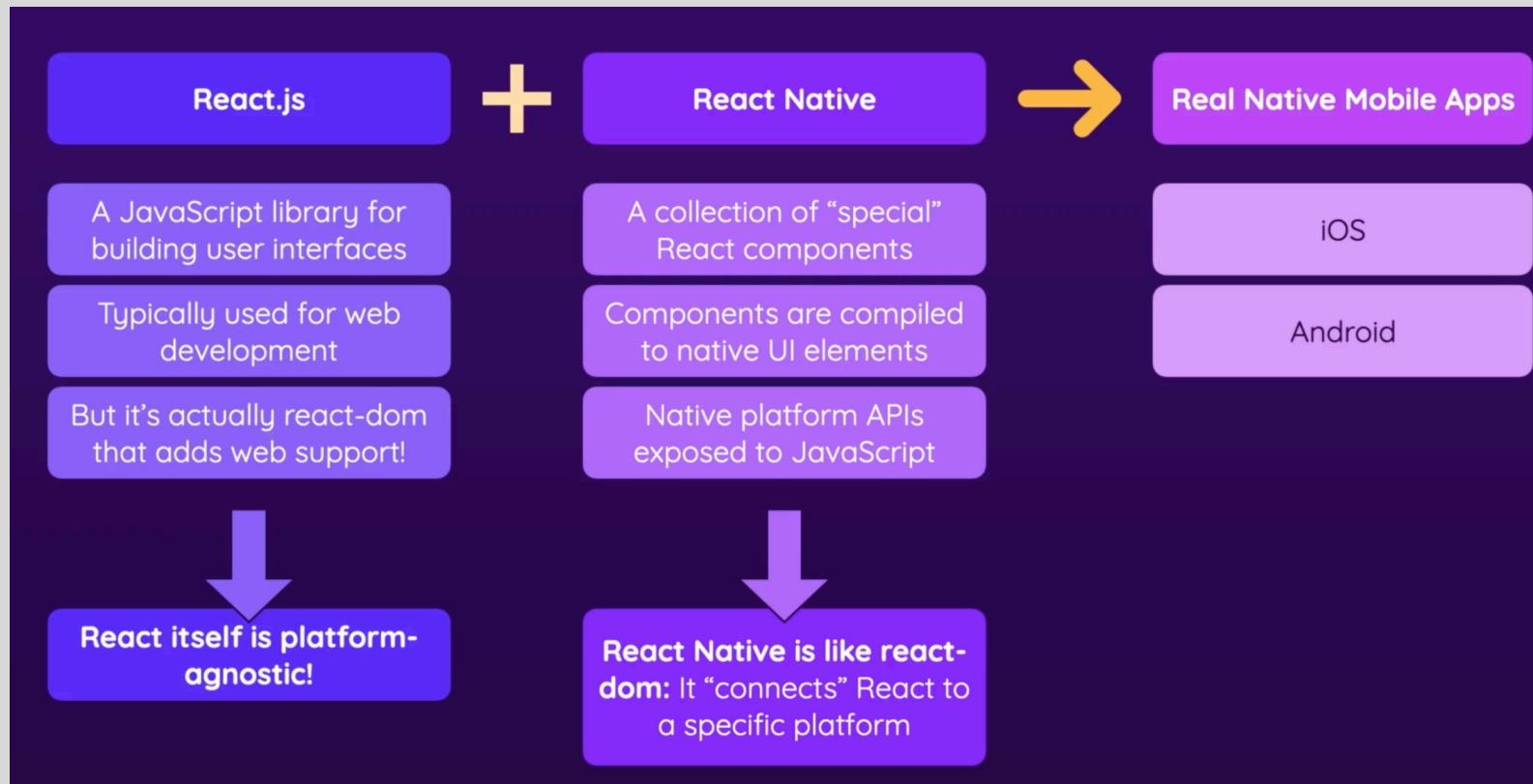


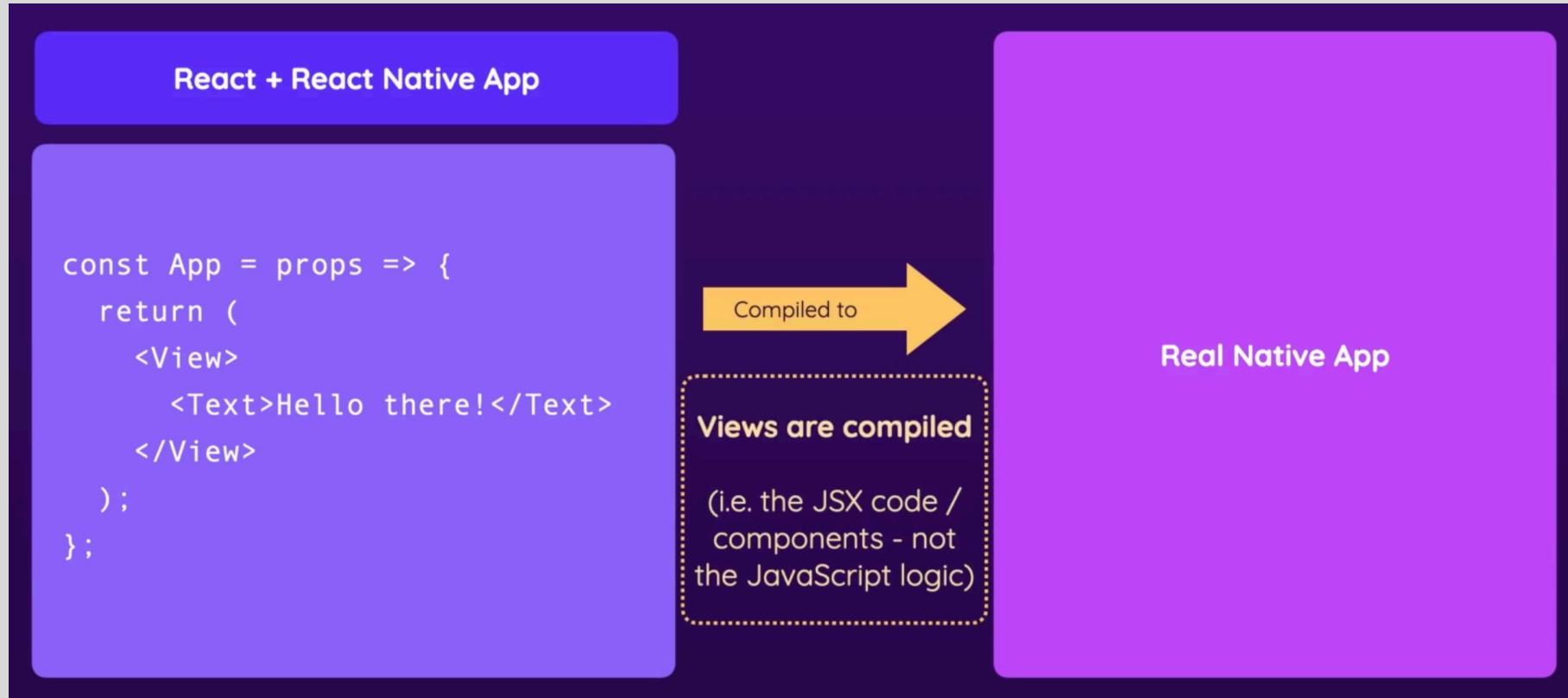
MOBİL PROGRAMLAMA

SUNU I

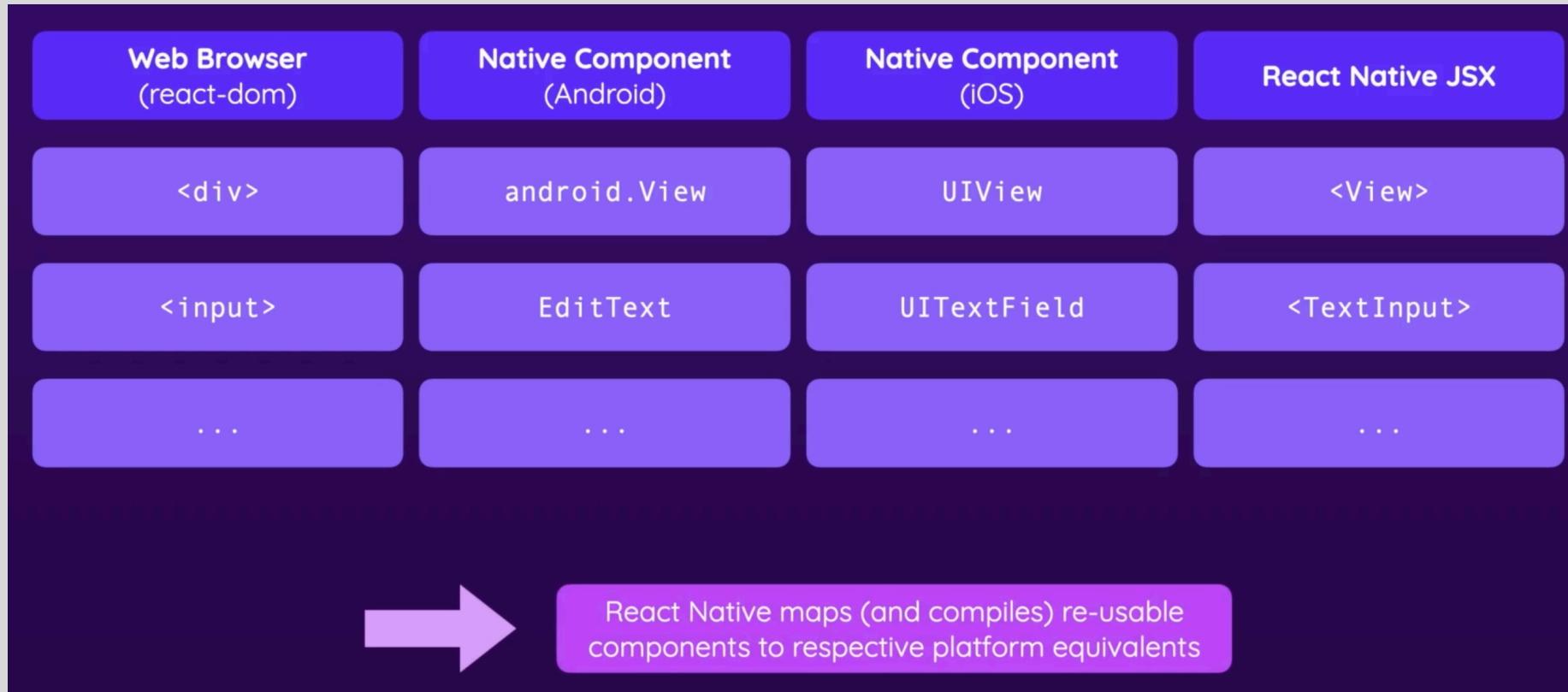
React vs React Native



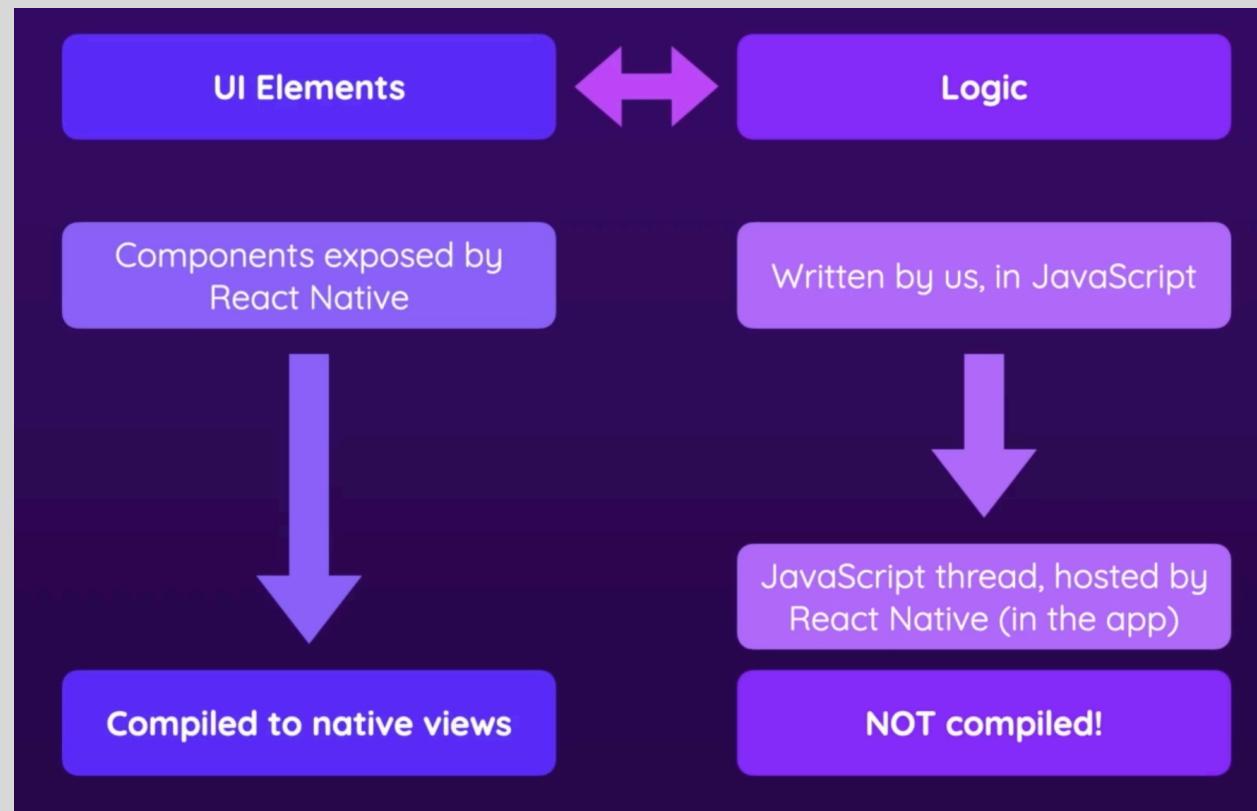
Compilation



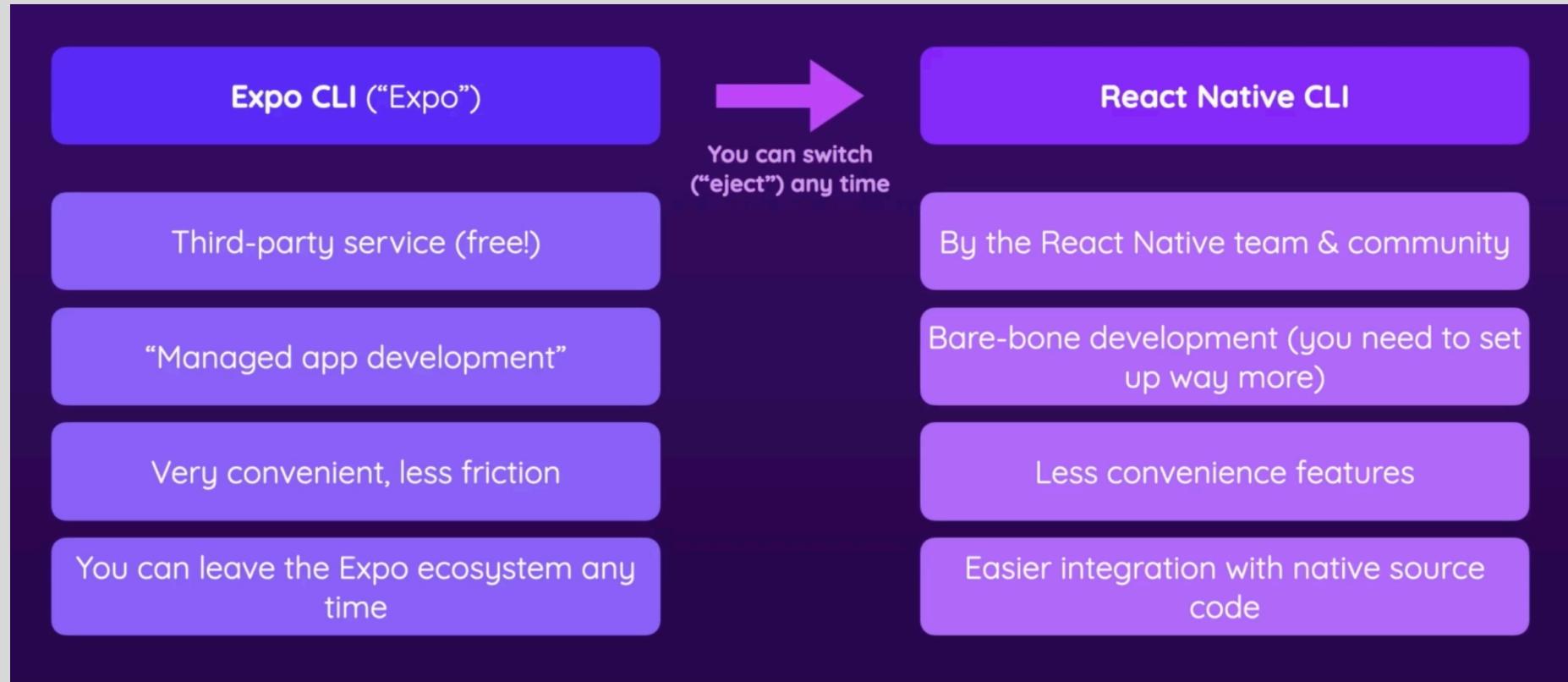
Mapping



Compile: UI vs Logic



Expo CLI vs React Native CLI



Install node.js

The screenshot shows the Node.js Downloads page. At the top, there's a navigation bar with links for Learn, About, API Docs, Download (which is underlined in green), and Certification. To the right of the navigation are a search icon, a dark mode icon, and a GitHub icon. Below the navigation, the page title is "Downloads". On