



Setup, Build And Run Martvill Customer Mobile App

To run an Expo React Native application, here are the step-by-step guideline for you:



Prerequisites

1. Install Node.js and npm (Node Package Manager) – **(Installation guideline)**
2. yarn install – **(Installation guideline)**
3. JDK (Java SE Development Kit)
4. Setup Android Studio – **(Installation guideline)**
5. Install Text Editor / Visual Studio code (recommended) – **(download link)**

Or Go to the official React Native website at **(<https://reactnative.dev/docs/environment-setup?guide=native#jdk>)** and follow the environment-setup instructions.

Ensure that all the prerequisites mentioned above are installed on your machine, and then proceed with the following steps.



Configuration, Build, and Run

STEP-1: To get started, please extract your **martvill.zip** file first.

STEP-2: Open VS Code and navigate to your project folder:

- Launch VS Code.
- Select “File” from the top menu, and then click “Open Folder...”
- Browse to the location where your app folder is and select the app folder. Click “Open.”

STEP-3: Install the project dependencies (**yarn install**):

- Open a new terminal in VS Code by selecting “View” from the top menu, then “Terminal,” and finally “New Terminal.”
- In the terminal, navigate to the project folder root if you’re not already there. You can use the **cd** command to change directories.
- Run **yarn install** to install the project dependencies. This command will read the **package.json** file and download the necessary packages.

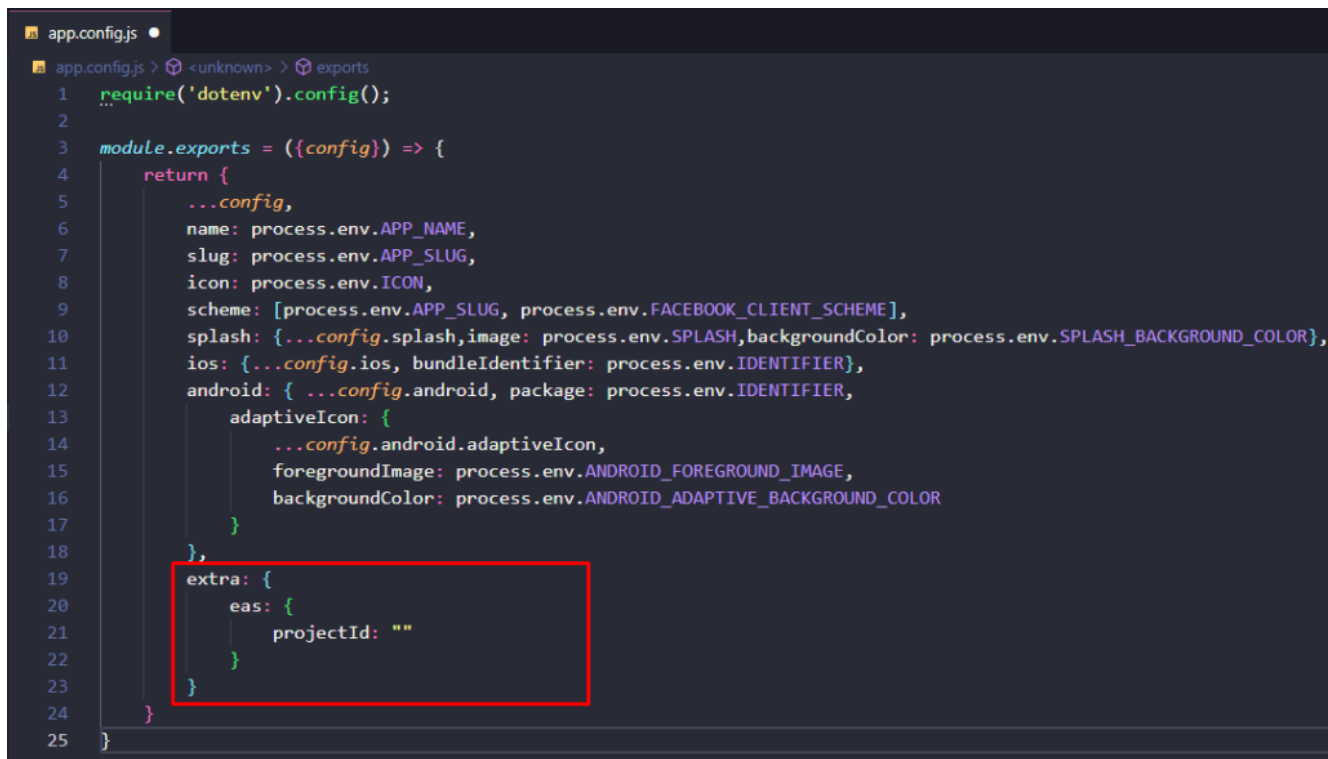
STEP-4: Open Android Studio:

