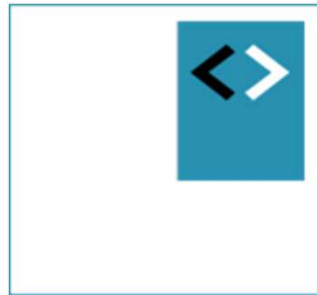


**PROUD
NERDS**

React Native Module – Expo APIs



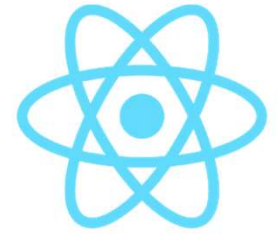
Peter Kassenaar –
info@kassenaar.com



Expo Device APIs

Native API's exposed as Expo modules that you can install/use

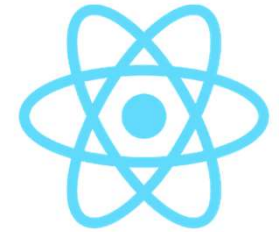
What is the Expo API?



*"The Expo SDK provides **access to device and system functionality** such as contacts, camera, gyroscope, GPS location, and so on, in the form of **packages**."*

*You can install any Expo SDK package using the **`npx expo install <package-name>`** command."*

Documentation online

A screenshot of the Expo documentation website. The left sidebar shows a navigation menu with 'API Reference' highlighted and 'SDK 47 (Latest)' selected. The main content area is titled 'API Reference' and contains text about the Expo SDK, a terminal snippet for installing packages, and a code block for importing modules. The 'SDK 47 (Latest)' option in the sidebar is circled in red.

Expo > Docs

Search Ctrl K Auto

Guides

Expo Application Services

API Reference

SDK 47 (Latest)

Configuration files

app.json / app.config.js

metro.config.js

Expo SDK

Accelerometer

AppleAuthentication

Application

Asset

AsyncStorage

Audio

AuthSession

AV

BackgroundFetch

API Reference

The Expo SDK provides access to device and system functionality such as contacts, camera, gyroscope, GPS location, and so on, in the form of packages. You can install any Expo SDK package using the `npx expo install` command. For example, three different packages are installed using the following command:

```
Terminal
```

```
→ npx expo install expo-camera expo-contacts expo-sensors
```

After installing one or more packages, you can import them into your JavaScript code:

```
import { Camera } from 'expo-camera';
import * as Contacts from 'expo-contacts';
import { Gyroscope } from 'expo-sensors';
```

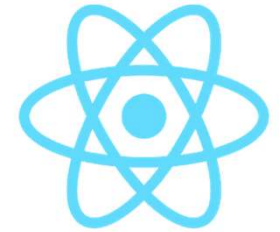
This allows you to write `Contacts.getContactsAsync()` and read the contacts from the device, read the gyroscope sensor to detect device movement, or start the phone's camera and take photos.

These packages work in bare React Native apps too

The easiest way to create a bare React Native app with support for the Expo SDK is by running the command:

<https://docs.expo.dev/versions/latest/>

Mind the versioning!

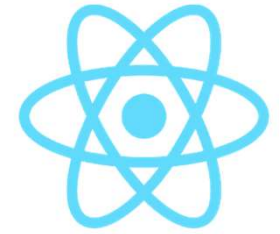


Make sure to match versions, otherwise it will (probably) not work!

Each Expo SDK version depends on a React Native version

Every quarter there is a new Expo SDK release that typically updates to the latest stable versions of React Native and React Native Web, and includes a variety of bug fixes, features, and improvements to the Expo SDK.

Expo SDK version	React Native version	React Native Web version
47.0.0	0.70.5	0.18.9
46.0.0	0.69.6	0.18.7
45.0.0	0.68.2	0.17.7
44.0.0	0.64.3	0.17.1



What's in the Expo API?

- JavaScript access to device API's that were previously only available via native (Java/Swift) code.
- *A LOT* more than in the early PhoneGap/Cordova-days.

