Getting Started with React Native and Expo



Table of Contents

[1. Introduction 3](#_Toc178779163)

[2. Overview 3](#_Toc178779164)

[3. Installing the Development Environment 3](#_Toc178779165)

[3.1 Computer Installation 3](#_Toc178779166)

[3.1.1 Install Node.js 3](#_Toc178779167)

[3.2 Phone Installation (iOS or Android) 4](#_Toc178779168)

[4. Creating a New React Native App with Expo 4](#_Toc178779169)

[4.1 Initializing the Project 4](#_Toc178779170)

[4.2 Viewing Your App on Your Phone 4](#_Toc178779171)

[4.3 Understanding the Project Structure 6](#_Toc178779172)

[4.3.1. Essential Project Structure/ 6](#_Toc178779173)

[4.3.2. Explanation 6](#_Toc178779174)

[5. Network Configuration 7](#_Toc178779175)

[5.2.1. Set up IP Address 7](#_Toc178779176)

[5.2.2. Ports 8](#_Toc178779177)

[6. Creating App.js File 8](#_Toc178779178)

[7. Modifying Your App 9](#_Toc178779179)

[7.1 Open and Edit Your index.js File 9](#_Toc178779180)

[8. Troubleshooting 10](#_Toc178779181)

[9. Additional Resources 10](#_Toc178779182)

[10. Key Terms 10](#_Toc178779183)

[11. Glossary 10](#_Toc178779184)

# 1. Introduction

React Native is an open-source framework for building mobile applications using JavaScript, allowing developers to create apps for both Android and iOS with a single codebase. This approach greatly reduces development time and effort, making it an efficient choice for cross-platform app development.

Expo, on the other hand, is a platform and set of tools that simplify React Native development, especially for beginners. It offers a streamlined development experience by providing pre-configured workflows, a built-in library of components, and easy access to native device features. With Expo, you can avoid the complexities of native code, making it easier to get started and focus on building your app.

This guide is designed for people who are new to React Native and want a hands-on introduction. It will walk you through installing the necessary development environment, setting up Expo, and creating your first React Native app. By the end of this guide, you'll have a foundational understanding of how React Native works and how to use Expo to develop and run your applications efficiently.

# 2. Overview

What is React Native? React Native Combines React, a JavaScript library for building user interfaces, with the capabilities of Android or iOS platforms. It allows you to create mobile apps using JavaScript without needing to learn native languages like Kotlin, Java, Swift, or Objective-C.

Who is this guide for? It is intended for those with very basic knowledge of JavaScript and React, although prior experience is helpful but not required.

# 3. Installing the Development Environment

## 3.1 Computer Installation

You need to install two essential programs on your computer: Node.js and Expo CLI. You can use either PowerShell or the Windows Command Prompt for this process. It's recommended to run the installation as an administrator to ensure you have the necessary permissions to modify system files and correctly add global packages to the PATH environment variable.

### 3.1.1 Install Node.js

1. Visit the [Node.js download page](<https://nodejs.org/en/download/> ).

2. Download the latest LTS version suitable for your operating system (Windows, macOS, or Linux).

3. Open the downloaded file and follow the setup wizard:

- Click “Next” to proceed.

- Accept the license agreement and click “Next.”

- Choose your preferred installation location, then click “Next.”

- Ensure the "Add to PATH" option is selected (important).

- Click “Install” to begin, and then “Finish” when the installation completes.

4. Verify the installation by running the following commands in your terminal/command prompt:

**node –v** and **npm -v**

## 3.2 Phone Installation (iOS or Android)

Download the **Expo Go** app from the App Store (iOS) or Google Play Store (Android).

Ensure your computer and phone are connected to the **same Wi-Fi network** for seamless communication.

# 4. Creating a New React Native App with Expo

## 4.1 Initializing the Project

1. Open your terminal/command prompt.

2. Navigate to your preferred folder:

**cd path/to/your/folder**

Create a new project by running:

**npx create-expo-app HelloWorld**

- This command will fetch and run the latest “create-expo-app” tool without needing a global installation.

