Integration of React Native with Snyk: Report

Introduction

<hr>

This report provides detailed documentation on integrating Snyk, a security tool, with React Native, a popular framework for building mobile applications. By integrating Snyk into your React Native project, you can enhance the security posture of your application by identifying and fixing vulnerabilities in third-party dependencies.

Table of Contents

Background

<hr>

React Native is a JavaScript framework for building native mobile applications. Snyk is a security platform that helps developers find and fix vulnerabilities in their code and dependencies. Integrating Snyk with React Native allows developers to scan their projects for security vulnerabilities and receive actionable insights to improve the security of their applications.

<hr>

Prerequisites

<hr>
Before integrating Snyk with React Native, ensure you have the following prerequisites:

Node.js installed on your development machine
React Native project set up and running
Snyk account created (you can sign up at Snyk website)

<hr>

Integration Steps

Step 1: Install Snyk CLI

```
npm i -g snyk
```

Step 2: Authenticate Snyk CLI

```
snyk auth
```

Follow the prompts to authenticate with your Snyk account.

Follow the prompts to integrate Snyk with your React Native project.

<hr>

Testing

<hr>

After integrating Snyk with your React Native project, you can run security scans to identify vulnerabilities:

```
snyk code test
```

Snyk will analyze your project's dependencies and report any security vulnerabilities found.

<hr>

Conclusion

<hr>

Integrating Snyk with React Native is a straightforward process that can significantly enhance the security of your mobile applications. By regularly scanning your projects for vulnerabilities, you can proactively address security issues and mitigate potential risks.

<hr>

References

<hr>

Snyk Documentation (https://docs.snyk.io/)
<br>
React Native Documentation (https://reactnative.dev)

React Native + snyk

See Introduction here (./INTRO.md)

<hr>

Prerequisites for Windows

Android

<hr>

```
Note: IOS can't be run or build in windows, Mac is required
```

1. Install node for Windows:

Install node [here (https://nodejs.org/en)](https://nodejs.org/en)

2. Install Android studio

Install android studio [here (https://developer.android.com/studio/index.html)](https://developer.android.com/studio/index.html), and follow google's instructions

3. Set Env
    1. Open the Windows Control Panel.
    2. Click on User Accounts, then click User Accounts again
    3. Click on Change my environment variables
    4. Click on New... to create a new ANDROID_HOME user variable that points to the path to your Android SDK: %LOCALAPPDATA%\Android\Sdk
    5. Open the Windows Control Panel.
    6. Click on User Accounts, then click User Accounts again
    7. Click on Change my environment variables
    8. Select the Path variable.
    9. Click Edit.
    10. Click New and add the path to platform-tools to the list. %LOCALAPPDATA%\Android\Sdk\platform-tools

4. Reboot

Prerequisites for Linux

Android

<hr>

```
Note: IOS can't be run or build in linux, Mac is required
```

1. Install node for Linux:

Install node for your own distro [here (https://nodejs.org/en/download/package-manager/)](https://nodejs.org/en/download/package-manager/)

2. Install Android studio

Install android studio [here (https://developer.android.com/studio/index.html)](https://developer.android.com/studio/index.html), and follow google's instructions

3. Setup ENV

Append this to .bashrc

```
export ANDROID_HOME=$HOME/Android/Sdk
export PATH=$PATH:$ANDROID_HOME/emulator
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

4. Reboot

Prerequisites for Mac

Android

<hr>

1. Install node and watchman:

```
brew install node
brew install watchman
```

2. Install jdk

```
brew install --cask zulu@17
brew info --cask zulu@17
```

3. Install Android studio

Install android studio [here (https://developer.android.com/studio/index.html)](https://developer.android.com/studio/index.html), and follow google's instructions

Append this to .zshrc

```
export ANDROID_HOME=$HOME/Library/Android/sdk
export PATH=$PATH:$ANDROID_HOME/emulator
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