- If you haven’t already, install Android Studio from the official website (<https://developer.android.com/studio>) and set up an Android emulator.
- Open Android Studio and click on the “**AVD Manager**” icon in the toolbar.
- Create a new Android Virtual Device (AVD) if you don’t have one already. Choose a suitable device configuration and Android version, then click “Next” and “Finish” to create the AVD.
- Make a note of the AVD name, as you’ll need it in the next step.

STEP-5: Update your project’s environment variables in the “**.env**” file.:

- Locate the “**.env**” file in your project directory.
- Make the necessary changes to the environment variables, following the specified format (e.g., **VARIABLE_NAME=value**).
- Save the changes to the “**.env**” file.
- Ensure that you don’t accidentally change the file extension or rename the file.
- Double-check the modifications to verify that the changes are accurate and properly formatted.
- Once you’re satisfied with the changes, proceed to use the updated environment variables in your project.

STEP-6: Ensure that the “projectId” field remains empty in your “**app.config.js**” file.:



```

1  require('dotenv').config();
2
3  module.exports = ({config}) => {
4    return {
5      ...config,
6      name: process.env.APP_NAME,
7      slug: process.env.APP_SLUG,
8      icon: process.env.ICON,
9      scheme: [process.env.APP_SLUG, process.env.FACEBOOK_CLIENT_SCHEME],
10     splash: {...config.splash, image: process.env.SPLASH, backgroundColor: process.env.SPLASH_BACKGROUND_COLOR},
11     ios: {...config.ios, bundleIdentifier: process.env.IDENTIFIER},
12     android: { ...config.android, package: process.env.IDENTIFIER,
13       adaptiveIcon: {
14         ...config.android.adaptiveIcon,
15         foregroundImage: process.env.ANDROID_FOREGROUND_IMAGE,
16         backgroundColor: process.env.ANDROID_ADAPTIVE_BACKGROUND_COLOR
17       }
18     },
19     extra: {
20       eas: {
21         projectId: ""
22       }
23     }
24   }
25 }

```

STEP-7: Install the latest EAS CLI:

EAS CLI is the command-line app that you will use to interact with EAS (Expo Application Service) services from your terminal. To install it, run the following command:

```
npm install -g eas-cli
```

STEP-8: Log in to your Expo account:

If you are already signed in to an Expo account using Expo CLI, you can skip the steps described in this section. If you are not, run the following command to log in:

```
eas login
```

STEP-9: Configure the platforms for EAS Build:

Follow the given instructions to configure an iOS or an Android project for EAS Build:

- Run this command ***"eas build:configure"*** in your vs code terminal
- When you run the command ***"eas build:configure"***, you will receive a warning message saying 'EAS project not configured.' Additionally, a query will be prompted, asking whether you *'would like to automatically create an EAS project for '@your_name/martvill?'* At this point, you should press **'Y'** for "Yes" to proceed."
- And once again, you will receive another warning message to complete the setup process, set **"extra.eas.projectId"** in your **app.config.js** or **app.json**:

All the steps of the process will be displayed in the following manner:

```
TERMINAL
user@Explore MINGW64 /d/client-handover-project/martvill
$ eas build:configure
EAS project not configured.
✓ Would you like to automatically create an EAS project for @apelmahmud/martvill? ... yes
✓ Created @apelmahmud/martvill: https://expo.dev/accounts/apelmahmud/projects/martvill on Expo

Warning: Your project uses dynamic app configuration, and the EAS project ID can't automatically be added to it.
https://docs.expo.dev/workflow/configuration/#dynamic-configuration-with-appconfigjs

To complete the setup process, set "extra.eas.projectId" in your app.config.js or app.json:

{
  "expo": {
    "extra": {
      "eas": {
        "projectId": "bbc7bc0c-4a05-40f9-9d23-f5ec34aeb006"
      }
    }
  }
}

✗ Linking local project to EAS project bbc7bc0c-4a05-40f9-9d23-f5ec34aeb006
Cannot automatically write to dynamic config at: app.config.js
Error: build:configure command failed.
```

- Open the **"app.config.js"** file and ensure to retain the **projectId** while saving the changes.

```
app.config.js
app.config.js > <unknown> > exports > extra > eas > projectId
1  ...require('dotenv').config();
2
3  module.exports = ({config}) => {
4    return {
5      ...config,
6      name: process.env.APP_NAME,
7      slug: process.env.APP_SLUG,
8      icon: process.env.ICON,
9      scheme: [process.env.APP_SLUG, process.env.FACEBOOK_CLIENT_SCHEME],
10     splash: { ...config.splash, image: process.env.SPLASH, backgroundColor: process.env.SPLASH_BACKGROUND_COLOR },
11     ios: { ...config.ios, bundleIdentifier: process.env.IDENTIFIER },
12     android: { ...config.android, package: process.env.IDENTIFIER,
13       adaptiveIcon: {
14         ...config.android.adaptiveIcon,
15         foregroundImage: process.env.ANDROID_FOREGROUND_IMAGE,
16         backgroundColor: process.env.ANDROID_ADAPTIVE_BACKGROUND_COLOR
17       }
18     },
19     extra: {
20       eas: {
21         projectId: "bbc7bc0c-4a05-40f9-9d23-f5ec34aeb006"
22       }
23     }
24   }
25 }
```

- After setting the **projectId**, simply run again the command **"eas build:configure"**, and if you found any steps, simply adhere to the provided instructions.
- EAS Build requires you to use a git repository for your project. At this point, you should press **'Y (yes)'** for **"git init"** and simply adhere to the next provided instruction for **"Commit message"**.

```
TERMINAL
bash + v [] [] ... x

user@Explore MINGW64 /d/client-handover-project/martvill
$ eas build:configure
💡 The following process will configure your iOS and/or Android project to be compatible with EAS Build. These changes only apply to your local project files and you can safely revert them at any time. It looks like you haven't initialized the git repository yet. EAS Build requires you to use a git repository for your project.
✓ Would you like us to run 'git init' in D:\client-handover-project\martvill for you? ... yes
We're going to make an initial commit for your repository.
? Commit message: » Initial commit
```

STEP-10 (Optional): Or you can manually Initialize your git repository and make your initial commit to securely store your project.:

- *git init*

- `git add .` (optional)
- `git commit -m "commit message"` (optional)
- Also, you will have to meet with a question to select platforms that would you like to configure for EAS Build like below (Select by using the **up** and **down** arrow):

```
? Which platforms would you like to configure for EAS Build? » - Use arrow-keys. Return to submit.
> All
  iOS
  Android
```

- You will receive a message stating "🎉 **Your project is ready to build**" upon successful completion.