4. Navigate into your project folder:

**cd HelloWorld**

5. Start your project:

**npx expo start**

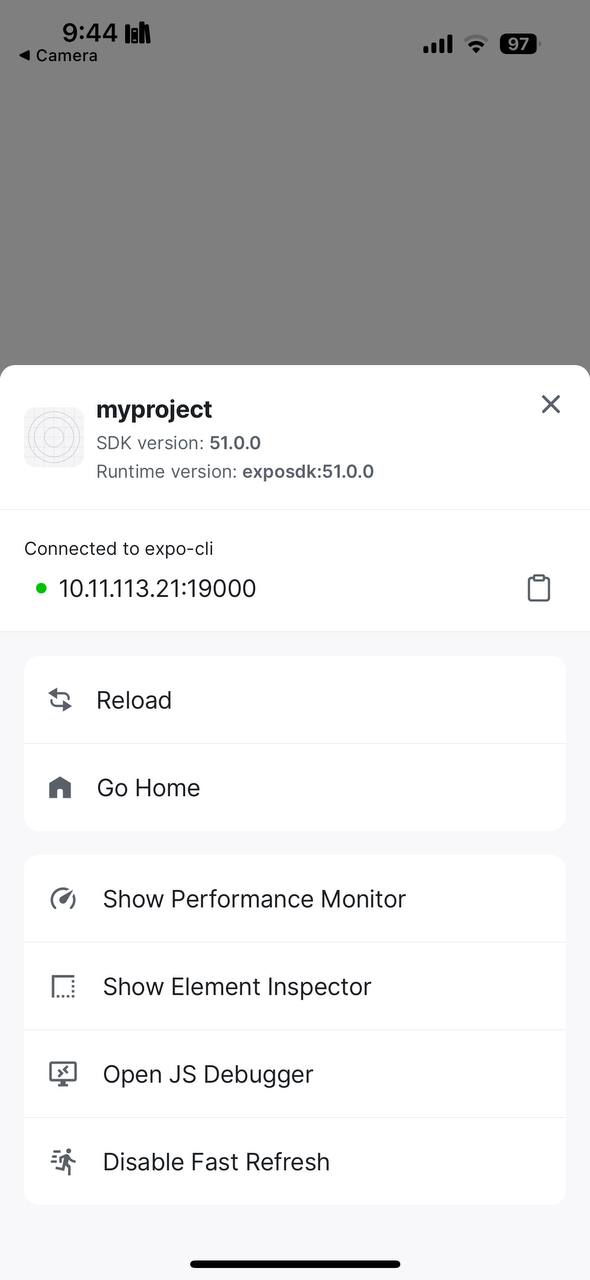
## 4.2 Viewing Your App on Your Phone

1. Open the Expo Go app on your phone.

2. Scan the QR code displayed in your terminal or web browser.

3. You should see your app running on your phone.

You can access the Dev Menu to test and debug your project efficiently. To open the Dev Menu, simply shake your device if you're running the app on a physical device, or press “d” if you're using an emulator/simulator. The Dev Menu provides options such as reloading the app, opening debugger, and inspecting elements, making it easier to troubleshoot and enhance your project during development.



# 4.3 Understanding the Project Structure

Most of the folders and files in a React Native project are unnecessary for basic functionality. You can simplify your project by keeping only the essential files and folders. Specifically, you need to keep the following structure, and add **“App.js” to the root of your project** to set up the main entry point for routing and navigation. You should also **“metro.config.js”** for managing module resolution and bundling settings, which are important for ensuring compatibility and customization during development.

### 4.3.1. Essential Project Structure/

|-- App.js // Entry point for the routing system

|-- app/ // Contains pages or components used by the app

|-- index.js // This serves as the main page of your app

|-- app.json // Configuration file for Expo

|-- node\_modules/ // Contains the installed dependencies

|-- metro.config.js // Metro bundler configuration for handling polyfills

|-- package.json // Project metadata and dependencies

### 4.3.2. Explanation

**- App.js:** This file is the entry point of your project, responsible for setting up routing using “expo-router, it doesn't contain UI elements directly but manages the app's navigation and routes.

**- app/index.js:** This is where the home page content is defined. It’s the main page of your app, and you can define other pages inside the app/ folder, when your app loads, it will render the content from “index.js” (or other pages as defined by the routes).

**- metro.config.js:** Custom configurations for the Metro bundler, and other Node.js modules required by the app.

**- app.json:** This is the configuration file for the Expo app, containing details like the app name, version, and additional plugins or settings.