the left, there's a breadcrumb trail: "HOME / DOWNLOADS". On the right, it says "LATEST LTS VERSION V16.17.1 (includes npm)". The main content area is titled "Recommended for most users" and features three download options:

- Windows Installer**: Shows the Windows logo and the file name "node-v16.17.1-x86.msi".
- MAC Installer**: Shows the Apple logo and the file name "node-v16.17.1.pkg".
- Source Code**: Shows a cube icon and the file name "node-v16.17.1.tar.gz".

Below each download option is a small "Current" badge.

React Native: Setup the Dev Environment

[Expo Go Quickstart](#)

[React Native CLI Quickstart](#)

Run the following command to create a new React Native project called "AwesomeProject":

[npm](#)

[Yarn](#)

```
npx create-expo-app AwesomeProject
```

```
cd AwesomeProject
```

```
npm start # you can also use: npx expo start
```

This will start a development server for you.

Running your React Native application

Install the [Expo Go](#) app on your iOS or Android phone and connect to the same wireless network as your computer. On Android, use the Expo Go app to scan the QR code from your terminal to open your project. On iOS, use the built-in QR code scanner of the default iOS Camera app.

Expo

Installation

To develop applications with Expo, you need two tools. A command-line tool called [Expo CLI](#) to serve your project, and a mobile client app called [Expo Go](#) to open the project on iOS and Android platforms. Additionally, you can use any web browser to run the project on the web.

- ⓘ You don't need macOS to build an iOS app with Expo. You only need an iOS device to run the Expo Go app. Windows, Linux, and macOS are all supported for your development machine.

Expo CLI

1. Expo CLI

[Expo CLI](#) is a command-line tool that is the primary interface between a developer and other Expo tools. You are going to use it for different tasks in the development life cycle of your project such as serving the project in development, viewing logs, opening the app on an emulator or a physical device, and so on.

