

# Popup Coding Meetup

At Whole Food -Taylor Rd- Today  
Thursday Jan 9<sup>th</sup> 6:30PM

---

Building Iphone/Android Apps  
with React Native



# First Thing - React Native And Expo:

from: <https://hackernoon.com/understanding-expo-for-react-native-7bf23054bbcd>

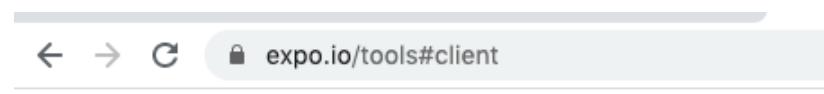
- A React Native app is a compiled app that is running some Javascript.
- Whenever you build and run your React Native project, a packager starts up called Metro. You've probably seen this output in your terminal before, letting you know the packager is running.
- The packager does a few things:
  1. Combines all your Javascript code into a single file, and translates any Javascript code that your device won't understand (like JSX or some of the newer JS syntax).
  2. Converts assets (e.g. PNG files) into objects that can be displayed by an Image component.
- <https://facebook.github.io/react-native/>
- <https://expo.io/learn>

Let Build  
Something



# Install Expo App on your Phone.

**<https://expo.io/tools#client>**



## Expo Client for Android & iOS

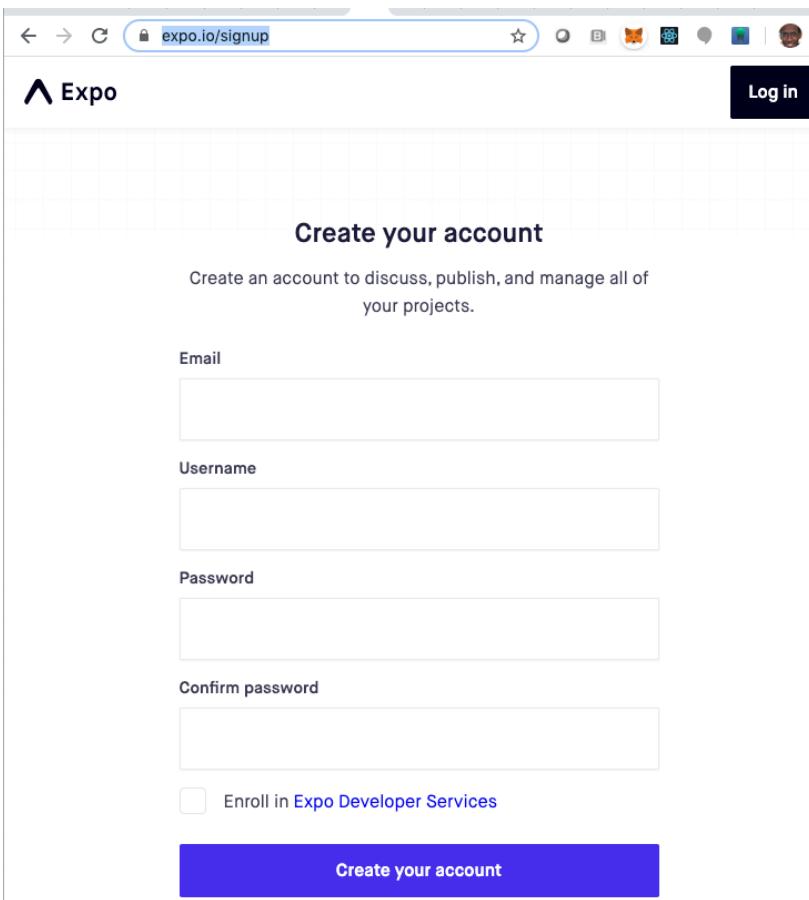
Run your projects before you deploy. Open projects by scanning QR codes. If you need to, [Download IPA 2.14.0](#) or [Download APK 2.14.0](#).



