Tutorial for setup vs-code environment to debug react-native

1. Install VS-Code

2. Install React-Native-Cli globally on your computer:

# npm install react-native-cli -g

3. Install Android Studio & Android SDK

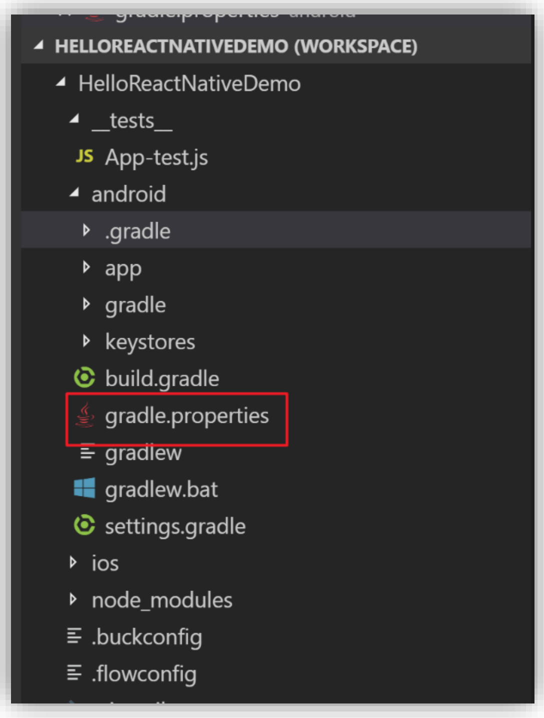
Following the link: https://reactnative.cn/docs/getting-started.html

4. Create project by react-native-cli

# react-native init <project-name>

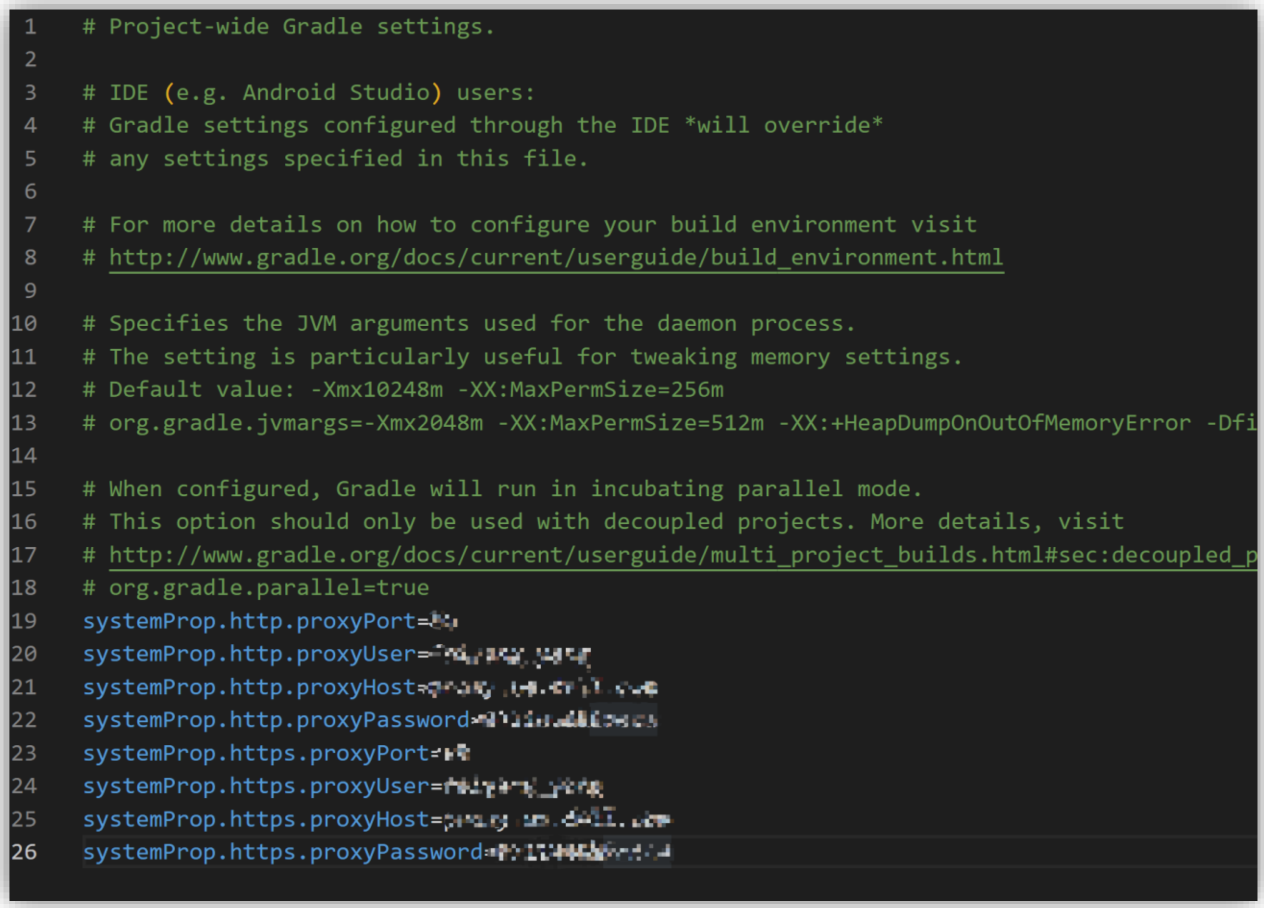
5. Add proxy for gradle

```
Replace the “xxxx” to your VPN information.
   systemProp.http.proxyPort=xxxx
   systemProp.http.proxyUser=xxxx
   systemProp.http.proxyHost=xxxx
   systemProp.http.proxyPassword=xxxx
   systemProp.https.proxyPort=xxxx
   systemProp.https.proxyUser=xxxx
   systemProp.https.proxyHost=xxxx
   systemProp.https.proxyPassword=xxxx
```



```
gradle.properties - android
HELLOREACTNATIVEDEMO (WORKSPACE)
  HelloReactNativeDemo
    __tests__
      App-test.js
    android
      .gradle
      app
      gradle
      keystores
      build.gradle
      gradle.properties
      gradlew
      gradlew.bat
      settings.gradle
    ios
    node_modules
    .buckconfig
    .flowconfig
```