Expo > Docs

Search Ctrl | K Auto

Guides

Expo Application Services

API Reference

Expo SDK

- Accelerometer
- AppleAuthentication
- Application
- Asset
- AsyncStorage
- Audio
- AuthSession
- AV
- BackgroundFetch
- BarcodeScanner
- Barometer
- Battery
- BlurView
- Brightness
- BuildProperties
- Calendar
- Camera
- captureRef
- Cellular
- Checkbox
- Clipboard

The Expo SDK provides access to device and system functionality such as contacts, camera, gyroscope, GPS location, and so on, in the form of packages. You can install any Expo SDK package using the `npx expo install` command. For example, three different packages are installed using the following command:

```
Terminal
~ npx expo install expo-camera expo-contacts expo-sensors
```

After installing one or more packages, you can import them into your JavaScript code:

```
import { Camera } from 'expo-camera';
import * as Contacts from 'expo-contacts';
import { Gyroscope } from 'expo-sensors';
```

This allows you to write `Contacts.getContactsAsync()` and read the contacts from the device, read the gyroscope sensor to detect device movement, or start the phone's camera and take photos.

These packages work in bare React Native apps too

The easiest way to create a bare React Native app with support for the Expo SDK is by running the command:

```
Terminal
# Create a project named my-app
~ npx create-expo-app my-app --template bare-minimum
```

Existing apps
Projects that were created with `npx react-native init` require additional setup to use the Expo SDK. →

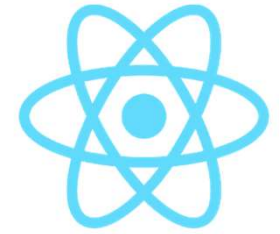
Using libraries
Learn how to install Expo SDK packages in your project. →

Most of the API's work *async*. It comes in handy if you know a little bit about Promises and/or *async/await*.



General workflow

Installing and using expo api's



General workflow

1. Install the api inside your project

- Like: `npx expo install expo-battery`
- `npx expo install expo-sensors, etc.`

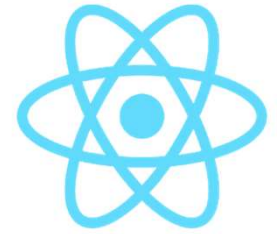
2. You can install multiple modules at once, e.g.

`npx expo install expo-battery expo-sensors expo-camera ...`

3. Read the Usage section in docs, e.g.

- Import the correct modules (like `import * as Battery from 'expo-battery'`)
- Initialize variables in state and/or use API using `useState(...)`
- See examples in docs or blogs

Questions?



Some examples



Battery

Reading the battery level from device and (charging) status

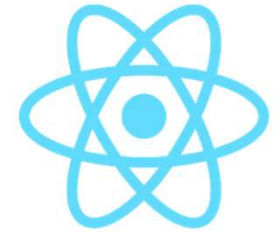
Battery API



- The package `expo-battery` provides **battery information for the physical device**
 - battery level, whether or not the device is charging, and more
 - as well as corresponding **event listeners**.
- Not available on emulators or desktop-pc's

Platform Compatibility

Android Device	Android Emulator	iOS Device	iOS Simulator	Web
✓	✓	✓	✗	✓



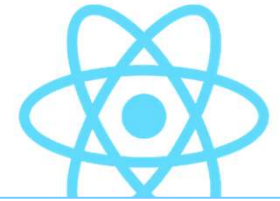
Installing package

```
npx expo install expo-battery
```

```
> Stopped server
PS C:\Users\Gebruiker\Desktop\myExpoProject> npx expo install expo-battery
> Installing 1 SDK 47.0.0 compatible native module using npm
> npm install
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wa
t: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev
```