# In Web Browser on your Laptop or desktop

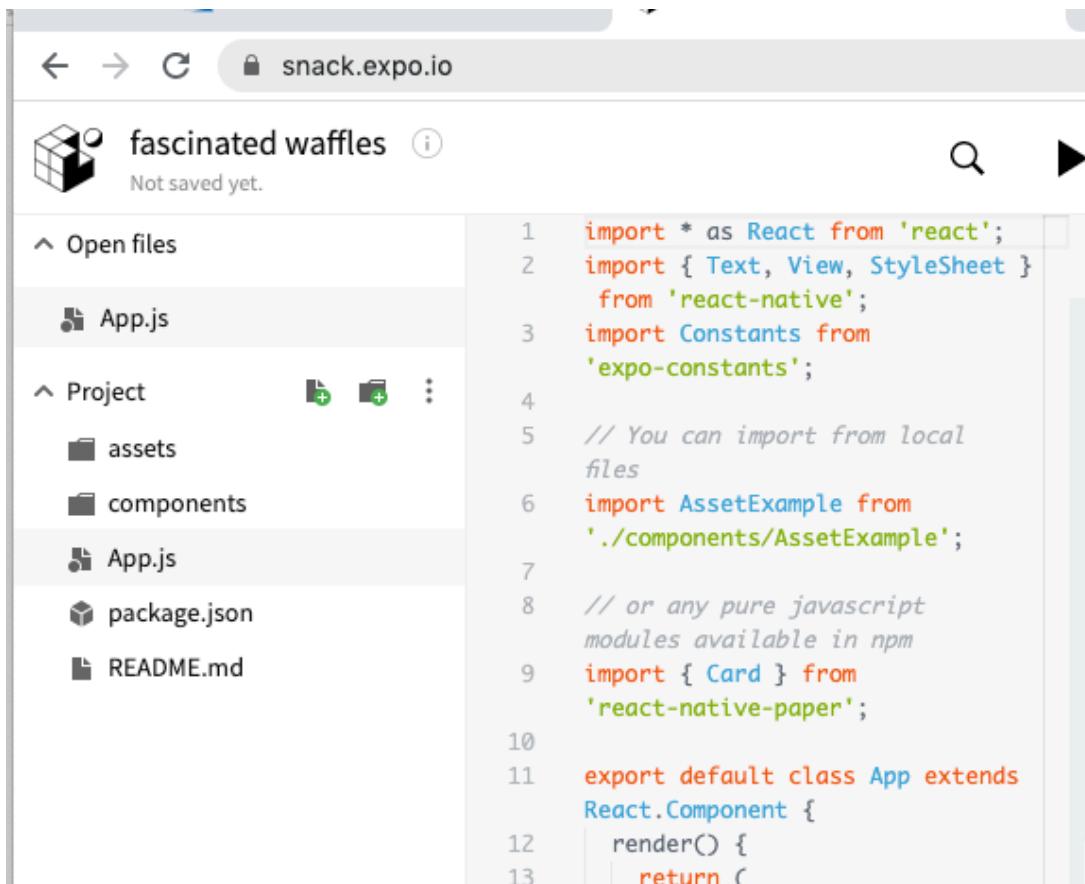
## Create Expo Account

<https://expo.io/signup>



# In Web Browser Go To Snacks URL

## <https://snack.expo.io/>



The screenshot shows a web browser window with the URL [snack.expo.io](https://snack.expo.io) in the address bar. The page title is "fascinated waffles" with a note "Not saved yet." Below the title is a sidebar with "Open files" and "Project" sections. The "Open files" section contains "App.js". The "Project" section contains "assets", "components", another "App.js" (which is selected), "package.json", and "README.md". The main area displays a portion of the "App.js" code:

```
1 import * as React from 'react';
2 import { Text, View, StyleSheet }
3   from 'react-native';
4 import Constants from
5   'expo-constants';
6 // You can import from local
7 // files
8 import AssetExample from
9   './components/AssetExample';
10 // or any pure javascript
11 // modules available in npm
12 import { Card } from
13   'react-native-paper';
14
15 export default class App extends
16   React.Component {
17   render() {
18     return (
19       <View>
20         <Text>Hello world</Text>
21       </View>
22     );
23   }
24 }
```