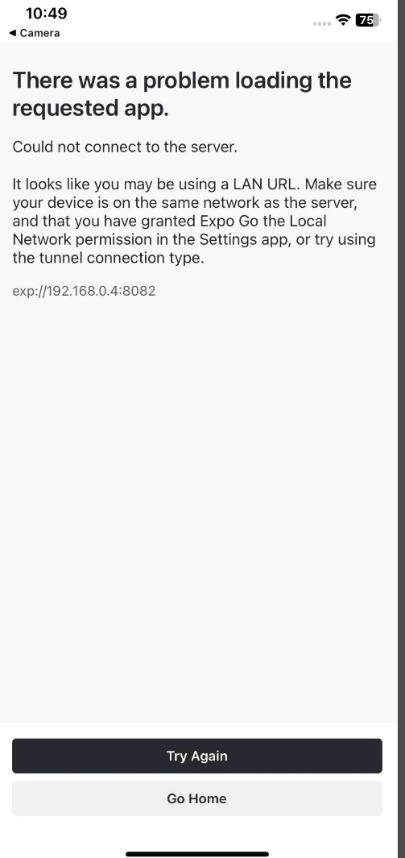
Entry Point: App.js is the entry point for the entire project, managing the routing system with expo-router. index.js is the main page (home page) of your app and serves the content rendered initially when the app is loaded.

**Note:** If the Expo Go app fails to connect to your project, especially on a public network, you may need to adjust your network settings. For detailed instructions, refer to the "Network Configuration" section otherwise you can skip section 5 and continue with section 6.

# 5. Network Configuration

1. Make sure your device is on the same network as the server, and
2. that you have granted Expo Go the Local Network permission in the Settings app, or
3. try using the tunnel connection type.

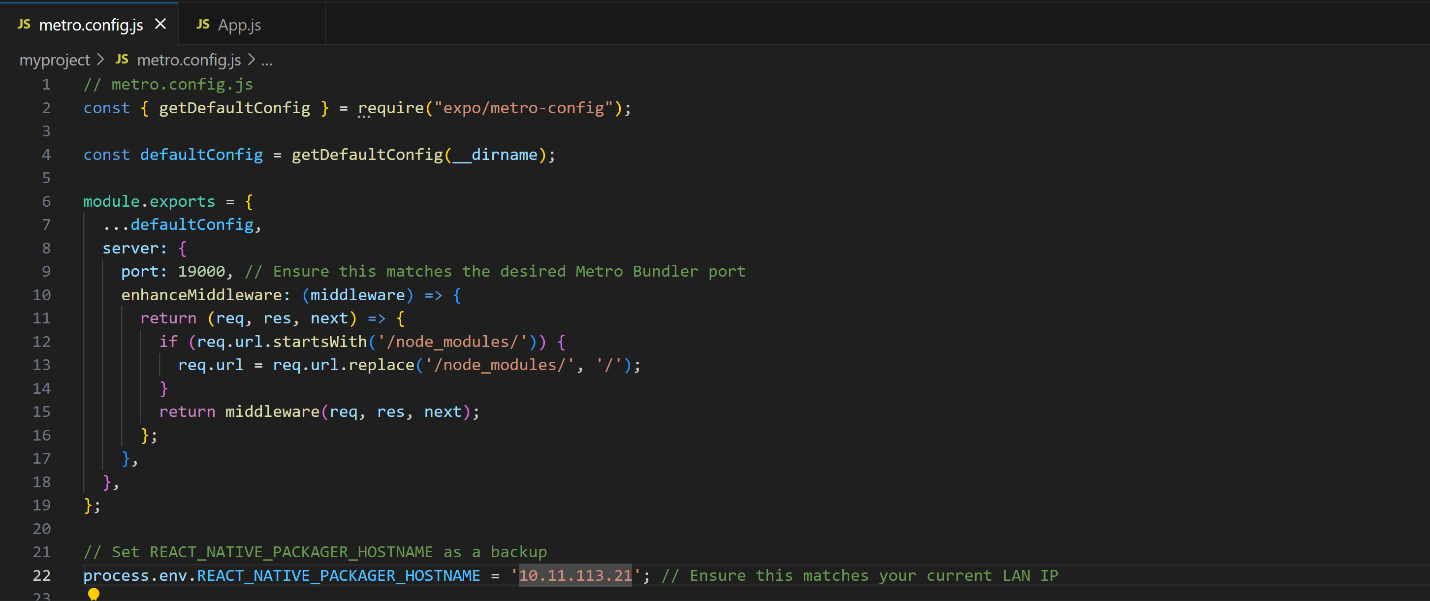
“**npx expo start --tunnel**”



### 5.2.1. Set up IP Address

If you want to set a fixed IP address for your React Native project, follow these steps:

1. Create or edit the “metro.config.js” file in your project root and add the following configuration:



2. Set the “REACT\_NATIVE\_PACKAGER\_HOSTNAME” environment variable by running the following command in your terminal:

“**export REACT\_NATIVE\_PACKAGER\_HOSTNAME=<your-computer-ip>**”

Replace “<your-computer-ip>” with your actual IP address, for example, “192.168.1.10”.