Figure 1 The gradle.properties

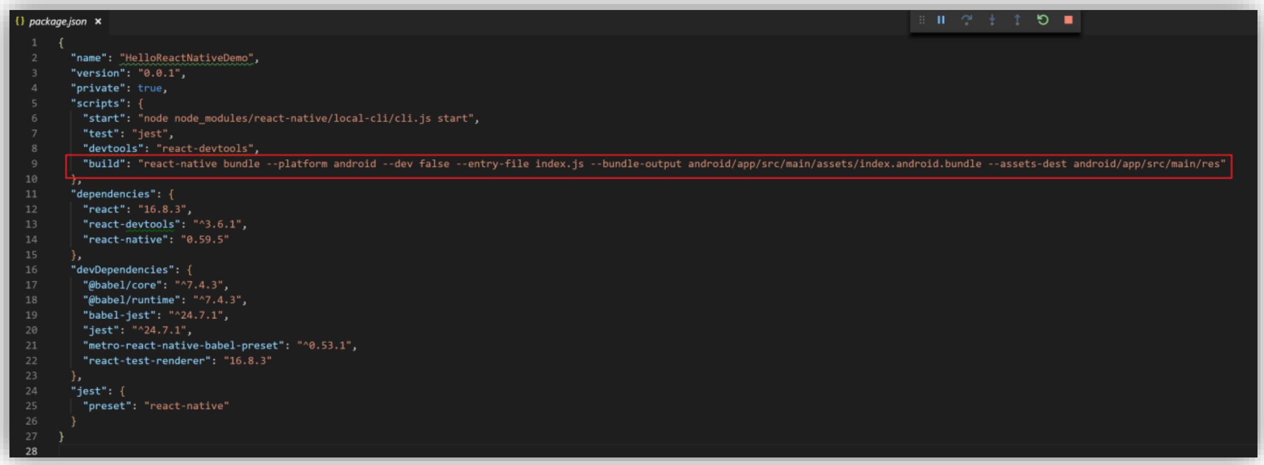


```
# Project-wide Gradle settings.

# IDE (e.g. Android Studio) users:
# Gradle settings configured through the IDE *will override*
# any settings specified in this file.

# For more details on how to configure your build environment visit
# http://www.gradle.org/docs/current/userguide/build_environment.html

# Specifies the JVM arguments used for the daemon process.
# The setting is particularly useful for tweaking memory settings.
# Default value: -Xmx10248m -XX:MaxPermSize=256m
# org.gradle.jvmargs=-Xmx2048m -XX:MaxPermSize=512m -XX:+HeapDumpOnOutOfMemoryError -Dfi

# When configured, Gradle will run in incubating parallel mode.
# This option should only be used with decoupled projects. More details, visit
# http://www.gradle.org/docs/current/userguide/multi_project_builds.html#sec:decoupled_p
# org.gradle.parallel=true
systemProp.http.proxyPort=[redacted]
systemProp.http.proxyUser=[redacted]
systemProp.http.proxyHost=[redacted]
systemProp.http.proxyPassword=[redacted]
systemProp.https.proxyPort=[redacted]
systemProp.https.proxyUser=[redacted]
systemProp.https.proxyHost=[redacted]
systemProp.https.proxyPassword=[redacted]
```