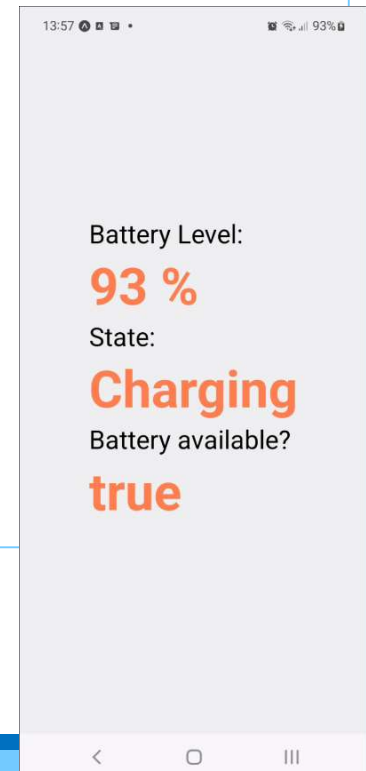
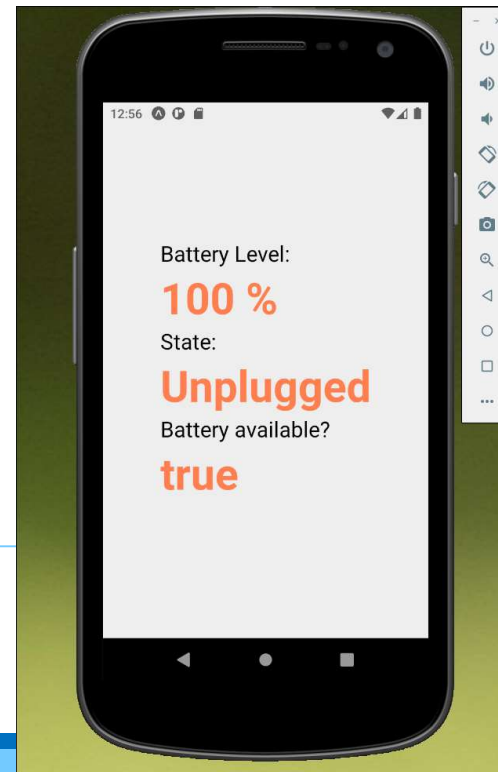
```
{
  //package.json
  "version": "x.x.x",
  "dependencies": {
    "expo": "~47.0.8",
    "expo-battery": "~7.0.0"
    ...
  },
}
```

Example on using Battery

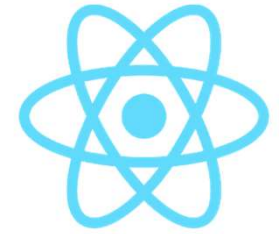


```
// BatteryApi.js
...
useEffect(() => {
  // 2. Using an async function inside useEffect, and calling it immediately.
  (async () => {
    const [isAvailable, batteryLevel, batteryState, lowPowerMode] = await Promise.all([
      Battery.isAvailableAsync(),
      Battery.getBatteryLevelAsync(),
      Battery.getBatteryStateAsync(),
      Battery.isLowPowerModeEnabledAsync(),
    ]);

    // 3. await all the above Promises,
    // then set the state variables:
    setIsAvailable(isAvailable);
    setBatteryLevel(batteryLevel);
    setBatteryState(batteryState);
    setLowPowerMode(lowPowerMode);
  })();
}
...
```



Demo code



- See example in `../components/BatteryApi.js`.

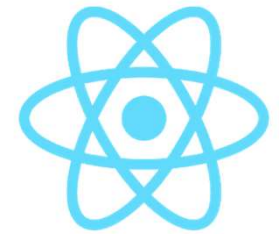
```
<View style={styles.container}>
  {/*Only return status etc. if battery is available*/}
  {isAvailable
    ?
    <View>
      <Text style={styles.label}>Battery Level:</Text>
      <Text style={styles.value}>{Math.round(batteryLevel * 100)} %</Text>
      <Text style={styles.label}>State:</Text>
      <Text style={styles.value}>{getBatteryStateString(batteryState)}</Text>
      <Text style={styles.label}>Battery available?</Text>
      <Text style={styles.value}>{isAvailable.toString()}</Text>
    </View>
    : <Text> 'Battery API is not supported on this device' </Text>}
</View>
```



Geolocation

Read the GPS-status from the device

API's with additional permissions



- Some API's require **additional permissions** from the user
 - Access to location,
 - Access to camera
 - Access to contacts, and so on
- Most of the time you **don't have to set this manually**
 - App requests it upon first usage
 - Some functions work only in submitted apps and approval by Google|Apple

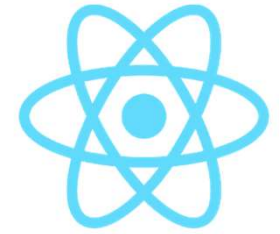
Android permissions

When you install the `expo-location` module, it automatically adds the following permissions:

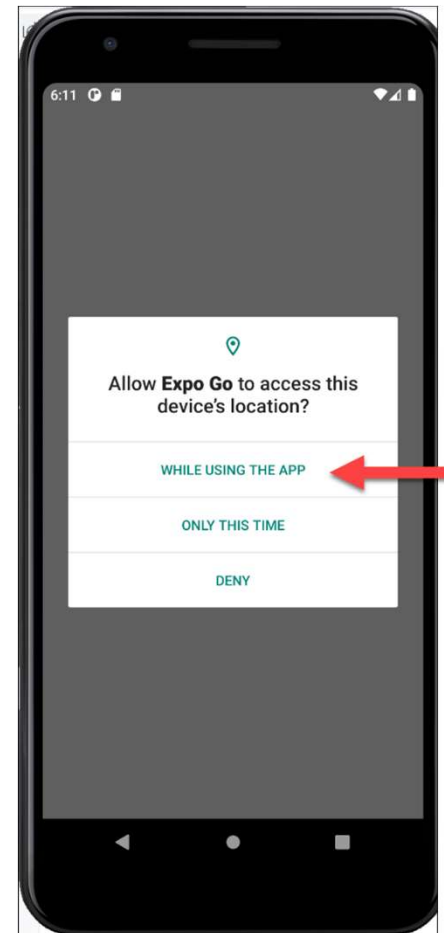
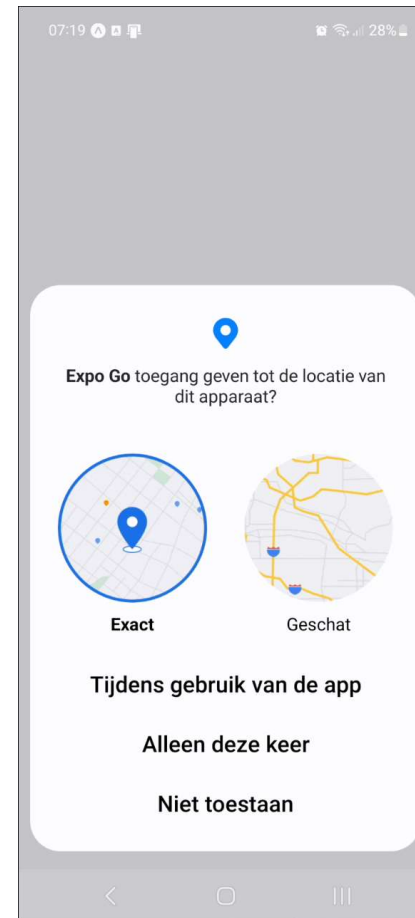
- `ACCESS_COARSE_LOCATION` : for approximate device location
- `ACCESS_FINE_LOCATION` : for precise device location
- `FOREGROUND_SERVICE` : to subscribe to location updates while the app is in use

To use background location features, you must add the `ACCESS_BACKGROUND_LOCATION` in `app.json` and [submit your app for review and request access to use the background location permission](#).

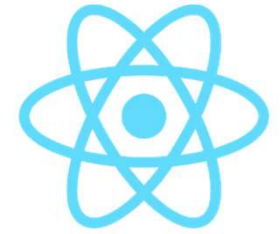
Location



- Location requires permission – you have to prepare your app for denial
- Use the `useEffect()` hook for that
- See `<LocationApi />` for example code

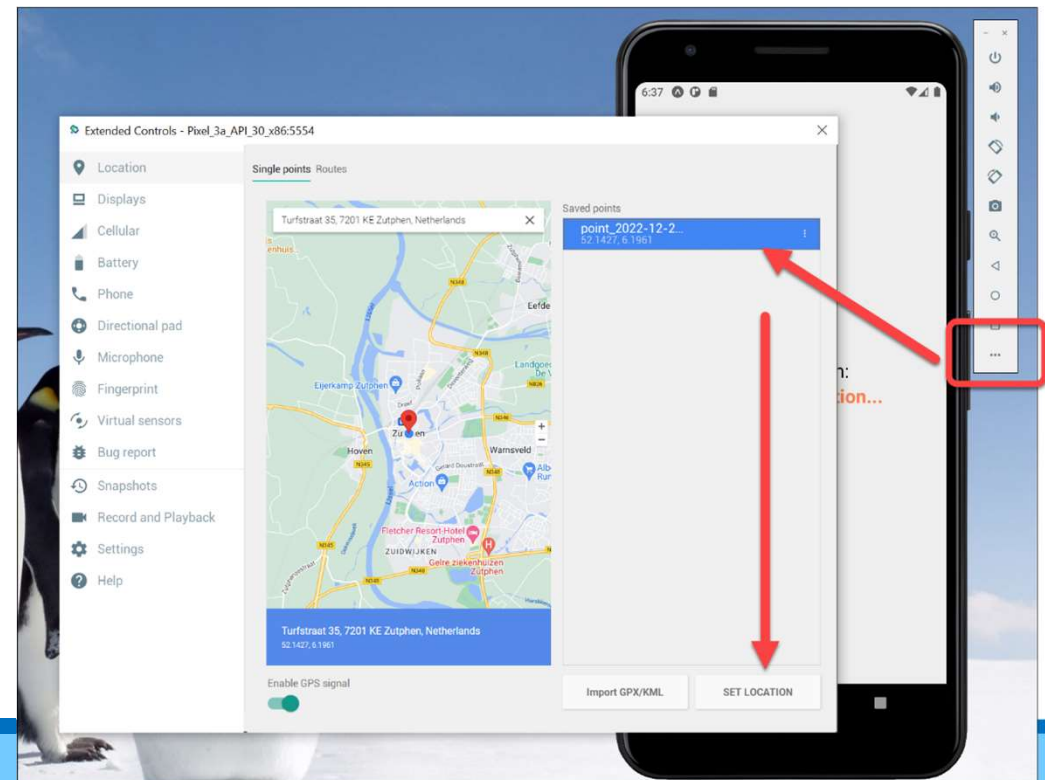
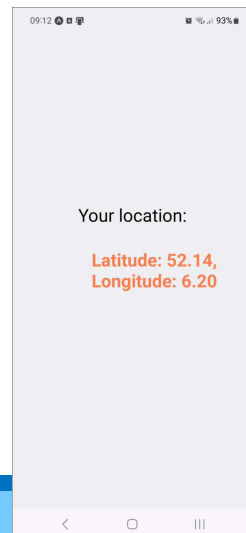