Prerequisites for Mac

IOS

<hr>

1. Install node and watchman:

```
brew install node
brew install watchman
```

2. Install XCode

Install XCode from app store

3. Reboot# Report

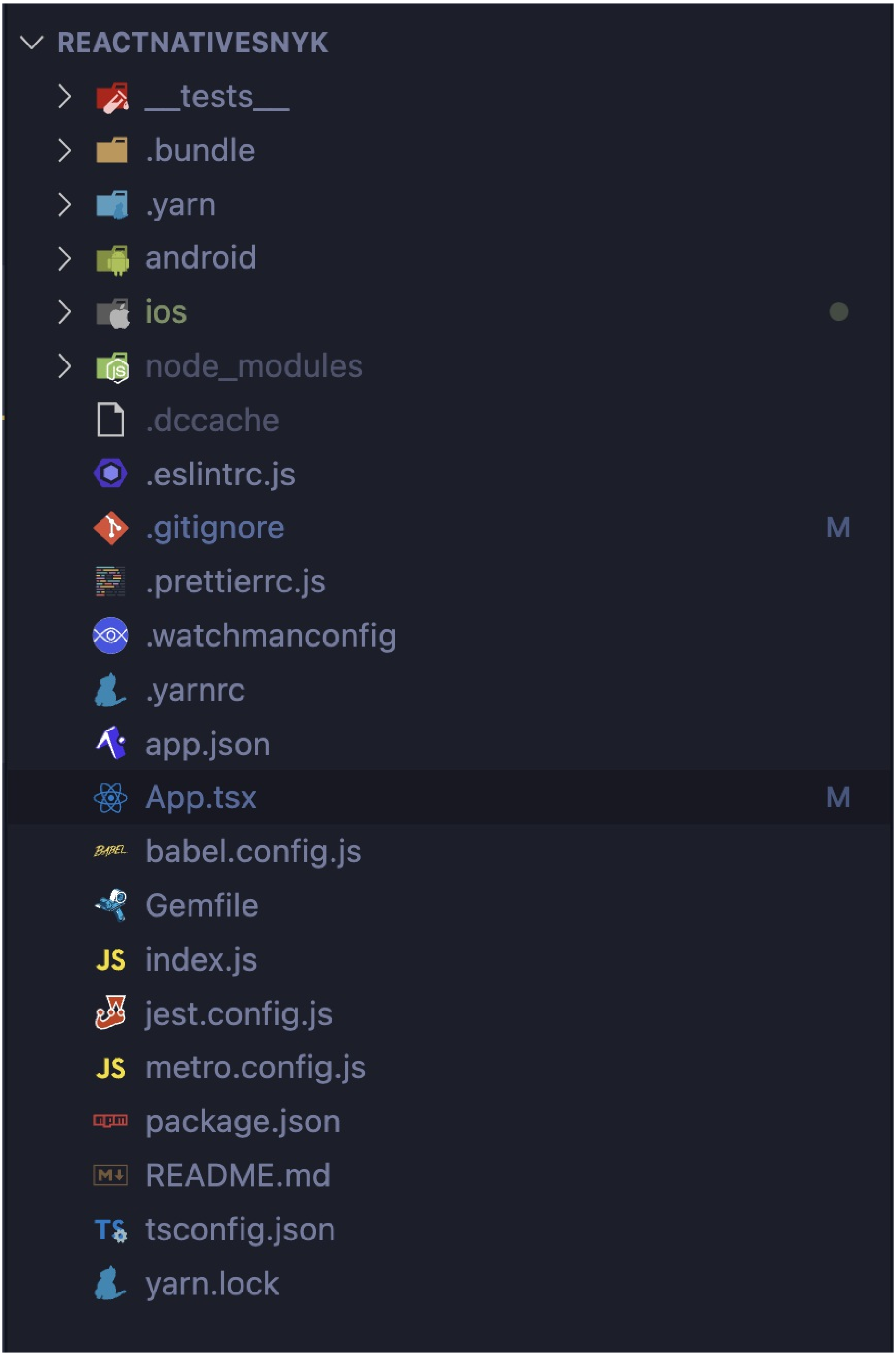
See Introduction [here (./INTRO.md)](./INTRO.md)
See Setup GUIDE [here (./SETUP.md)](./SETUP.md)