Figure 2 Edit the content of "gradle.properties"

6. If you met the red-screen error "unable to load script from assets index.android.bundle", when you run the project in simulated device.
   Please follow this link to resolve this issue:
   https://stackoverflow.com/questions/54975709/unable-to-load-script-from-assets-index-android-bundle-react-native
   1) Create the "assets" folder under "android/app/src/main"
   2) run this command
      # react-native bundle --platform android --dev false --entry-file index.js --bundle-output android/app/src/main/assets/index.android.bundle --assets-dest android/app/src/main/res
   3) Add "build" command to the "package.json", can run this script every time you rebuild the project.

```
"build": "react-native bundle --platform android --dev false --entry-file
index.js --bundle-output android/app/src/main/assets/index.android.bundle
--assets-dest android/app/src/main/res"
```



```json
{
  "name": "HelloReactNativeDemo",
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "start": "node node_modules/react-native/local-cli/cli.js start",
    "test": "jest",
    "devtools": "react-devtools",
    "build": "react-native bundle --platform android --dev false --entry-file index.js --bundle-output android/app/src/main/assets/index.android.bundle --assets-dest android/app/src/main/res"
  },
  "dependencies": {
    "react": "16.8.3",
    "react-devtools": "^3.6.1",
    "react-native": "0.59.5"
  },
  "devDependencies": {
    "@babel/core": "^7.4.3",
    "@babel/runtime": "^7.4.3",
    "babel-jest": "^24.7.1",
    "jest": "^24.7.1",
    "metro-react-native-babel-preset": "^0.53.1",
    "react-test-renderer": "16.8.3"
  },
  "jest": {
    "preset": "react-native"
  }
}
```

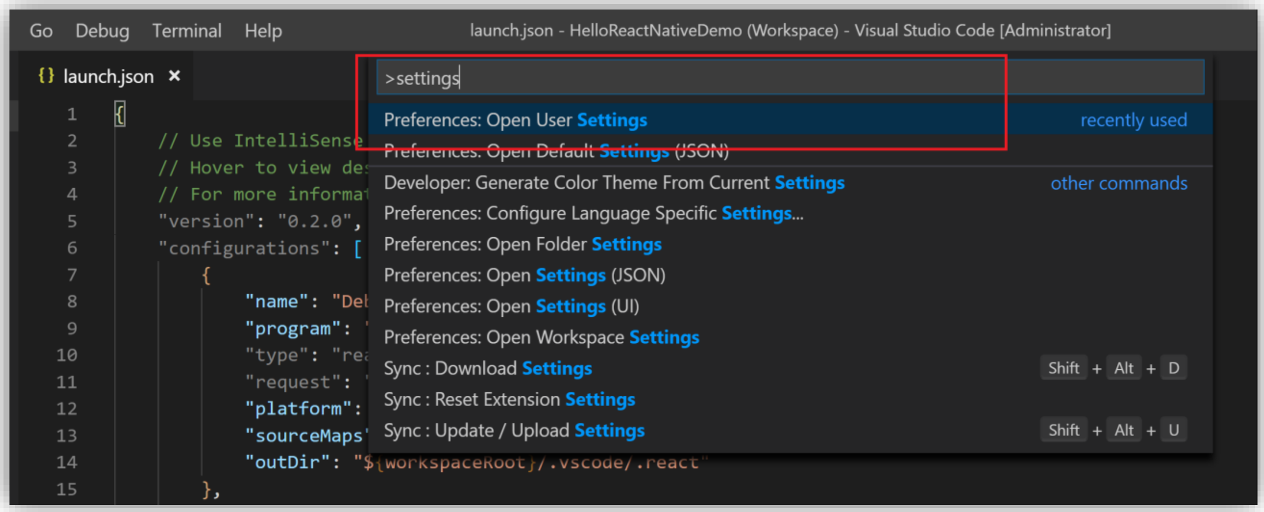
Figure 3 Add build command for build the project