Enable location in emulator



- In iPhone Simulator and Android emulator you have to explicitly enable location services
- Depends on your platform. For Android:
- Resetting perms:

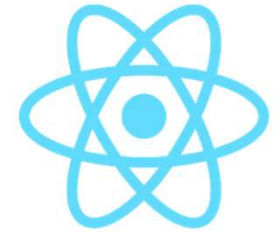
```
adb shell pm reset-permissions
```





BarCodeScanner

Using the camera to scan barcodes and QR-codes



BarCodeScanner

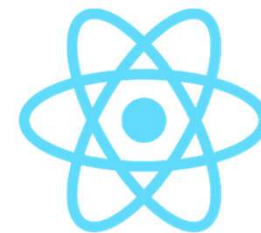
- Scan barcodes and QR-codes
- Request Camera permission on startup
- Supported by all devices
- `npx expo install expo-barcode-scanner`

`expo-barcode-scanner` provides a React component that renders a viewfinder for the device's camera (either front or back) and will scan bar codes that show up in the frame.

Platform Compatibility

Android Device	Android Emulator	iOS Device	iOS Simulator	Web
✓	✓	✓	✓	✗

Again – work mainly async

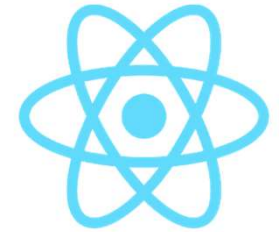


```
// 1. useEffect, getting (camera) permission for the barcodescanner one time only
useEffect(() => {
  const getBarCodeScannerPermissions = async () => {
    // 1a. Destructuring the status in a variable
    const {status} = await BarCodeScanner.requestPermissionsAsync();
    // 1b. Set true or false to hasPermission.
    setHasPermission(status === 'granted');
  };

  // 1c. Get the permissions on startup
  getBarCodeScannerPermissions();
}, []);
```

../components/BarCodeScannerApi.js

General usage



- Always look at:
 - Installation instructions
 - Example
 - API options & props

Component

BarCodeScanner

Type: `Component<BarCodeScannerProps>`

Props

barCodeTypes

Optional • Type: `string[]`

An array of bar code types. Usage: `BarCodeScanner.Constants.BarCodeType.<codeType>` where `codeType` is one of these [listed above](#). Defaults to all supported bar code types. It is recommended to provide only the bar code formats you expect to scan to minimize battery usage.