<hr>

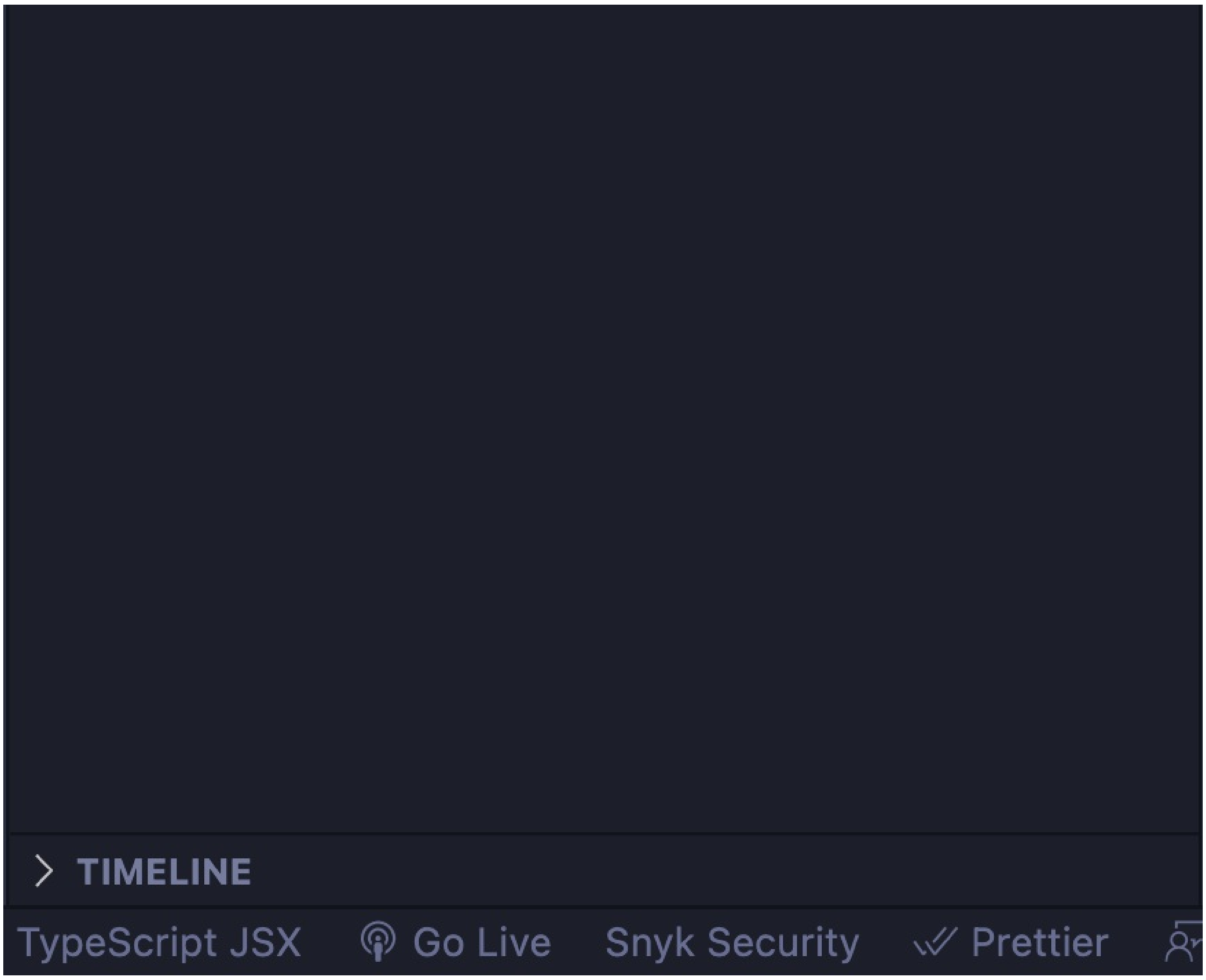
I have created React Native app named ReactNativeSnyk

```
npx react-native@latest init ReactNativeSnyk
```

Here is my project structure:



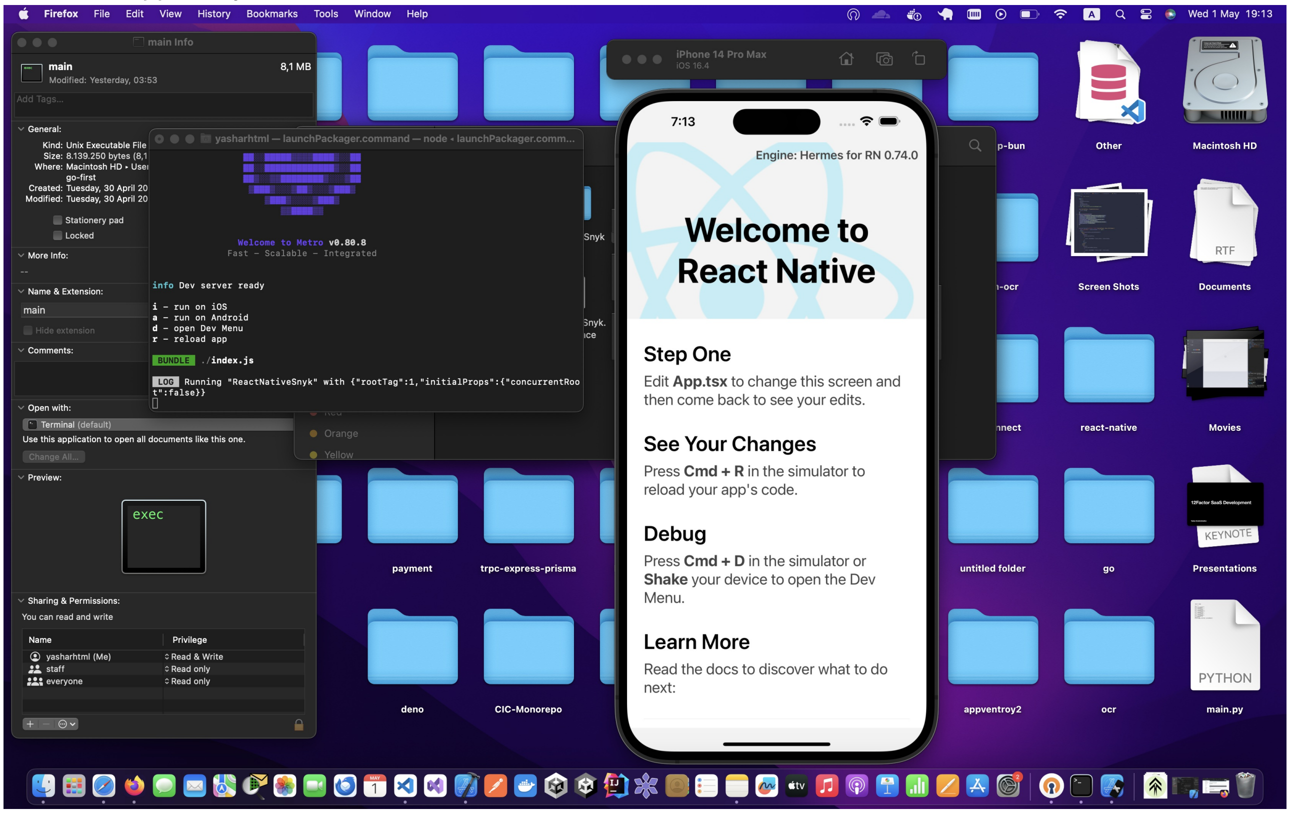
```
REACTNATIVESNYK
  > __tests__
  > .bundle
  > .yarn
  > android
  > ios                              ●
  > node_modules
    .dccache
    .eslintrc.js
    .gitignore                       M
    .prettierrc.js
    .watchmanconfig
    .yarnrc
    app.json
    App.tsx                          M
    babel.config.js
    Gemfile
    index.js
    jest.config.js
    metro.config.js
    package.json
    README.md
    tsconfig.json
    yarn.lock
```



Now starting the application

```
npm run ios
```

I have the app ready:



main Info

main 8,1 MB
Modified: Yesterday, 03:53

General:
Kind: Unix Executable File
Size: 8.139.250 bytes (8,1
Where: Macintosh HD ▸ User
go-first
Created: Tuesday, 30 April 20
Modified: Tuesday, 30 April 20

Stationery pad
Locked

More Info:
--

Name & Extension:
main

Open with:
Terminal (default)
Use this application to open all documents like this one.

Sharing & Permissions:
You can read and write

Name	Privilege
yasharhtml (Me)	Read & Write
staff	Read only
everyone	Read only

yasharhtml — launchPackager.command — node ◂ launchPackager.comm...

```
                 Welcome to Metro v0.80.8
              Fast - Scalable - Integrated


info Dev server ready

i - run on iOS
a - run on Android
d - open Dev Menu
r - reload app

 BUNDLE  ./index.js

 LOG  Running "ReactNativeSnyk" with {"rootTag":1,"initialProps":{"concurrentRoo
t":false}}
```

