Unity SDK

RapidoReach Unity Integration Guide

Get Your API Key

Sign-up for a new developer account and create a new Unity app here and copy your API Key.

Download the Plugin

Download the latest version of the RapidoReach Unity Plugin here.

Import the Unity Package

From Unity go to Assets menu → Import package → Custom package → choose the unzipped unity package.

Android

Ensure the RapidoReach-1.0.0.aar and other files were successfully imported with the RapidoReach.cs file in your "Assets/Plugins" folder.

In your player settings ensure the minimum API is set to 15 (Jelly Bean) or higher.

Initialize RapidoReach

We recommend initializing RapidoReach as soon as possible so we can begin preparing surveys for the user. In the Initialize method you'll set your API key and the user's ID that will be passed back into your server side callback when the user has earned currency for completing a survey.

```
#if UNITY_ANDROID || UNITY_IOS
```

```csharp
  // Your GameObject that triggers the Reward Center

  void Start()
  {
    ConfigureRapidoReach();
  }
    public void ConfigureRapidoReach()
  {
#if UNITY_IOS || UNITY_ANDROID
    RapidoReach.Initialize("d5ece53df8ac97409298325fec81f3f7", "ANDROID_TEST_ID");
    RapidoReach.SetListener(gameObject.name);

    // optional
    RapidoReach.SetNavigationBarText("RapidoReach Unity N");
    RapidoReach.SetNavigationBarColor("#211548");
    RapidoReach.SetNavigationBarTextColor("#FFFFFF");
#endif
  }


  // call this function to show reward center on button click etc
  public void showRewardCenter(){
    Debug.Log("UNITY: Calling show reward center");
#if UNITY_IOS || UNITY_ANDROID
    RapidoReach.ShowRewardCenter();
#endif
  }

  void OnReward(string quantity)
  {
    FindObjectOfType<ReceivedRewards>().GetComponent<Text>().text = quantity;
    Debug.Log("RapidoReach OnReward: " + quantity);
  }


  void OnRewardCenterOpened()
  {
    FindObjectOfType<ReceivedRewards>().GetComponent<Text>().text = "Loading ...";
    Debug.Log("RapidoReach OnRewardCenterOpened!");
  }


  void OnRewardCenterClosed()
```

```
{

  Debug.Log("RapidoReach OnRewardCenterClosed!");

}


void RapidoReachSurveyAvailable(string available)

{

  Debug.Log("RapidoReach RapidoReachSurveyAvailable: " + available);

}


#endif
```

Google Play Services

A Google Advertising ID helps us serve offers and surveys so we recommend adding Google Play Services to your project. We provide the files to pull this in for you.

iOS

Ensure the UnityPluginBridge.mm and RapidoReachSDK-Bridging-Header.h files and frameworks folder are in your "Assets/Plugins/iOS" folder and the RapidoReach.cs file is in your "Assets/Plugins" folder.

Initialize RapidoReach

We recommend initializing RapidoReach as soon as possible so we can begin preparing surveys for the user. In the Initialize method you'll set your API key and the user's ID that will be passed back into your server side callback when the user has earned currency for completing a survey.

```
#if UNITY_ANDROID || UNITY_IOS


// Your GameObject that triggers the Reward Center


void Start()

{

  ConfigureRapidoReach();

}

  public void ConfigureRapidoReach()

  {

#if UNITY_IOS || UNITY_ANDROID

    RapidoReach.Initialize("d5ece53df8ac97409298325fec81f3f7", "ANDROID_TEST_ID");
```

```csharp
      RapidoReach.SetListener(gameObject.name);


      // optional
      RapidoReach.SetNavigationBarText("RapidoReach Unity N");
      RapidoReach.SetNavigationBarColor("#211548");
      RapidoReach.SetNavigationBarTextColor("#FFFFFF");
#endif
  }


  // call this function to show reward center on button click etc
  public void showRewardCenter(){
      Debug.Log("UNITY: Calling show reward center");
#if UNITY_IOS || UNITY_ANDROID
      RapidoReach.ShowRewardCenter();
#endif
  }


  void OnReward(string quantity)
  {
      FindObjectOfType<ReceivedRewards>().GetComponent<Text>().text = quantity;
      Debug.Log("RapidoReach OnReward: " + quantity);
  }


  void OnRewardCenterOpened()
  {
      FindObjectOfType<ReceivedRewards>().GetComponent<Text>().text = "Loading ...";
      Debug.Log("RapidoReach OnRewardCenterOpened!");
  }


  void OnRewardCenterClosed()
  {
      Debug.Log("RapidoReach OnRewardCenterClosed!");
  }


  void RapidoReachSurveyAvailable(string available)
  {
      Debug.Log("RapidoReach RapidoReachSurveyAvailable: " + available);
  }
```

```
#endif
```

Libraries

We include a post processing script to automatically include all libraries into your project and perform any additional configuration in the build settings. See the iOS guide if you would like to verify your Xcode project setup.

Reward Callback

To ensure safety and privacy, we notify you of all awards via a server side callback. This callback will be triggered for both Android and iOS apps.

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!).

In order to receive notifications you must implement the OnReward method on the GameObject you'd like to receive notifications that a user earned content. Then you must register that gameObject with the SetListener method.

Additionally you can also implement OnRewardCenterOpened and OnRewardCenterClosed to listen to events and RapidoReachSurveyAvailable to listen to when a survey is available.

```
#if UNITY_ANDROID || UNITY_IOS


// Your GameObject that triggers the Reward Center
```

```
  void Start()
  {
    ConfigureRapidoReach();
  }

    public void ConfigureRapidoReach()

  {
#if UNITY_IOS || UNITY_ANDROID
    RapidoReach.Initialize("d5ece53df8ac97409298325fec81f3f7", "ANDROID_TEST_ID");

    RapidoReach.SetListener(gameObject.name);


    // optional
    RapidoReach.SetNavigationBarText("RapidoReach Unity N");

    RapidoReach.SetNavigationBarColor("#211548");

    RapidoReach.SetNavigationBarTextColor("#FFFFFF");
#endif

  }


  // call this function to show reward center on button click etc
  public void showRewardCenter(){
    Debug.Log("UNITY: Calling show reward center");
#if UNITY_IOS || UNITY_ANDROID
    RapidoReach.ShowRewardCenter();
#endif

  }


  void OnReward(string quantity)
  {
    FindObjectOfType<ReceivedRewards>().GetComponent<Text>().text = quantity;

    Debug.Log("RapidoReach OnReward: " + quantity);

  }


    #endif
```

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 dashboardt 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.SetNavigationBarText("RapidoReach Unity N");

RapidoReach.SetNavigationBarColor("#211548");

RapidoReach.SetNavigationBarTextColor("#FFFFFF");
```

Revision #7

Created Tue, Mar 16, 2021 8:19 AM by Sudarshan Kondgekar

Updated Tue, Mar 16, 2021 8:27 AM by Sudarshan Kondgekar