7. Install Ract-Native Tools in VsCode

1) Press F1 in VSCode
2) Type "launch" in the command text-box.
3) Click "Add Configuration" button in the right-bottom.
4) Select "React-Native", then the React-Native Tools extension will add default settings for you.

8. Change the default port for React-Native, if you need. When you install McFee, the original default port which is 8081 will be occupied by Mcfee.

1) Press F1 to open command mode.
2) Type "settings" in the command text-box.
3) Select "Preferences: Open User Settings"



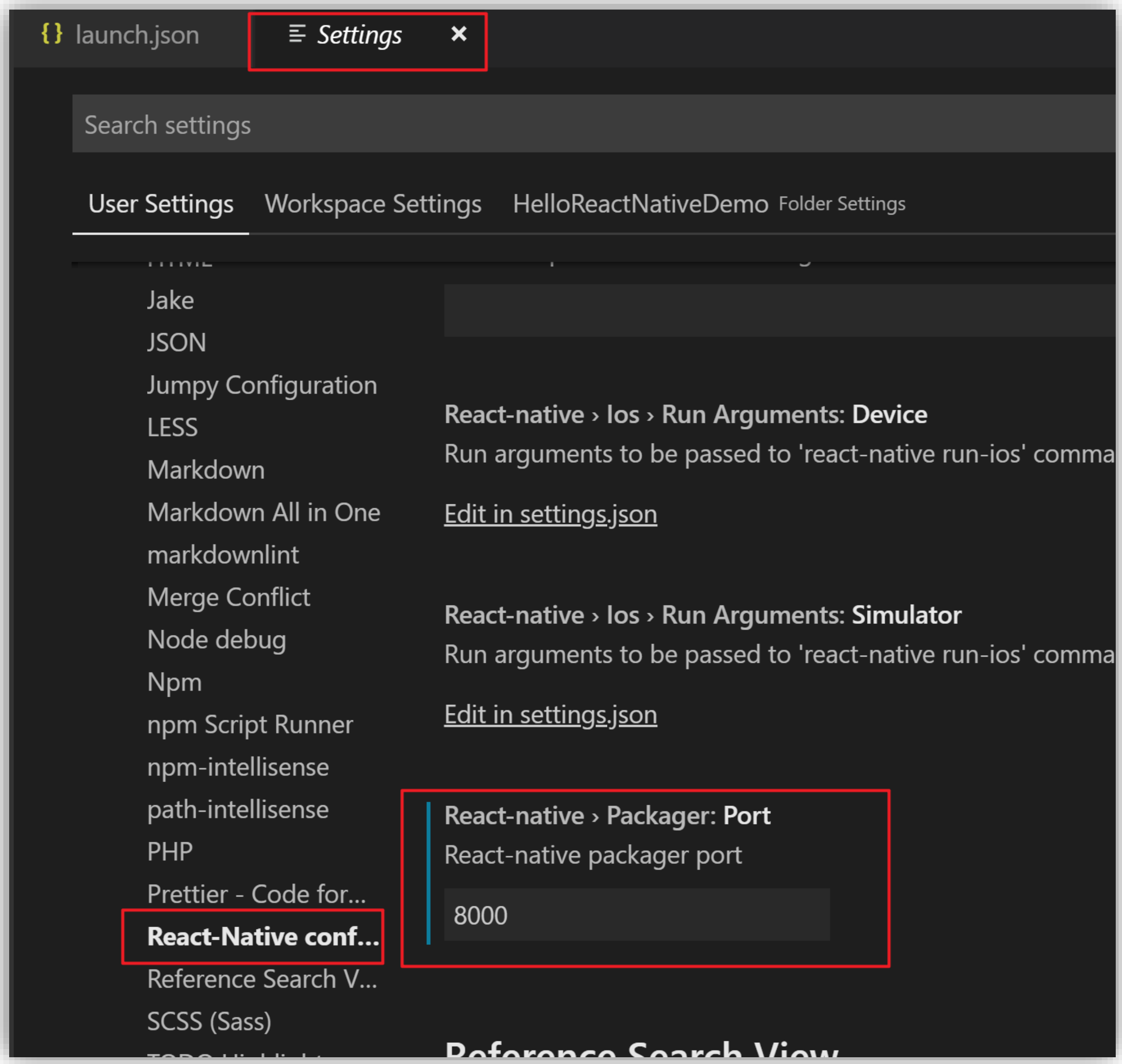
Go
Debug
Terminal
Help
launch.json - HelloReactNativeDemo (Workspace) - Visual Studio Code [Administrator]
launch.json
>settings
Preferences: Open User Settings
recently used
Preferences: Open Default Settings (JSON)
Developer: Generate Color Theme From Current Settings
other commands
Preferences: Configure Language Specific Settings...
Preferences: Open Folder Settings
Preferences: Open Settings (JSON)
Preferences: Open Settings (UI)
Preferences: Open Workspace Settings
Sync : Download Settings
Shift + Alt + D
Sync : Reset Extension Settings
Sync : Update / Upload Settings
Shift + Alt + U

```
{
    // Use IntelliSense
    // Hover to view des
    // For more informat
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Del
            "program":
            "type": "rea
            "request":
            "platform":
            "sourceMaps'
            "outDir": "${workspaceRoot}/.vscode/.react"
        },
```