For example: `barCodeTypes=[BarCodeScanner.Constants.BarCodeType.qr]`.

onBarCodeScanned

Optional • Type: `BarCodeScannedCallback`

A callback that is invoked when a bar code has been successfully

On this page

- Installation
- Configuration
- Supported formats
- Usage
- API

Component

- BarCodeScanner

Props

- barCodeTypes
- onBarCodeScanned
- type

Inherited Props

-

Hooks

- usePermissions()

Methods

- getPermissionsAsync()
- requestPermissionsAsync()
- scanFromURLAsync()

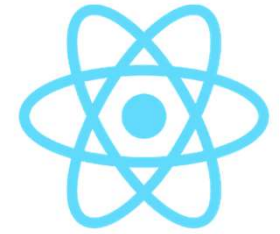
Interfaces

- PermissionResponse

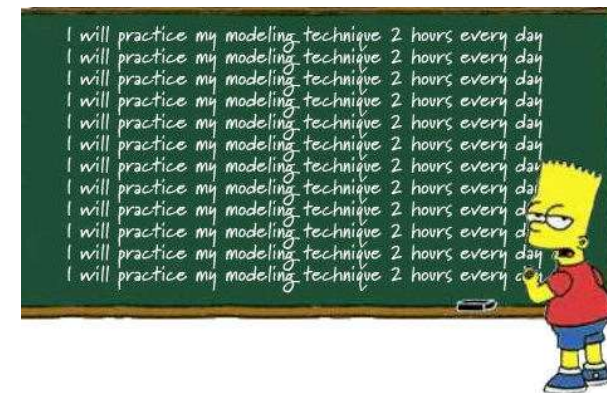
Types

- BarCodeBounds
- BarCodeEvent
- BarCodeEventCallbackArg...

Workshop



- 1. Open the example `../100-expo-apis` and use the different components to see the workings of
 - Battery
 - Location
 - BarCodeScanner
- 2. Add these (or other) expo API's to your own project and test them. For instance:
 - Accelerometer
 - Audio (`expo-av`)
 - Camera
 - DateTimePicker, ...
- <https://docs.expo.dev/versions/latest/>

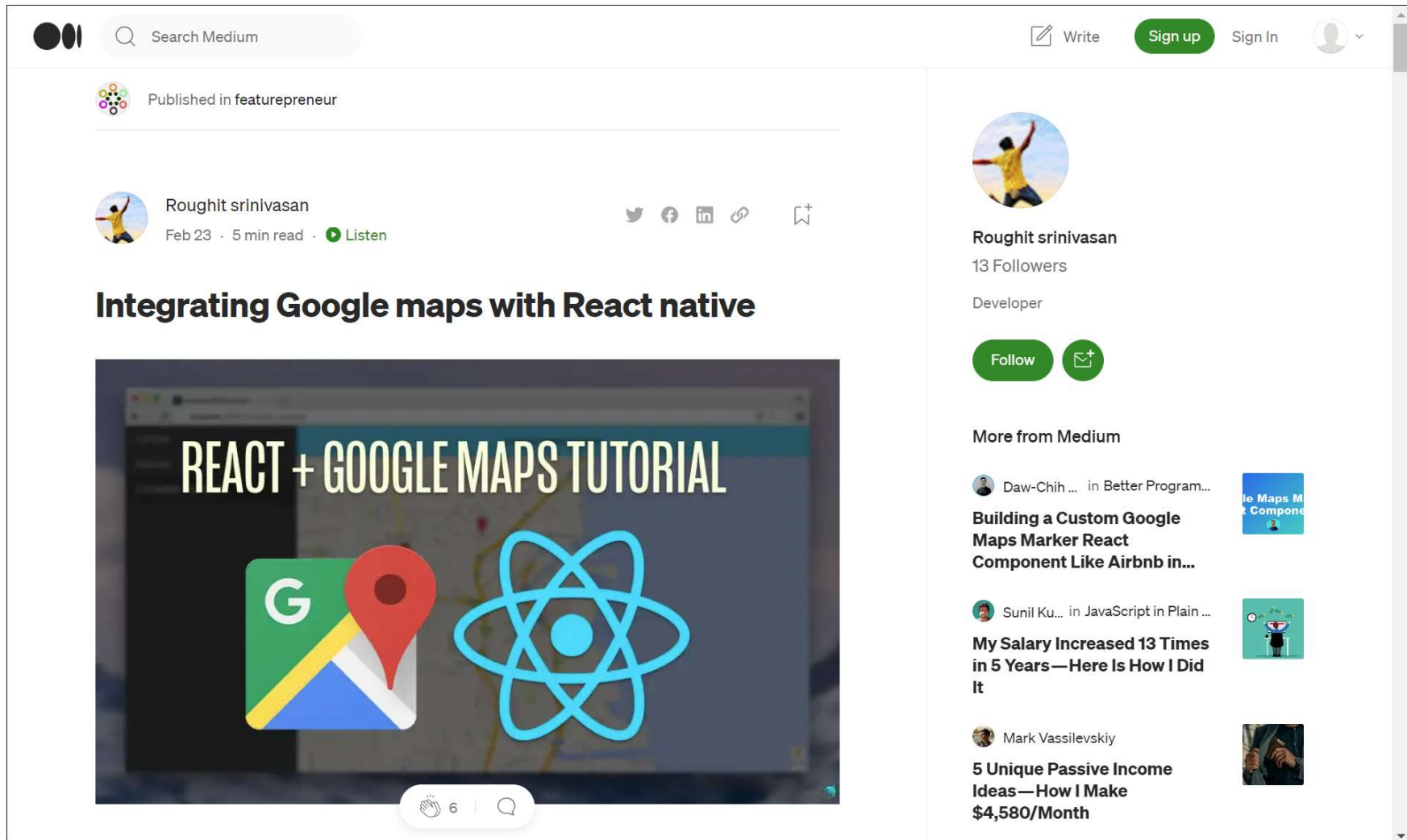
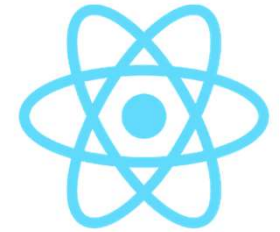