All the above steps of the process will be displayed in the following:

```
user@Explore MINGW64 /d/client-handover-project/martvill
$ eas build:configure
🔔 The following process will configure your iOS and/or Android project to be compatible with EAS Build. These changes only apply to your local project files and you can safely revert them at any time.
It looks like you haven't initialized the git repository yet.
EAS Build requires you to use a git repository for your project.
✓ Would you like us to run 'git init' in D:\client-handover-project\martvill for you? ... yes
We're going to make an initial commit for your repository.
✓ Commit message: ... Initial commit
✓ Which platforms would you like to configure for EAS Build? » All
🎉 Your project is ready to build.

- Run eas build when you are ready to create your first build.
- Once the build is completed, run eas submit to upload the app to app stores.
- Learn more about other capabilities of EAS Build: https://docs.expo.dev/build/introduction
```

for more information, you will visit this site – (<https://docs.expo.dev/deploy/build-project/>).

STEP-11: Build **preview** apk for android(emulator/device) run the following command:

```
eas build -p android --profile preview
```

STEP-12: Build **development** apk for android(emulator/device) run the following command:

```
eas build --profile development --platform android
```

STEP-13: Create a **production** build, and run the following command:

```
eas build --platform android
```

for more information, you will visit this site – (<https://docs.expo.dev/deploy/build-project/>).

STEP-14: Start the development server:

- In the VS Code terminal, run **`yarn start --dev-client`** to start the Expo development server. This command will build your app and provide a QR code for device testing.
- If prompted, choose the option to run the app in the Android emulator.
- If the emulator doesn't launch automatically, you can manually start it by selecting your AVD from the Android Virtual Device Manager and clicking the green "Play" button.

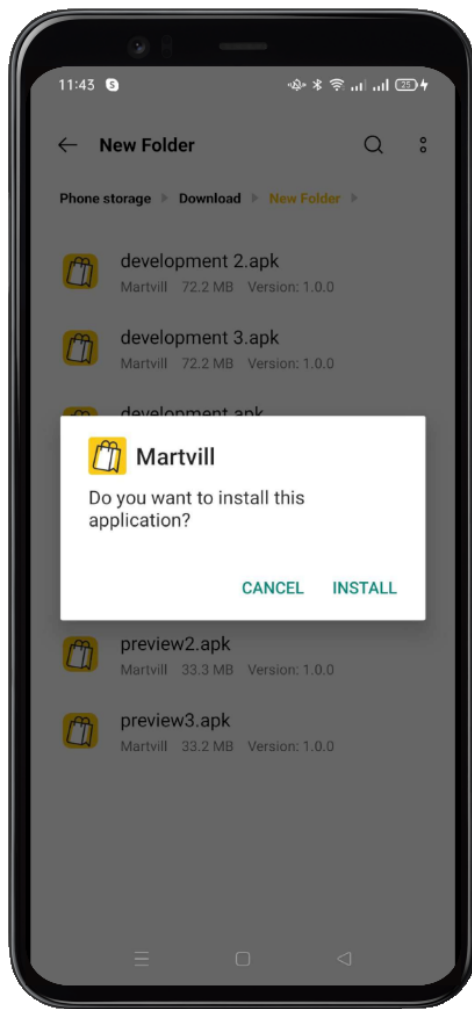
STEP-15:. Test the app on the Android emulator:

- Once the Expo development server is running and the Android emulator is launched, the app should be built and installed automatically.
- The emulator should display your app shortly after the bundling process completes.
- You can interact with the app on the emulator just like on a physical Android device.

STEP-16 (Additional): How to install your build app on your device.

- Download the APK file, open the Downloads folder or your preferred file manager app.
- Tap on the APK file to begin the installation process.

Follow the on-screen prompts to proceed with the installation.



- Once the installation is complete, you can find the app icon on your device's home screen or app drawer.

That's it! You should now be able to run the Expo React Native app on the Android emulator or Physical device using VS Code. Remember, you can also test your app on a physical device by scanning the QR code generated by the Expo development server.

Submit the app store

Coming soon...

Troubleshooting

If you have faced any issues during the app build and running time please feel free to contact our Techvillage team.