# Run Snacks App

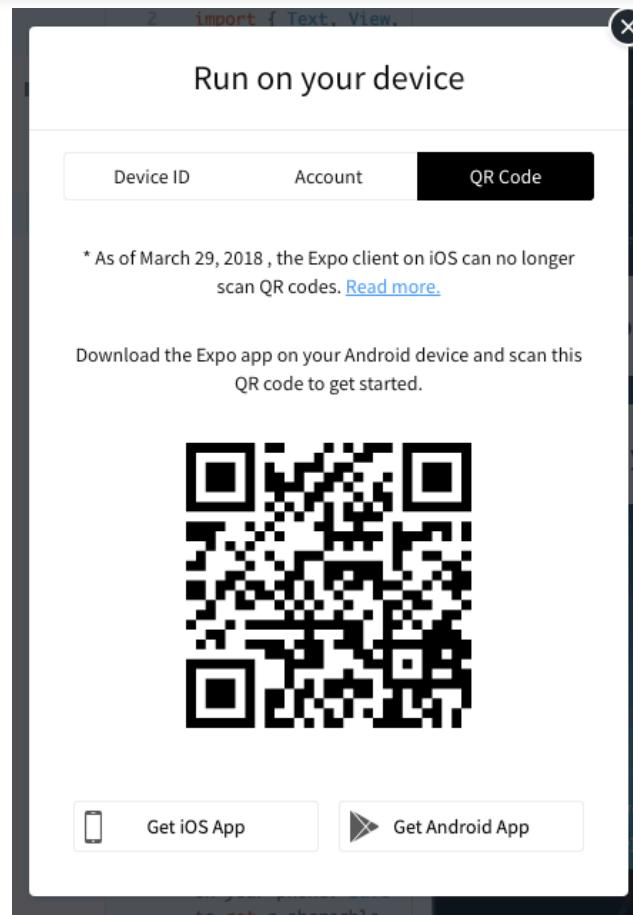
Select Phone Type or Web

The screenshot shows the snack.expo.io interface. At the top, there's a navigation bar with back, forward, and refresh buttons, followed by a URL field containing "snack.expo.io". To the right of the URL are icons for star, search, and other settings. Below the URL is a project title "fascinated waffles" with a note "Not saved yet." On the left, there's a sidebar titled "Open files" showing "App.js" is open. Below that is a "Project" section with "assets" and "components" folders. The main area is a code editor with the following code:

```
1 import * as React
  from 'react';
2 import { Text, View,
  StyleSheet } from
  'react-native';
3 import Constants
  from
  'expo-constants';
4
5 // You can import
  from local files
```

To the right of the code editor are three tabs: "iOS", "Android", and "Web", with "Web" currently selected. There's also a "Save" button and a user profile icon.

# On Phone bring up QR reader And load app on Phone.



**On Phone You should See This  
Next we will change code a little  
and the app will change in real time.**

**Change code in the editor and  
watch it change on your  
phone! Save to get a  
shareable url.**

**Local files and assets can be  
imported by dragging and  
dropping them into the editor**



# Go back to Web Browser and Modify Code

<https://snack.expo.io>

The screenshot shows a code editor interface for a React Native project named "fascinated waffles". The left sidebar lists "Open files" (App.js) and "Project" files (assets, components, App.js, package.json, README.md). The main area displays the following code:

```
1 import * as React from 'react';
2 import { Text, View, StyleSheet } from 'react-native';
3 import Constants from 'expo-constants';
4
5 // You can import from local files
6 import AssetExample from './components/AssetExample';
7
8 // or any pure javascript modules available in npm
9 import { Card } from 'react-native-paper';
10
11 export default class App extends React.Component {
12   render() {
13     return (
14       <View style={styles.container}>
15         <Text style={styles.paragraph}>
16           I Changed this I am building Iphone Apps Smiles.
17         </Text>
18     </View>
19   }
20 }
```

A callout bubble on the right side points to the text "I Changed this I am building Iphone Apps Smiles." with the label "Update Code in Text Section". The entire "View" component at the bottom of the code is highlighted with a red dashed rectangle.