iPhone 14 Pro Max
iOS 16.4

7:13

Engine: Hermes for RN 0.74.0

Welcome to React Native

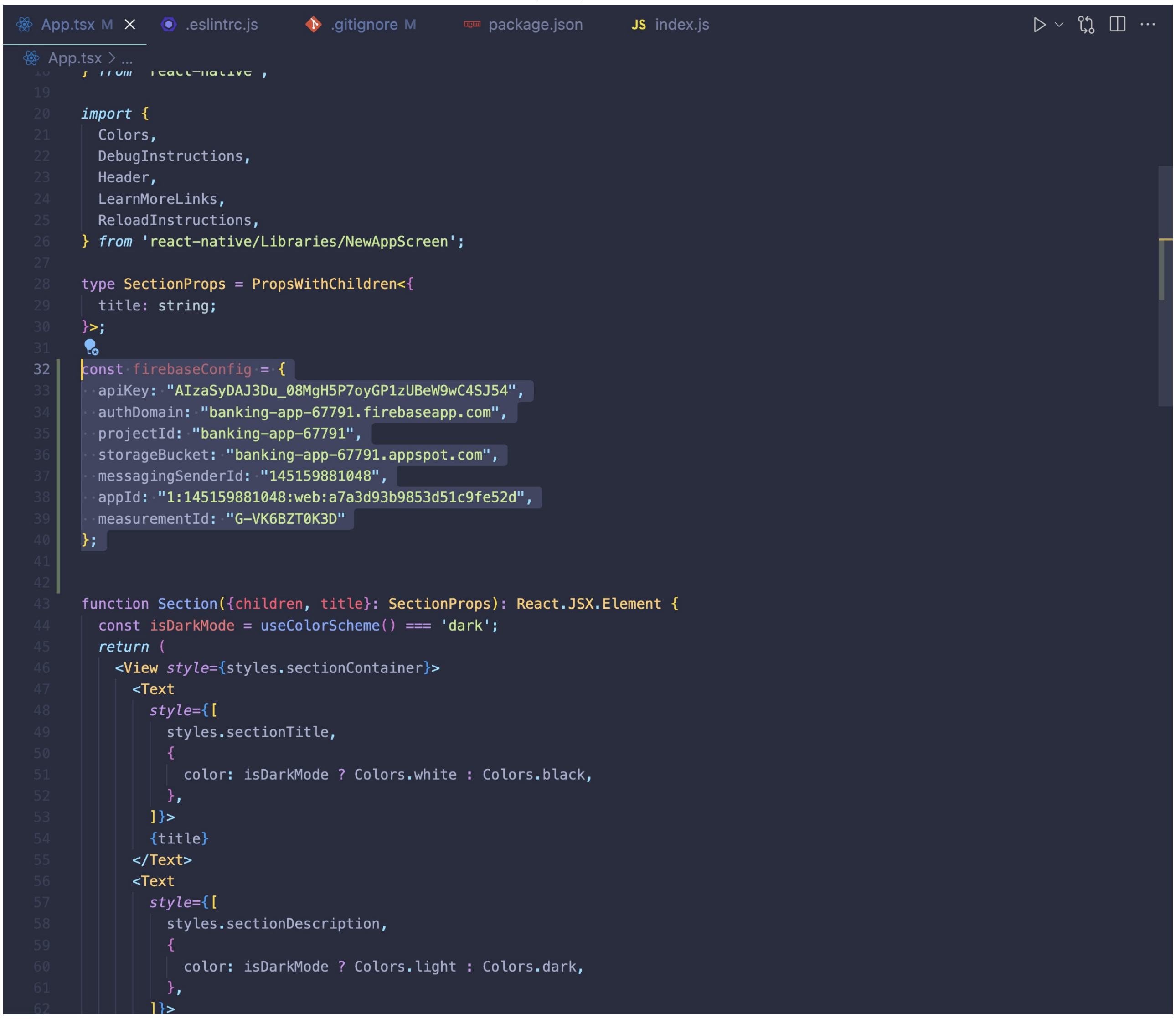
Step One
Edit **App.tsx** to change this screen and then come back to see your edits.

See Your Changes
Press **Cmd + R** in the simulator to reload your app's code.

Debug
Press **Cmd + D** in the simulator or **Shake** your device to open the Dev Menu.

