

# Overview

---

- This project has been built based on Laravel's UI (<https://github.com/laravel/ui>) reactjs preset.
- Assume that your project is in `C:\Projects\react-laravel-sample-project`

## Software requirement

---

- [nodejs/npm \(version 16.14.1 LTS as of time of writing this guide\)](#)

OR

- [Docker](#)

## How to run with Docker

---

- Download and install docker from [here](#)
- Use following commands to start the dev server (for both back-end and front-end)

```
cd C:\Projects\react-laravel-sample-project  
docker-compose up
```

- If it's first time, it will take sometime to build container, when it's done you can access <http://localhost:3000>, front-end will be served from PHP server

## How to run on Windows host

---

- install nodejs/npm

run with hot reload (browser auto reload assets everytime you make changes)

- open cmd and run following command

```
cd C:\Projects\react-laravel-sample-project\src  
npm install  
npm run hot
```

- open another cmd and run following command

```
cd C:\Projects\react-laravel-sample-project\src  
npm run web-sever
```

or if you have php and composer installed locally

```
composer install  
php artisan serve --port 3000
```

- open web browser and access <http://localhost:3000>

Run with watch (assets will be compiled automatically but you will have to refresh browser manually to see changes)

- open cmd and run following command

```
cd C:\Projects\react-laravel-sample-project\src  
npm install  
npm run watch
```

- open <public/index.html> and modify [line 13](#) and [line 18](#), change <//localhost:8080> into <//localhost>
- open another cmd and run following command

```
cd C:\Projects\react-laravel-sample-project\src  
npm run web-server
```

- open web browser and access <http://localhost:3000>

## Project dependencies

---

- [React](#)
- [React Router](#)
- [Recoil \(react's state management library\)](#)
- [Bootstrap](#)
- [Shoelace](#)
- [Blueprintjs](#)
- [Animate CSS Grid](#)
- [Axios](#)

## create new pages and components

---

- new page will be registered in [resources/js/Router.jsx](#), component file for page will be created in [resources/js/](#)

- please refer to other components and official documentation to know how to compose new components

## state management

- this project use recoil to manage states, it utilizes react's hook, therefore any component that use state need to be defined in function style;
- for global states, please define new atom in `state/index.js` (or new module file if necessary) to access these states in any component (for example: `compactSidebar` state, which is accessed in root, sidebar and header)
- for local state, please define it right in the component (for example: users list, see `ExampleUsersList.jsx` for more information)

## styles

---

- since all major browsers support css variable (custom property) natively, please refrain from using sass's variable and use native variable whenever possible
- refer to `_variables.scss` and other `.scss` file to know how to define and use css variables

## note

---

- when incorporate with other back-end like laravel, there's no need to keep `public/web-server.js`, this one only serve the purpose of standalone front-end dev workflow