# Run Snacks App

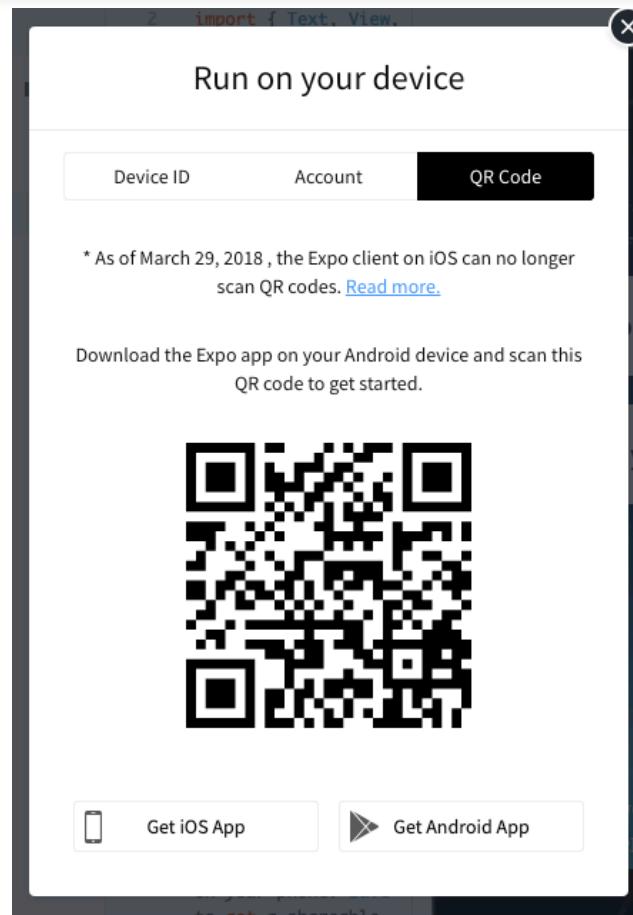
Select Phone Type or Web

The screenshot shows the snack.expo.io interface. At the top, there's a navigation bar with back, forward, and refresh buttons, followed by a URL field containing "snack.expo.io". To the right of the URL are icons for star, search, and other settings. Below the URL is a project title "fascinated waffles" with a note "Not saved yet." On the left, there's a sidebar titled "Open files" showing "App.js" is selected. Below that is a "Project" section with "assets" and "components" folders. The main area is a code editor with the following code:

```
1 import * as React
from 'react';
2 import { Text, View,
  StyleSheet } from
'react-native';
3 import Constants
from
'expo-constants';
4
5 // You can import
from local files
```

To the right of the code editor are three tabs: "iOS", "Android", and "Web", with "Web" currently selected. There's also a "Save" button and a user profile icon.

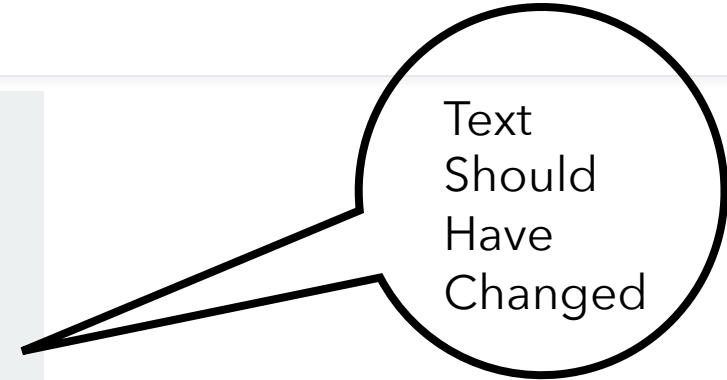
# On Phone bring up QR reader And load app on Phone.



# On Phone You should See This

## Next we will change code a little and the app will change in real time.

Check Your Phone  
It will update automatically.



Yes it is  
that Easy

---



# Expo - Hybrid App Development

- Hybrid Apps a combination of a web app and a native container app.
- A hybrid app fulfils its functionality using UIWebView in IOS and WebView on Android. In a hybrid app, the coding for the app is done on a web-based platform like HTML, CSS etc.
- It is a prerequisite requirement you download a host client.
- Hybrid apps are a cross-platform development for smartphones.

# Expo Benefits

<https://apiko.com/blog/expo-vs-vanilla-react-native/>

- **Fast and simple project** installation (in just several minutes)
- **A convenient utility Expo CLI that opens in a browser and helps to check the app status**, devices it runs on, scan QR code or send the link via email to open the app in Expo client, switch the production / development mode and publish your app on Expo server.
- **Expo client is an app that is installed from Google Play and Apple Store on your phone**. It allows opening projects during development without build via XCode or Android Studio. With Expo client, you can send your app to others for review, which is very useful when testing as you can see all changes in code in Expo client without creating apk or ipa files.
- **Over the Air - very handy Expo** feature for updating your app over the air without repeated deployment on Google Play or Apple Store. When users open your app, it will automatically update new changes in JavaScript code.

# Expo drawbacks

<https://apiko.com/blog/expo-vs-vanilla-react-native/>

- You can't add native modules written in Objective-C, Swift, Java, Kotlin
- You can't use packages with native languages that require linking
- The app has a big size as it is built with all Expo SDK solutions, even those you don't use. An application with Hello World weighs 25 MB
- Often everything works well in Expo client during testing, but certain problems may occur in a standalone app.

***NOTE: This does not apply to you when you are just trying to learn ReactJs Native and Testing Among Friends.***