

# Using Libraries

React Native provides a set of built-in Core Components and APIs ready to use in your app. You're not limited to the components and APIs bundled with React Native. React Native has a community of thousands of developers. If the Core Components and APIs don't have what you are looking for, you may be able to find and install a library from the community to add the functionality to your app.

## Selecting a Package Manager

React Native libraries are typically installed from the npm registry using a Node.js package manager such as npm CLI or Yarn Classic.

If you have Node.js installed on your computer then you already have the npm CLI installed. Some developers prefer to use Yarn Classic for slightly faster install times and additional advanced features like Workspaces. Both tools work great with React Native. We will assume npm for the rest of this guide for simplicity of explanation.

💡 The terms "library" and "package" are used interchangeably in the JavaScript community.

## Installing a Library

To install a library in your project, navigate to your project directory in your terminal and run the installation command. Let's try this with `react-native-webview`:

npm   **Yarn**

```
yarn add react-native-webview
```

The library that we installed includes native code, and we need to link to our app before we use it.

## Linking Native Code on iOS

React Native uses CocoaPods to manage iOS project dependencies and most React Native libraries follow this same convention. If a library you are using does not, then please refer to their README for additional instruction. In most cases, the following instructions will apply.

Run `pod install` in our `ios` directory in order to link it to our native iOS project. A shortcut for doing this without switching to the `ios` directory is to run `npx pod-install`.

```
npx pod-install
```

Once this is complete, re-build the app binary to start using your new library:

npm   **Yarn**

---

```
yarn ios
```

## Linking Native Code on Android

React Native uses Gradle to manage Android project dependencies. After you install a library with native dependencies, you will need to re-build the app binary to use your new library:

npm   **Yarn**

---

```
yarn android
```

# Finding Libraries

[React Native Directory](#) is a searchable database of libraries built specifically for React Native. This is the first place to look for a library for your React Native app.

Many of the libraries you will find on the directory are from [React Native Community](#) or [Expo](#).

Libraries built by the React Native Community are driven by volunteers and individuals at companies that depend on React Native. They often support iOS, tvOS, Android, Windows, but this varies across projects. Many of the libraries in this organization were once React Native Core Components and APIs.

Libraries built by Expo are all written in TypeScript and support iOS, Android, and `react-native-web` wherever possible.

After React Native Directory, the [npm registry](#) is the next best place if you can't find a library specifically for React Native on the directory. The npm registry is the definitive source for JavaScript libraries, but the libraries that it lists may not all be compatible with React Native. React Native is one of many JavaScript programming environments, including Node.js, web browsers, Electron, and more, and npm includes libraries that work for all of these environments.

## Determining Library Compatibility

### Does it work with React Native?

Usually libraries built *specifically for other platforms* will not work with React Native. Examples include `react-select` which is built for the web and specifically targets `react-dom`, and `rimraf` which is built for Node.js and interacts with your computer file system. Other libraries like `lodash` use only JavaScript language features and work in any environment. You will gain a sense for this over time, but until then the easiest way to find out is to try it yourself. You can remove packages using `npm uninstall` if it turns out that it does not work in React Native.

## Does it work for the platforms that my app supports?

React Native Directory allows you to filter by platform compatibility, such as iOS, Android, Web, and Windows. If the library you would like to use is not currently listed there, refer to the README for the library to learn more.

## Does it work with my app version of React Native?

The latest version of a library is typically compatible with the latest version of React Native. If you are using an older version, you should refer to the README to know which version of the library you should install. You can install a particular version of the library by running `npm install <library-name>@<version-number>`, for example: `npm install @react-native-community/netinfo@^2.0.0`.

Is this page useful?



Edit this page

Last updated on **Jun 21, 2023**