Requirements

To use Expo CLI, you need to have the following tools installed on your developer machine:

- [Node.js LTS release](#)
- [Git](#)
- [Watchman](#), required only for macOS or Linux users

Recommended Tools

Recommended tools

- [VS Code Editor](#)
 - [VS Code Expo Extension](#) for **app.json** debugging and autocomplete
 - [Yarn](#)
- Windows users: [PowerShell](#), Bash via WSL, or the VS Code terminal

Using Expo CLI

Using Expo CLI

You can use Expo CLI without installation by leveraging `npx` — a Node.js package runner. For example, to see a list of available commands in Expo CLI, open the terminal on your development machine and run the following command:

Terminal

 Copy

```
# See a list of available commands in Expo CLI
→ npx expo -h
```

Using Expo CLI

Terminal

 Copy

```
→ npx expo whoami
```

This command checks which Expo account is currently authenticated on your machine. You will see a **Not logged in** message since you are not logged in to an Expo account. You do not need an account to start and can proceed further with your project. However, if you want to register a new Expo account, run the following command to register a new account:

Terminal

 Copy

```
→ npx expo register
```

Using Expo CLI

If you already have an Expo account, you can log in to it by running the command:

Terminal

 Copy

```
→ npx expo login
```

Expo Go App

2. Expo Go app for iOS and Android

The fastest way to get up and running is to use the [Expo Go](#) client app on your iOS or Android device. It allows you to open up apps served through Expo CLI and run your projects faster when developing them. It is available on both the iOS App Store and Android Play Store.

- [Android Play Store](#) - Android Lollipop (5) and greater
- [iOS App Store](#) - iOS 11 and greater

Expo Go App

Open the Expo Go app after it has finished installing. If you have created an account with Expo CLI, you can sign in by clicking the "Login" button in the top header on the "Home" tab. Signing in will make it easier for you to open projects in the Expo Go app while developing them — they will appear automatically under the "Projects" section on the Home tab of the app.

- ⓘ It's often useful to be able to run your app directly on your computer instead of on a separate physical device. If you would like to set this up, you can learn more about [installing an Android Emulator](#) and [installing the iOS Simulator \(macOS only\)](#) .

Android Studio Emulator

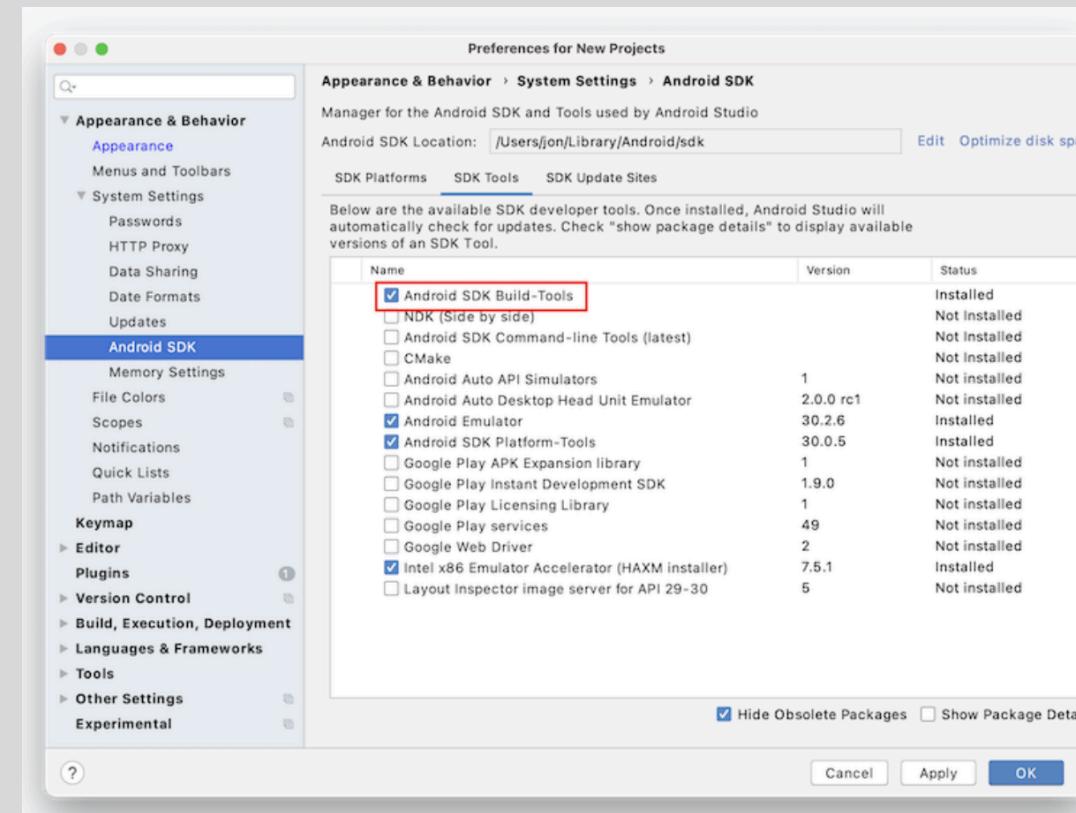
Android Studio Emulator

If you don't have an Android device available to test with, we recommend using the default emulator that comes with Android Studio. If you run into any problems setting it up, follow the steps in this guide.