More info

More info on the topics in this module

Integrating Google Maps

A screenshot of a Medium article page. The article is titled "Integrating Google maps with React native" by Roughit srinivasan, published in the "featurepreneur" collection. The article's featured image shows a Google Maps interface with the text "REACT + GOOGLE MAPS TUTORIAL" overlaid, along with the Google Maps logo and the React logo. The article has 6 claps. On the right, the author's profile is shown with a "Follow" button and a list of "More from Medium" articles, including "Building a Custom Google Maps Marker React Component Like Airbnb in..." and "My Salary Increased 13 Times in 5 Years—Here Is How I Did It".

Search Medium

Published in featurepreneur

Roughit srinivasan
Feb 23 · 5 min read · Listen

Integrating Google maps with React native

REACT + GOOGLE MAPS TUTORIAL

6

Roughit srinivasan
13 Followers
Developer

Follow

More from Medium

Daw-Chih ... in Better Program...
Building a Custom Google Maps Marker React Component Like Airbnb in...

Sunil Ku... in JavaScript in Plain ...
My Salary Increased 13 Times in 5 Years—Here Is How I Did It

Mark Vassilevskiy
5 Unique Passive Income Ideas—How I Make \$4,580/Month

<https://medium.com/featurepreneur/integrating-google-maps-with-react-native-62fc8b7ecded>

Camera




freeCodeCamp (🔥) Forum Donate

Learn to code — free 3,000-hour curriculum

OCTOBER 21, 2020 / #REACT NATIVE

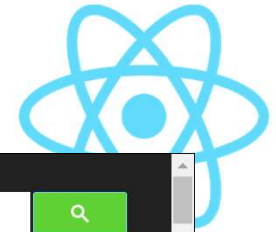
How to Create a Camera App with Expo and React Native

 Said Hayani



<https://www.freecodecamp.org/news/how-to-create-a-camera-app-with-expo-and-react-native/>

Sending SMS



**ABOUTREACT**

ENHANCED BY Google

HomeJust StartedBeginnerAdvancePro LevelUseful ComponentsNavigation ExamplesFrequent IssuesOnline EmulatorUseful ▾

Hey! We are open for any offer. Read More..

React Navigation

- Stack Navigation - React Navigation
- Navigation Drawer - React Navigation
- Bottom Navigation - React Navigation
- Navigation
- Tab - React Navigation
- Swipe Gestures not Working in Android
- RN Android Fragment

More Navigation Examples

Android Only

- Handle Android Back Button
- Ask Run Time Android Permission

Networking & Properties

- Make HTTP Request/ RN Fetch
- Networking using Axios
- Call GraphQL Query, Mutation and Subscription

Example to Send Text SMS on Button Click in React Native

Table of Contents [hide]

- 1 Introduction
- 2 To Send SMS
- 3 To Make a React Native App
- 4 Installation of Dependency
- 5 CocoaPods Installation
- 6 Permission to Send SMS for Android
- 7 Code to Send Text SMS
 - 7.1 App.js
- 8 To Run the React Native App
- 9 Output Screenshots

Introduction

<https://aboutreact.com/send-text-sms-in-react-native/>