PACKAGE MANAGERS

CSCI2720 2022-23 Term 1

Building Web Applications

Dr. Chuck-jee Chau
chuckjee@cse.cuhk.edu.hk

香港中文大學
The Chinese University of Hong Kong

OUTLINE

- Packages and libraries

- Package managers: npm, yarn, pnpm, …

- npm

- npx

- create-react-app

PACKAGES AND LIBRARIES

- It is possible for web development done by one person, yet…

  - Are you willing to write all the HTML, CSS, JS code by yourself alone?

- Modern development: incorporating building blocks for efficiency and convenience

  - Packages, libraries, frameworks, …

  - *e.g.,* Bootstrap for CSS, jQuery for JS, React for apps, …

- An important issue: **dependency**

  - Do different versions of things work together?

PACKAGE MANAGER

- Finding correct package JS files

- Checking to ensure absence of *vulnerabilities*

- Downloading, managing, and putting them in *proper locations*

- Including packages in the development application

- Handling *sub-dependencies*

- *Clearing files* for unnecessary packages

- See: *https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Understanding_client-side_tools/Package_management*

NPM, YARN, PNPM, ...

- Generally, the package managers are executed in the command line interface
  - More sophisticated option settings
  - More convenient for **scripting and automation**

- npm: *www.npmjs.com*

- Yarn: *yarnpkg.com*

- pnpm: *pnpm.js.org*

- Which one to use? Too many comparisons on Internet...
  - *See: https://blog.logrocket.com/javascript-package-managers-compared/*

NPM

- Part of Node.js, originally as the "*Node Package Manager*"
  - Since January 2010
- To obtain *npm*, download for your platform with *Node.js*
  - *https://nodejs.org/en/download/*
- *Local packages*: in a folder **node_modules** in project
- *Global packages*: in system folders if you have admin rights
- Most packages are working on the Node.js environment for backend development
  - *We will use Node.js in the later part of the semester*
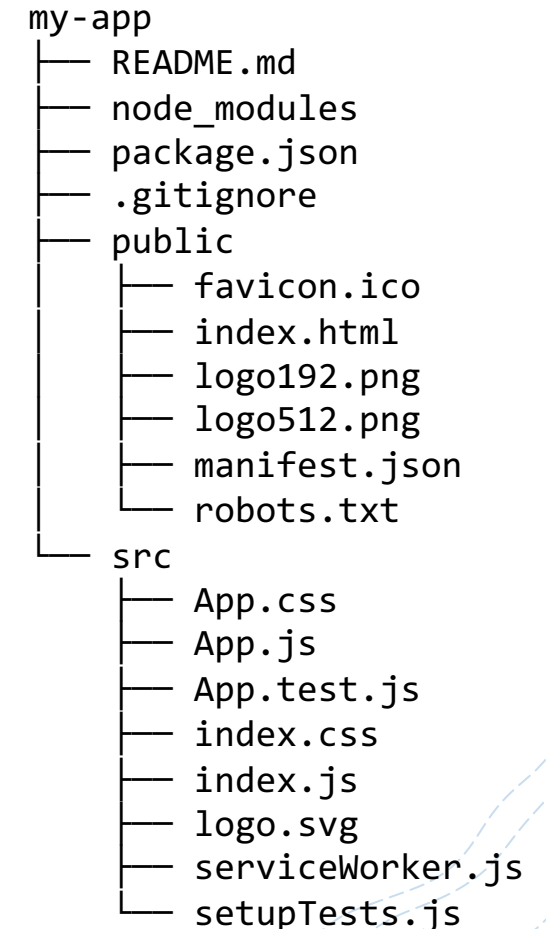
NPM COMMANDS

- Initializing a project
  - **npm init**

- Installing a local package
  - **npm install** *(package name)*

- Installing a global package (*admin rights required*)
  - **npm install -g** *(package name)*

- Installing all packages as defined in **package-lock.json**
  - **npm install**

NPX

- While **npm** only manages the packages, a supplementary tool **npx** allows you to execute Node packages without installing
  - Run scripts in packages
  - Run gists or code in GitHub
  - *e.g.,* **npx create-react-app my-app**
- *See more: https://blog.npmjs.org/post/162869356040/introducing-npx-an-npm-package-runner*

CREATE-REACT-APP

```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   ├── favicon.ico
│   ├── index.html
│   ├── logo192.png
│   ├── logo512.png
│   ├── manifest.json
│   └── robots.txt
└── src
    ├── App.css
    ├── App.js
    ├── App.test.js
    ├── index.css
    ├── index.js
    ├── logo.svg
    ├── serviceWorker.js
    └── setupTests.js
```

- A very common way to prepare a React app (which you can find in a lot of tutorials) is to use **create-react-app**
  - After installing *Node.js* and *npm*, only this command is needed to build the skeleton of a simple React app

    npx create-react-app *theAppName*

  - Then, this command will transpile the code, build the app, and start a web server on the local machine in development mode

    npm start

  - To build the app for production, run this command for optimized performance with HTML/CSS/JS in the build folder

    npm run build

- *See:* *https://create-react-app.dev/docs/getting-started*

READ FURTHER…

A Beginner's Guide to npm, the Node Package Manager

https://www.sitepoint.com/npm-guide/

npm CLI Commands

https://docs.npmjs.com/cli/v8/commands