Step 1: Set up Android Studio's tools

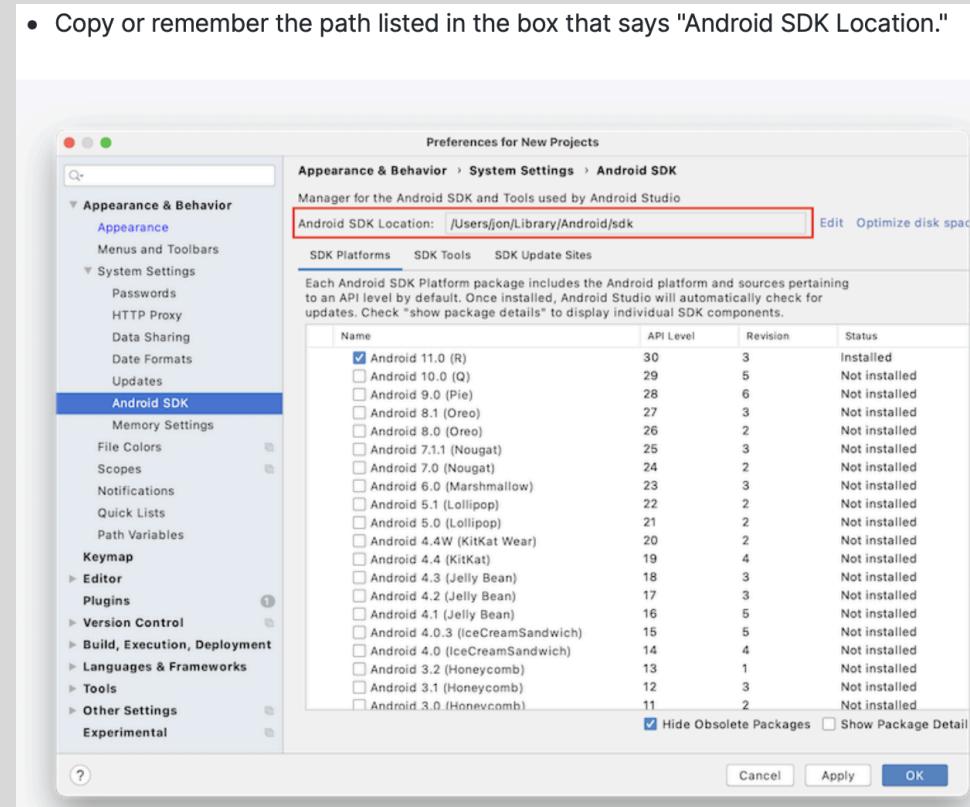
- [Download](#) and install Android Studio 3.0+.
- Select "Standard" for the "Install Type" inside the wizard.
- Inside Android Studio, go to Preferences > Appearance & Behavior > System Settings > Android SDK. Click on the "SDK Tools" tab and make sure you have at least one version of the "Android SDK Build-Tools" installed.

Android Studio Emulator



Android Studio Emulator

- Copy or remember the path listed in the box that says "Android SDK Location."



Android Studio Emulator

- If you are on macOS or Linux, add an [environment variable](#) pointing to the Android SDK location in `~/.bash_profile` (or `~/.zshenv` if you use Zsh) - eg. `export ANDROID_HOME=/your/path/here`. Copy and paste these two lines to do this automatically for Bash and Zsh:

```
[ -d "$HOME/Library/Android/sdk" ] && ANDROID_HOME=$HOME/Library/Android/sdk  
echo "export ANDROID_HOME=$ANDROID_HOME" >> ~/`[[ $SHELL == *"zsh" }}
```

Android Studio Emulator

- On macOS, you will also need to add `platform-tools` to your `~/.bash_profile` (or `~/.zshenv` if you use Zsh) - eg. `export PATH=your/path/here:$PATH`. Copy and paste this line to do this automatically for Bash and Zsh:

```
echo "export PATH=$ANDROID_HOME/platform-tools:\$PATH" >> ~/`[[ $SH
```