### 5.2.2. Ports

The Metro Bundler starts working by default with “localhost” on port 8081. If it doesn't work, you can switch to using your computer's IP address with port 19000, which is commonly used by the Expo development server for better connectivity with your device.

| **Port** | **Used By** | **Purpose** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| 8081 | React Native Metro Bundler | Serves the JavaScript bundle and handles live reloading for React Native projects. |

|  |  |  |
| --- | --- | --- |
| 19000 | Expo Dev Server | Main server for serving JavaScript code to the Expo Go app. |

|  |  |  |
| --- | --- | --- |
| 19001 | Expo WebSocket Server | Enables real-time communication, debugging, and logs for Expo projects. |

|  |  |  |
| --- | --- | --- |
| 19002 | Expo Developer Tools UI | Hosts the web-based Expo Developer Tools interface. |

# 6. Creating App.js File

When you create a new React Native project using the “npx” command (e.g., “**npx create-expo-app**”), the project setup does not automatically include an “App.js” file. In this case, you will need to create the “App.js” file manually in the **main root of your project** to serve as the entry point.

Once the “App.js” file is created, you also need to update your “app.json” configuration to ensure Expo recognizes it as the entry point for your app. This can be done by adding or modifying the "entryPoint" field in “app.json”.

Steps:

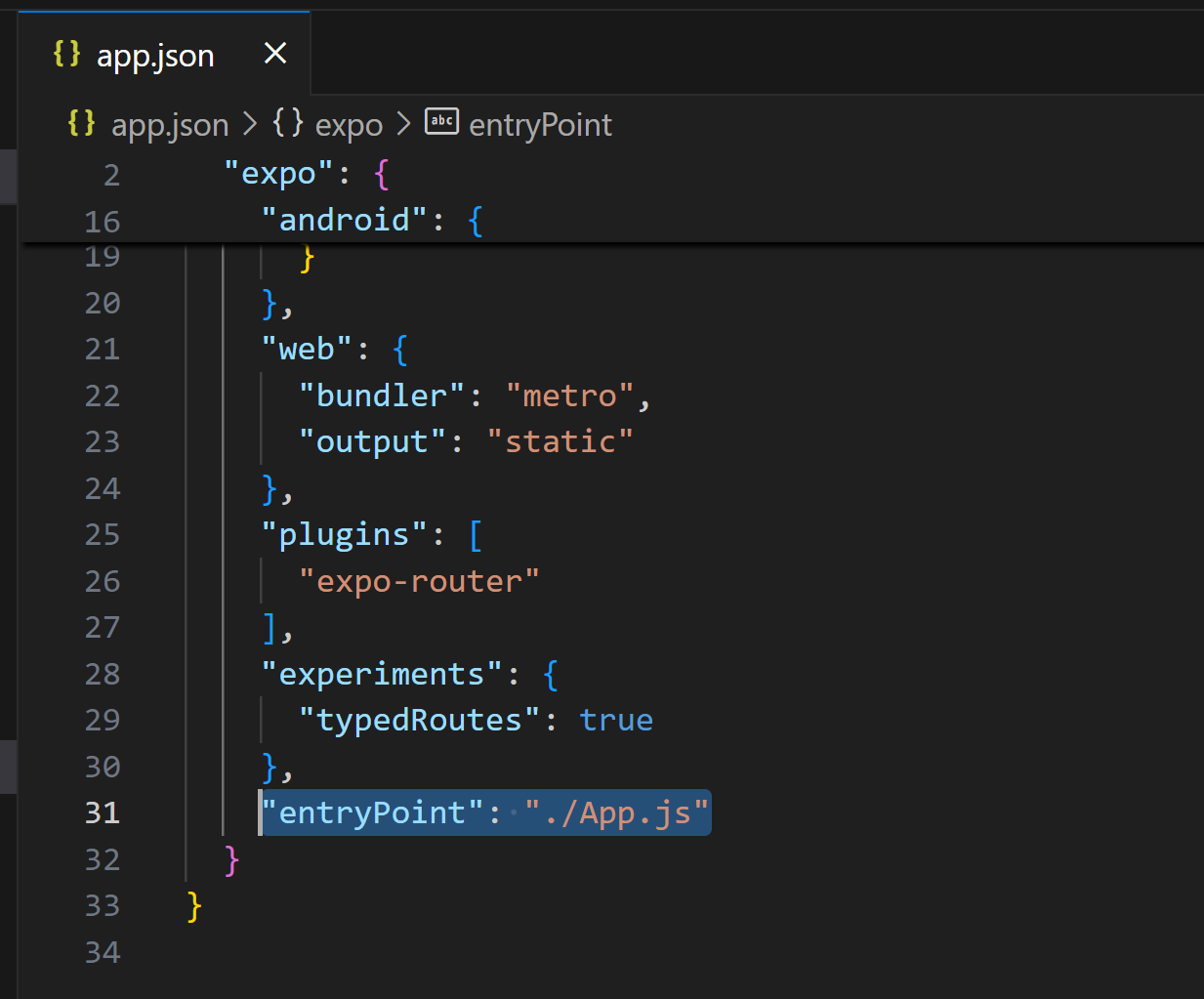
1. Create the “App.js” File:

- Manually create a new file named “App.js” in the root of your project.

- Define the root component and any routing logic you need in this file.

2. Modify “app.json” File:

- Open the “app.json” file and ensure it includes the following "entryPoint" field:



- This tells Expo that “App.js” is the main entry point of your app.

Once you've created the App.js file and set it as the entry point, **App.js acts as the router**, managing navigation and routing logic. However, your **main page content** should be written in **index.js** located inside the app/ folder.

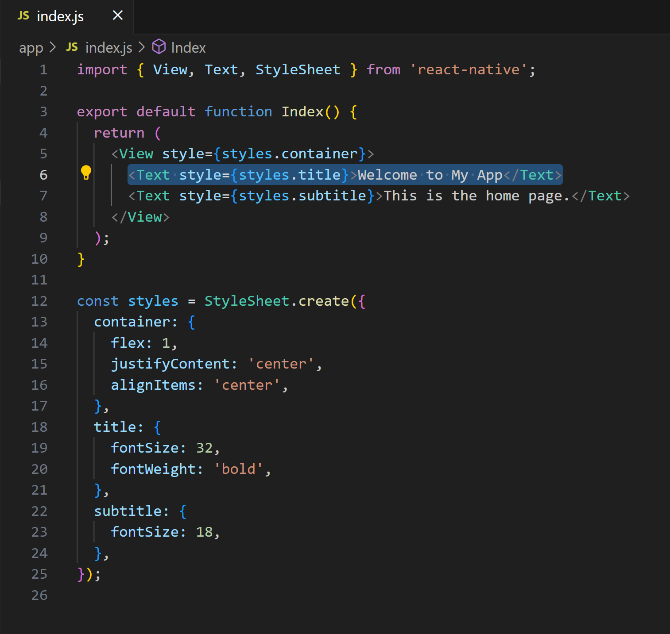
In this structure:

* **App.js** is responsible for routing, setting up the initial navigation, and rendering the app's root.
* **index.js** in the app/ folder contains the main content that will be displayed as the home page.

# 7. Modifying Your App

## 7.1 Open and Edit Your index.js File

1. Open “index.js” in a text editor (e.g., Visual Studio Code, Notepad).

2. Find the line with “<Text>” containing "Welcome to My App".

3. Change this text to " Buongiorno!" and save the file.

You should see the changes reflected in your app automatically.

# 8. Troubleshooting

If you encounter issues during setup, refer to the React Native [Setup Troubleshooting Guide] (https://reactnative.dev/docs/troubleshooting).

Common issues include:

Expo CLI not recognized: Ensure you installed it with “npm install -g expo-cli” and restart your terminal.

# 9. Additional Resources

[React Native Documentation] (<https://reactnative.dev/docs/getting-started> )

[JavaScript Tutorials by Mozilla] (<https://developer.mozilla.org/en-US/docs/Web/JavaScript> )

[React JavaScript Library] (<https://reactjs.org/> )

[Expo Documentation] (<https://docs.expo.dev/>)

# 10. Key Terms

**Command Prompt:** A text-based interface used to run commands on Windows.

**Terminal:** A similar tool to the command prompt, used in macOS/Linux.

**LTS:** Long-Term Support, which refers to software versions that receive updates and support for a longer period.

# 11. Glossary

**Node.js:** JavaScript runtime environment.

**npx:** A tool that runs Node.js packages without installing them globally, making it convenient for one-time use.

**JSX:** JavaScript XML syntax used in React.

**Props:** Custom properties for React components.

**State:** Data that controls a component's behavior in React.