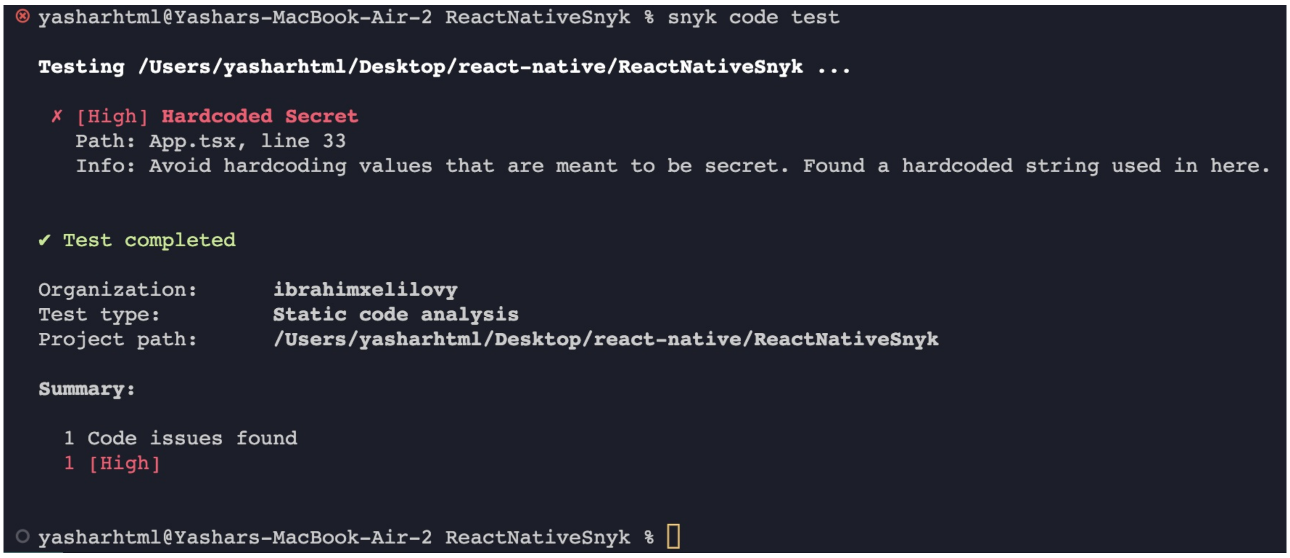
Learn More
Read the docs to discover what to do next:

Here hardcoded secrets are written and located by snyk:



```
19
20  import {
21    Colors,
22    DebugInstructions,
23    Header,
24    LearnMoreLinks,
25    ReloadInstructions,
26  } from 'react-native/Libraries/NewAppScreen';
27
28  type SectionProps = PropsWithChildren<{
29    title: string;
30  }>;
31
32  const firebaseConfig = {
33    apiKey: "AIzaSyDAJ3Du_08MgH5P7oyGP1zUBeW9wC4SJ54",
34    authDomain: "banking-app-67791.firebaseapp.com",
35    projectId: "banking-app-67791",
36    storageBucket: "banking-app-67791.appspot.com",
37    messagingSenderId: "145159881048",
38    appId: "1:145159881048:web:a7a3d93b9853d51c9fe52d",
39    measurementId: "G-VK6BZT0K3D"
40  };
41
42
43  function Section({children, title}: SectionProps): React.JSX.Element {
44    const isDarkMode = useColorScheme() === 'dark';
45    return (
46      <View style={styles.sectionContainer}>
47        <Text
48          style={[
49            styles.sectionTitle,
50            {
51              color: isDarkMode ? Colors.white : Colors.black,
52            },
53          ]}>
54          {title}
55        </Text>
56        <Text
57          style={[
58            styles.sectionDescription,
59            {
60              color: isDarkMode ? Colors.light : Colors.dark,
61            },
```

snyk code **test**



```
⊗ yasharhtml@Yashars-MacBook-Air-2 ReactNativeSnyk % snyk code test

  Testing /Users/yasharhtml/Desktop/react-native/ReactNativeSnyk ...

   ✗ [High] Hardcoded Secret
     Path: App.tsx, line 33
     Info: Avoid hardcoding values that are meant to be secret. Found a hardcoded string used in here.


  ✔ Test completed

  Organization:      ibrahimxelilovy
  Test type:         Static code analysis
  Project path:      /Users/yasharhtml/Desktop/react-native/ReactNativeSnyk

  Summary:

    1 Code issues found
    1 [High]


○ yasharhtml@Yashars-MacBook-Air-2 ReactNativeSnyk %
```