

React Native SDK

@rapidoreachsdk/react-native-rapidoreach

Before you start

Get your API key

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

Getting started

```
$ npm install @rapidoreachsdk/react-native-rapidoreach
```

```
$ yarn add react-native-webview
```

```
$ cd ios && pod install && cd .. # CocoaPods on iOS needs this extra step
```

We are all set up! Now let's use the module.

Usage

Initialize RapidoReach

First, you need to initialize the RapidoReach instance with `initWithApiKeyAndUserId` call.

```
// Import RapidoReach native module
import RapidoReach from '@rapidoreachsdk/react-native-rapidoreach';

componentDidMount() {
  // In your app initialization, initialize RapidoReach
  RapidoReach.initWithApiKeyAndUserId('YOUR_API_TOKEN', 'YOUR_USER_ID');
}
```

Reward Center

Next, 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.

```
onPressShowRewardCenter = () => {
  RapidoReach.isSurveyAvailable((isAvailable) => {
    // if a survey is available, show the reward center
    if (isAvailable) {
      RapidoReach.showRewardCenter();
    }
  })
}
```

Reward Callback

To ensure safety and privacy, we recommend using a server side callback to notify you of all awards. 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

If you do not have a server to handle server side callbacks we additionally provide you with the ability to listen to client side reward notification.

First, import Native Module Event Emitter:

```
import { RapidoReachEventEmitter } from '@rapidoreachsdk/react-native-rapidoreach';
```

Then, add event listener for award notification (in `componentWillMount`, for example):

```
this.onRewardListener = RapidoReachEventEmitter.addListener(  
  'onReward',  
  this.onReward,  
);
```

Implement the callback:

```
onReward = (quantity) => {  
  console.log('reward quantity: ', quantity);  
}
```

Reward Center Events

You can optionally listen for the `onRewardCenterOpened` and `onRewardCenterClosed` events that are fired when your Reward Center modal is opened and closed.

Add event listeners for `onRewardCenterOpened` and `onRewardCenterClosed`:

```
this.onRewardCenterOpenedListener = RapidoReachEventEmitter.addListener(  
  'onRewardCenterOpened',  
  this.onRewardCenterOpened,  
);  
this.onRewardCenterClosedListener = RapidoReachEventEmitter.addListener(  
  'onRewardCenterClosed',  
  this.onRewardCenterClosed,  
);
```

Implement event callbacks:

```
onRewardCenterOpened = () => {  
  console.log('onRewardCenterOpened called!');  
}  
  
onRewardCenterClosed = () => {  
  console.log('onRewardCenterClosed called!');  
}
```

```
}
```

Survey Available Callback

If you'd like to be proactively alerted to when a survey is available for a user you can add this event listener.

First, import Native Module Event Emitter:

```
import { RapidoReachEventEmitter } from '@rapidoreachsdk/react-native-rapidoreach';
```

Then, add event listener for award notification (in `componentWillMount`, for example):

```
this.rapidoreachSurveyAvailableListener = RapidoReachEventEmitter.addListener(  
  'rapidoreachSurveyAvailable',  
  this.rapidoreachSurveyAvailable,  
);
```

Implement the callback:

```
rapidoreachSurveyAvailable = (surveyAvailable) => {  
  if (surveyAvailable == "true") {  
    console.log('rapidoreach survey is available');  
  } else {  
    console.log('rapidoreach survey is NOT available');  
  }  
}
```

Finally, don't forget to remove your event listeners in the `componentWillUnmount` lifecycle method:

```
componentWillUnmount() {  
  this.onRewardListener.remove();  
  this.onRewardCenterOpenedListener.remove();  
  this.onRewardCenterClosedListener.remove();  
  this.rapidoreachSurveyAvailableListener.remove();  
}
```

Contact

Please send all questions, concerns, or bug reports to admin@rapidoreach.com.

FAQ

What do you do to protect privacy?

We take privacy very seriously. All data is encrypted before being sent over the network. We also use HTTPS to ensure the integrity and privacy of the exchanged data.

What kind of analytics do you provide?

Our dashboard will show metrics for sessions, impressions, revenue, and much more. We are constantly enhancing our analytics so we can better serve your needs.

What is your fill rate?

We have thousands of surveys and add hundreds more every day. Most users will have the opportunity to complete at least one survey on a daily basis.

I'm ready to go live! What are the next steps?

Let us know! We'd love to help ensure everything flows smoothly and help you achieve your monetisation goals!

Following the rewarded and/or the Offerwall approach

An example is provided on Github that demonstrates how a publisher can implement the rewarded and/or the Offerwall approach. Upon survey completion, the publisher can reward the user.

Limitations / Minimum Requirements

This is just an initial version of the plugin. There are still some limitations:

- You cannot pass custom attributes during initialization

- No tests implemented yet
- Minimum iOS is 9.0 and minimum Android version is 16

For other RapidoReach products, see RapidoReach docs.

ReactNativeSDK

Revision #3

Created 2021-03-16 08:32:25 UTC by Sudarshan Kondgekar

Updated 2021-03-16 08:34:54 UTC by Sudarshan Kondgekar