Figure 4 Open Settings View

4) Select Extensions in the settings view, then select "React-Native".

5) Then find the Packager-Port in the content-view, and modify it to your port.



React-native › Ios › Run Arguments: **Device**

Run arguments to be passed to 'react-native run-ios' comma

Edit in settings.json

React-native › Ios › Run Arguments: **Simulator**

Run arguments to be passed to 'react-native run-ios' comma

Edit in settings.json

React-native › Packager: **Port**

React-native packager port

8000

Reference Search View

Figure 5 Set default port for React-Native

9. Debug for Android, we don't have ios, so use android as an example.

1) Open Android Studio, and run simulate device. Please follow the first link to create the simulate device.

2) Open your project in VSCode, and press "F5" to run your project. After a while, the project will be started in the simulate device.

3) Press "ctrl+M" in your device

- Select "Dev Settings"
- Then select "Debug server host & port for device" in the prompt out dialog.
- Input 127.0.0.1:<customized Port>. In my case, I set it as 8000 in 8-5), so I put it as "127.0.0.1:8000"

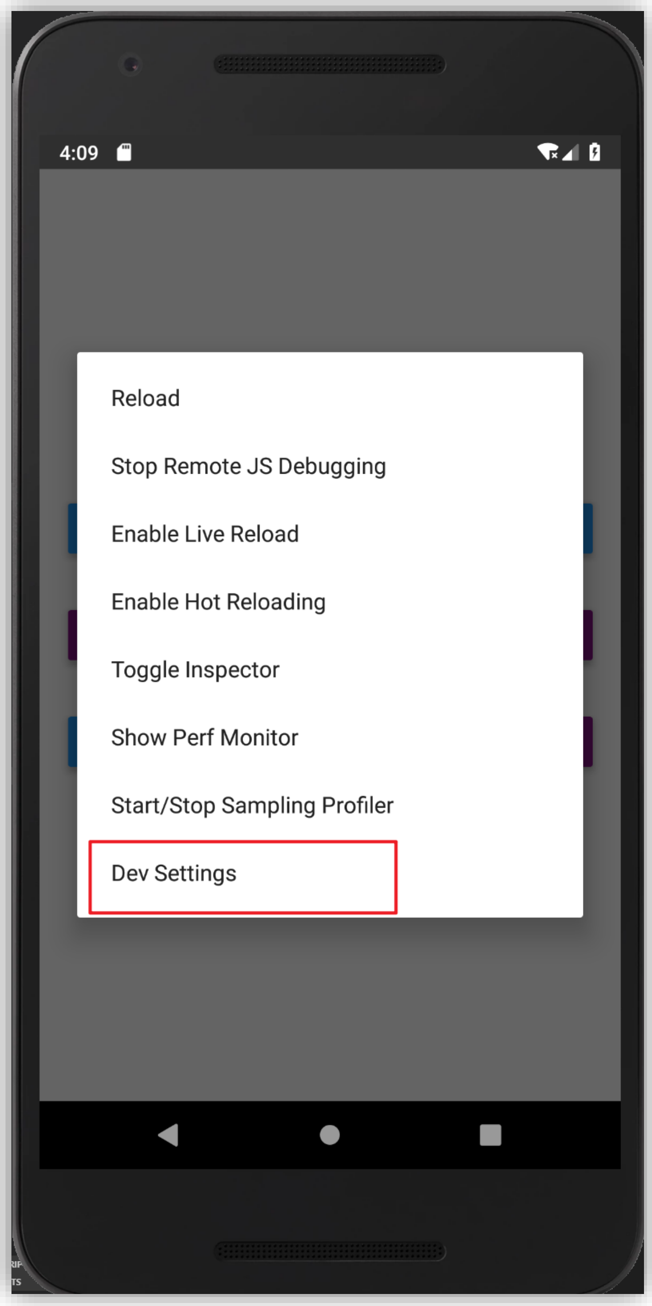
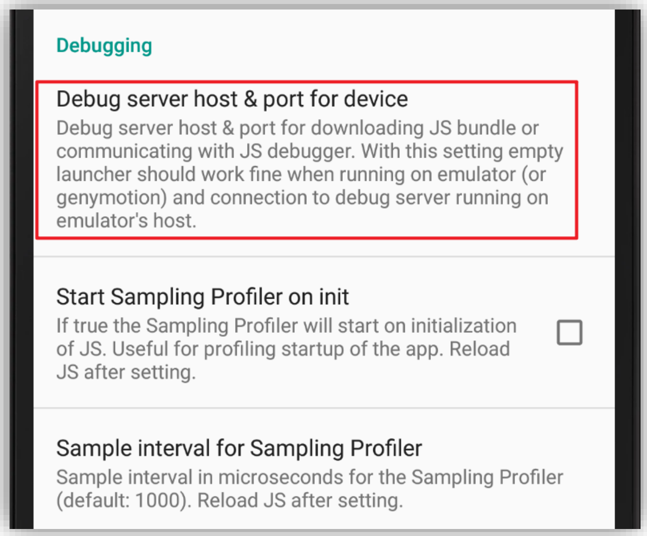


Figure 6 Get Dev-Settings



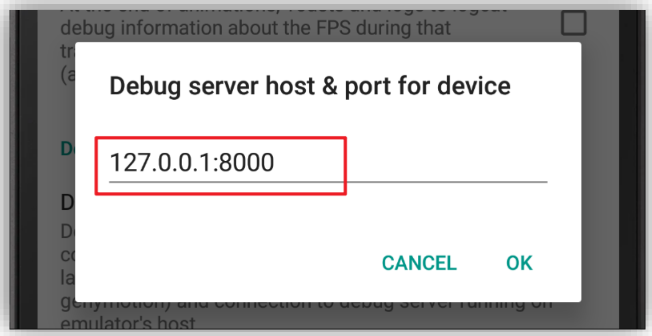
Debugging

Debug server host & port for device
Debug server host & port for downloading JS bundle or communicating with JS debugger. With this setting empty launcher should work fine when running on emulator (or genymotion) and connection to debug server running on emulator's host.

Start Sampling Profiler on init
If true the Sampling Profiler will start on initialization of JS. Useful for profiling startup of the app. Reload JS after setting. □

Sample interval for Sampling Profiler
Sample interval in microseconds for the Sampling Profiler (default: 1000). Reload JS after setting.

Figure 7 Select Settings Option



debug information about the FPS during that □

Debug server host & port for device

127.0.0.1:8000

Figure 8 Input your information

10. Then when you click on the device, you can get the break-point in VSCode.