUR_API_TOKEN",
"YOUR_USER_ID", "YOUR_ACTIVITY");

        // set RapidoReach survey event listener
        RapidoReach.getInstance().RapidoReachSurveyListener(this);
    }

    // reward center opened. time to start earning content!
    @Override
    public void onRewardCenterOpened() {
        Log.d(TAG, "onRewardCenterOpened");
    }

    // reward center closed. restart music/app.
    @Override
    public void onRewardCenterClosed() {
        Log.d(TAG, "onRewardCenterClosed");
    }
}
```

Survey Available Callback

If you'd like to be notified when a survey is available you can add a listener:

```
public class MyActivity extends Activity implements RapidoReachSurveyAvailableListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        // initialize RapidoReach
        super.onCreate();
        RapidoReach.initWithApiKeyAndUserIdAndActivityContext("YOUR_API_TOKEN",
"YOUR_USER_ID", "YOUR_ACTIVITY");

        // set RapidoReach survey available listener
        RapidoReach.getInstance().setRapidoReachSurveyAvailableListener(this);

    }

    // implement callback for survey available
    @Override
    public void rapidoReachSurveyAvailable(int surveyAvailable) {
        Log.d(TAG, "rapidoReachSurveyAvailable: " + surveyAvailable);
    }
}
```

Testing SDK

When you initially create your app we automatically set your app to Test mode. While in test mode a survey will always be available. Note - be sure to set your app to Live in your dashboard before your app goes live or you won't serve any real surveys to your users!

Customizing SDK

We provide several methods to customize the navigation bar to feel like your app.

```
RapidoReach.getInstance().setNavigationBarText("Demo App");
RapidoReach.getInstance().setNavigationBarColor("#17b4b3");
```

```
RapidoReach.getInstance().setNavigationBarTextColor("#FFFFFF");
```

Revision #1

Created 2021-03-16 08:08:13 UTC by Sudarshan Kondgekar

Updated 2021-03-16 08:11:31 UTC by Sudarshan Kondgekar