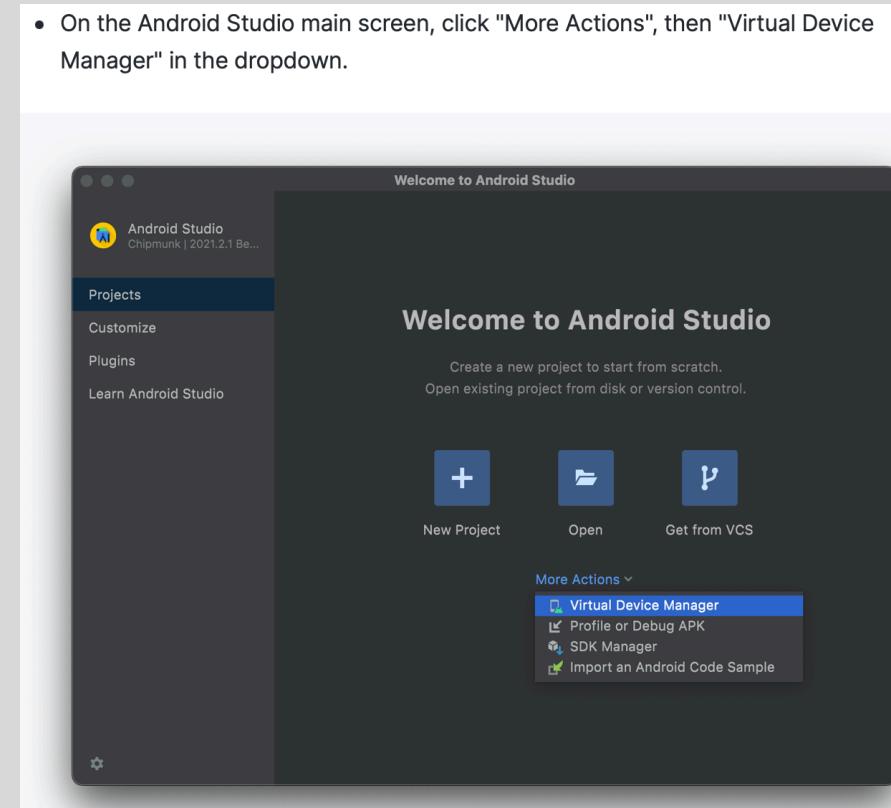
- Reload the path environment variables by running:

```
source ~/`[[ $SHELL == *"zsh" ]] && echo '.zshenv' || echo '.bash_p
```

- Finally, make sure that you can run `adb` from your terminal.

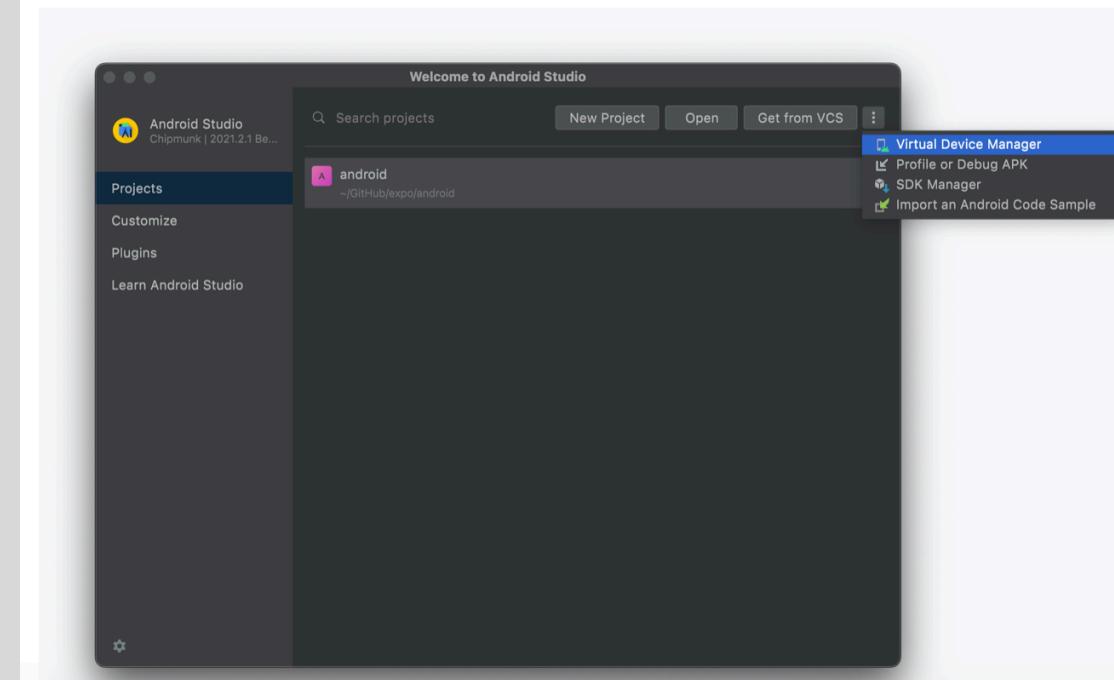
Setup a Virtual Device

- On the Android Studio main screen, click "More Actions", then "Virtual Device Manager" in the dropdown.



