Setting Up a React Native Development Environment on macOS

This document outlines the steps to set up a React Native development environment on a macOS system. It covers system requirements, installation instructions, configuration steps, project creation, running the project on an iOS Simulator, troubleshooting, and additional resources for reference.

System Requirements

Before starting, ensure that your macOS system meets the following specifications:

- **CPU**: Intel-based Mac.
- **RAM**: At least 8GB of RAM (16GB or more is recommended for smoother development).
- **macOS Version**: macOS 10.14 (Mojave) or newer.

Installation Instructions

To be able to run react native programs on android emulator you will need the following Homebrew, Node, Watchman, the React Native command line interface, a JDK(Java Development Kit), Android Studio, and a Android SDK.

Homebrew:

1. Open Terminal and type the following command:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

2. Type your admin password when prompt to.
3. A “Installation successful” message will appear in the Terminal.

Node:

1. Install Homebrew
2. Open Terminal and type the following command:

```
brew install node
```

Watchman:

1. Install Homebrew
2. Open Terminal and type the following command:

```
brew install watchman
```

Java Development Kit:

1. Install Homebrew
2. Open Terminal and type the following command:

```
brew tap homebrew/cask-versions
brew install --cask zulu11

# Get path to where cask was installed to double-click installer
brew info --cask zulu11
```

Android Studio:

1. Download Android Studio from https://developer.android.com/
2. Go thought the installation wizard, make sure the boxes with Android SDK, Android SDK Platform and Android Virtual Device are checked then click install.

Android SDK:

1. Open Android Studio:
2. Access the SDK Manager Click on "More Actions" (the gear icon) in the bottom right corner of the welcome screen. Select "SDK Manager" from the menu.
3. Navigate to SDK Platforms. In the SDK Manager, select the "SDK Platforms" tab.
4. Show Package Details. Check the box labeled "Show Package Details" at the bottom right corner.
5. Select Android 13 (Tiramisu). Find and expand the "Android 13 (Tiramisu)" entry in the list.
6. Ensure that the following items are checked:
    - Android SDK Platform 33
    - Intel x86 Atom_64 System Image or Google APIs Intel x86 Atom System Image (choose based on your machine)
    - For Apple M1 Silicon, choose Google APIs ARM 64 v8a System Image.
7. Navigate to SDK Tools. Switch to the "SDK Tools" tab in the SDK Manager.

8. Show Package Details. Check the box labeled "Show Package Details" here as well. Select Android SDK Build-Tools:
9. Find and expand the "Android SDK Build-Tools" entry. Ensure that version 33.0.0 is selected.
10. Click the "Apply" button to download and install the Android SDK and related build tools.

The macOS now has everything it needs to begin setting up the environment to create and run applications through android studio. These programs provide the emulator and a passage to connect the emulator to our projects.

Configuration steps

To build React Native apps with native code, you need to set up environment variables for Android development.

1. **Open Terminal**
2. **Edit Configuration Files**
   Depending on your shell, use a text editor (like `nano`, `vim`, or `code`) to edit your shell configuration file. Use the following commands based on your shell:
   - If you're using **zsh**, edit the `~/.zprofile` or `~/.zshrc` file:

     ```
     nano ~/.zprofile
     ```

     or

     ```
     nano ~/.zshrc
     ```

   - If you're using **bash**, edit the `~/.bash_profile` or `~/.bashrc` file:

     ```
     nano ~/.bash_profile
     ```

     ```
     or
     ```

     ```
     nano ~/.bashrc
     ```

3. **Add Environment Variables**:
   - In the opened configuration file, add the following lines at the end:

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

```
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

4. **Save and Exit**:
   - Save the file by pressing `Ctrl + O`, then press `Enter`.
   - Exit the text editor by pressing `Ctrl + X`.
5. **Apply Configuration**:
   - To apply the changes, run the following command (replace `~/.zprofile` with the appropriate file you edited):

```
source ~/.zprofile
```

```
or
```

```
source ~/.bash_profile
```

6. **Verify Configuration**:
   - Confirm that the `ANDROID_HOME` environment variable has been set by running:

```
echo $ANDROID_HOME
```

   - Verify that the appropriate directories have been added to your path by running:

```
echo $PATH
```

The macOS system is now configured with the required environment variables for Android development in React Native. These settings are essential for building and running Android apps using React Native.

Project creation

Now with the environment setup we can now make a project with react native and have the emulator running.

Creating a new application

1. **Open Terminal**
2. **Navigate to Your Preferred Project Directory**:
   - Use the `cd` command to change to the directory where you want to create your React Native project. For example, to navigate to your user's home directory, you can use:

```
cd DesiredDirectoryHere
```

3. **Create the React Native Project**:
   - Run the following command to create a new React Native project:

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

   - This command initializes a new React Native project with the latest version. It may take some time to download and set up the project files.
4. **Wait for Project Initialization**:
   - Allow the initialization process to complete. This includes downloading necessary dependencies and setting up the project structure.
5. **Project Created**:
   - Once the project creation is successful, you'll see a message indicating that the project has been created.

Preparing the Android device

1. **Open Android Studio**:
   - Launch Android Studio on your macOS.
2. **Navigate to the "AVD Manager"**:
   - In Android Studio, click on the icon that looks like the Android mascot's head (the AVD Manager) in the toolbar or go to "Tools" > "AVD Manager."
3. **Create a New Virtual Device**:
   - In the AVD Manager, click the "Create Virtual Device..." button.
4. **Select a Device Definition**:
   - Choose a device definition that matches your target Android device. You can pick any Phone from the list.
   - Click "Next."
5. **Select System Image**:
   - In the "System Image" section, locate the "Tiramisu API Level 33" image.
   - Click "Next."
6. **Configure AVD Settings**:
   - Customize your AVD by configuring settings like RAM, internal storage, and more. You can keep the default settings if unsure.
   - Click "Next."
7. **Verify AVD Configuration**:
   - Review the summary of your AVD configuration.

- Click "Finish" to create your AVD.

8. **Launch the AVD**:
   - In the AVD Manager, you should now see the newly created AVD listed.
   - Click on the Run button next to your AVD to launch it.

Running your React Native application

1. Start Metro:
   - To start Metro, run following command inside your React Native project folder:

```
npm start
```

2. Start your application
   - To start the application, run following command inside your React Native project folder:

```
npm run android
```

Trouble Shooting

A common problem that could occur when attempting to run the application is that react-native will not be able to find the Android SDK leading to the failure to launch the build. To solve this you would have to take the following steps:

1. Open the Project in the environment to modify code
2. In the "android" folder make a new file and name it local.properties
3. In the "local.properties" file write the following code:

```
sdk.dir = /Users/UsersNameHere/Library/Android/SDK
```

4. Save the program
5. Run emulator then run the application again

Resources

React Native Doc :
- https://reactnative.dev/docs/environment-setup
Youtube Tutorial:
- https://www.youtube.com/watch?v=L-EhHG3kfKM