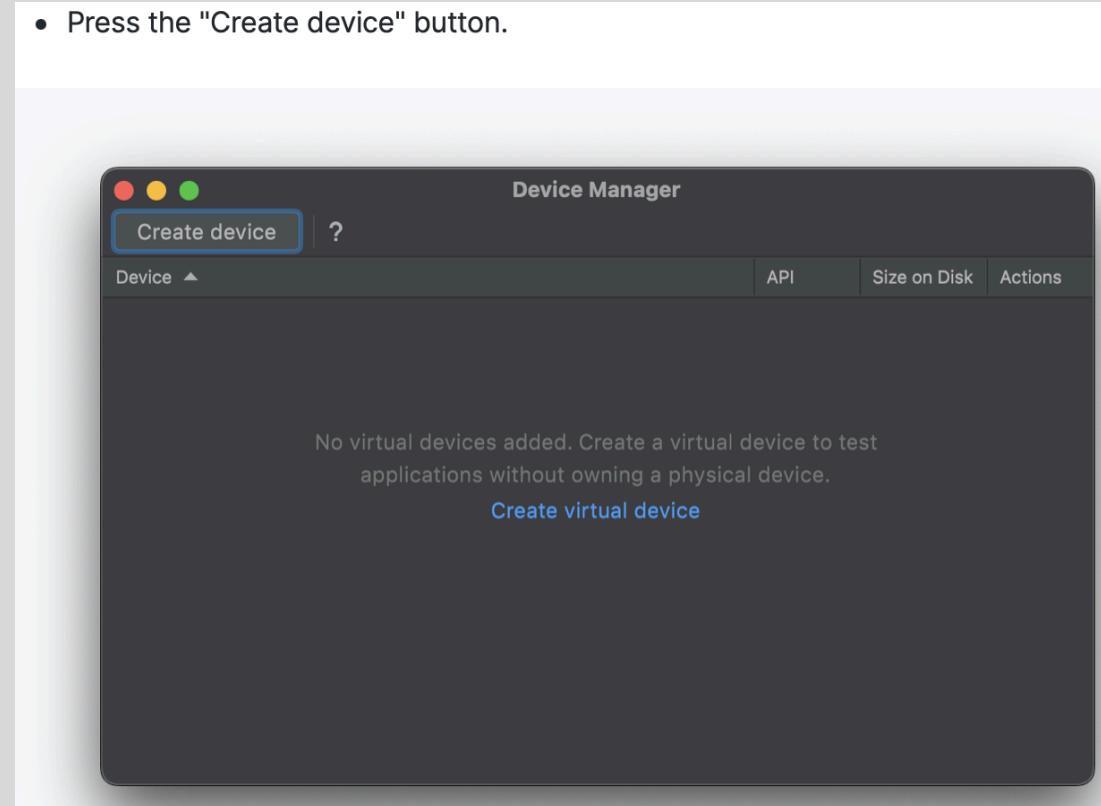
Setup a Virtual Device

If you already have a project, then the menu will show up under the three dots menu in the top right corner of the window.



Setup a Virtual Device

- Press the "Create device" button.



Setup a Virtual Device

- Choose the type of hardware you'd like to emulate. We recommend testing against a variety of devices, but if you're unsure where to start, the newest device in the Pixel line could be a good choice.
- Select an OS version to load on the emulator (probably one of the system images in the "Recommended" tab), and download the image.
- Change any other settings you'd like, and press "Finish" to create the virtual device. You can now run this device anytime by pressing the Play button in the AVD Manager window.

Multiple ADB versions

Having multiple `adb` versions on your system can result in the error `adb server version (xx) doesn't match this client (xx); killing...`

This is because the adb version on your system is different from the adb version on the android sdk platform-tools.

- Open the terminal and check the `adb` version on the system:

```
adb version
```

Multiple ADB versions

- And from the Android SDK platform-tool directory:

```
cd ~/Library/Android/sdk/platform-tools  
./adb version
```

- Copy `adb` from Android SDK directory to `usr/bin` directory:

```
sudo cp ~/Library/Android/sdk/platform-tools/adb /usr/bin
```

Create a New App

Before creating a new Expo app, you have to make sure that:

- Expo CLI is working on your development machine
- Expo Go app is installed on your iOS or Android physical device or emulator

Create a New App

Initializing the project

Terminal

 Copy

```
# Create a project named my-app
→ npx create-expo-app my-app

# Navigate to the project directory
→ cd my-app
```

Create a New App

Starting the development server

Start the development server by running the following command:

Terminal

 Copy

```
→ npx expo start
```

When you run `npx expo start` (or `yarn expo start`), Expo CLI starts [Metro Bundler](#).

This bundler is an HTTP server that compiles the JavaScript code of your app using [Babel](#) and serves it to the Expo app. Learn more about how [Expo Development Server](#) works.

Opening the App on Your Phone/Tablet

To open the app:

- On your Android device, press "Scan QR Code" on the "Home" tab of the Expo Go app and scan the QR code you see in the terminal.
- On your iPhone or iPad, open the default Apple "Camera" app and scan the QR code you see in the terminal.

You can open the project on multiple devices simultaneously. Go ahead and try it on an iPhone and Android phone at the same time if you have both handy.

Simulator and Emulator

If you are using a simulator or emulator, you may find the following Expo CLI keyboard shortcuts to be useful to open the app on any of the following platforms:

- Pressing `a` will open in an [Android Emulator or connected device](#).
- Pressing `i` will open in an [iOS simulator](#).
- Pressing `w` will open in a web browser. Expo supports all major browsers.

Make Your First Change

Open **App.js** file in your code editor and change the text to "Hello, world!". You are going to see it update on your device. This is great progress! You now have the Expo toolchain running on your machine, can edit the source code for a project, and see the changes live on your device.