React

The library for web and native user interfaces

React is a popular JavaScript library for building user interfaces. It provides a declarative and efficient way to create interactive UI components. Here's an overview of some core concepts and features of React:

React is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front end library responsible only for the view layer of the application.

It was created by **Jordan Walke,** who was a software engineer at **Facebook.** It was initially developed and maintained by Facebook and was later used in its products like **WhatsApp** & **Instagram.**

Facebook developed ReactJS in **2011** in its newsfeed section, but it was released to the public in the month of **May 2013.**

The current version of React is **React 18.2.0**, which was released on **14 June 2022**. **React 18** introduces a number of new features, including:

Suspense: A new feature that allows you to render your components in a suspenseful way, which can help to improve performance.

Lazy loading: A new feature that allows you to lazy load your components, which can also help to improve performance.

ReactDOM Server: A new server-side rendering API that allows you to render your React components on the server.

Visual Studio Code(IDE)

Visual Studio Code (often abbreviated as **VS Code**) is a free, open-source code editor developed by Microsoft. It's a lightweight but powerful source code editor that runs on your desktop and is available for Windows, macOS, and Linux. Here are some key features and details about VS Code:

1. Extensions: One of the most powerful features of VS Code is its extensibility. Users can install extensions to add new languages, themes, debuggers, and to connect to additional services. Extensions run in separate processes, ensuring they won't slow down the editor.

2. Integrated Git Support: VS Code has built-in Git integration, allowing you to commit, pull, and push your changes to a repository directly from the editor.

3. Debugging: VS Code comes with built-in debugging support, allowing you to set breakpoints, inspect variables, and control the execution flow of your programs.

4. IntelliSense: Provides smart completions based on variable types, function definitions, and imported modules.

5. Terminal: Integrated terminal that allows you to stay within VS Code while using the command line, whether it's for building, running scripts, or using Git commands.

6. Customizable: You can customize almost every aspect of the editor, from keyboard shortcuts to visual themes.

7. Languages: Out of the box, VS Code supports a wide range of programming languages, and with the addition of extensions, the list grows even longer.

8. Live Share: This feature allows developers to collaborate in real-time, sharing their VS Code instances for pair programming and debugging.

9. Remote Development: With the Remote Development extensions, you can develop on a remote machine, container, or even the Windows Subsystem for Linux, just as if you were running VS Code locally.

10. Performance: VS Code is built on Electron, but it's optimized to be lightweight and responsive.

11. Regular Updates: Microsoft releases new versions of VS Code monthly, ensuring that the community gets regular updates with new features and bug fixes.

12. Community: Due to its open-source nature, VS Code has a large and active community that contributes to its development, creates extensions, and provides support to other users.

https://code.visualstudio.com/download

React

Introduction

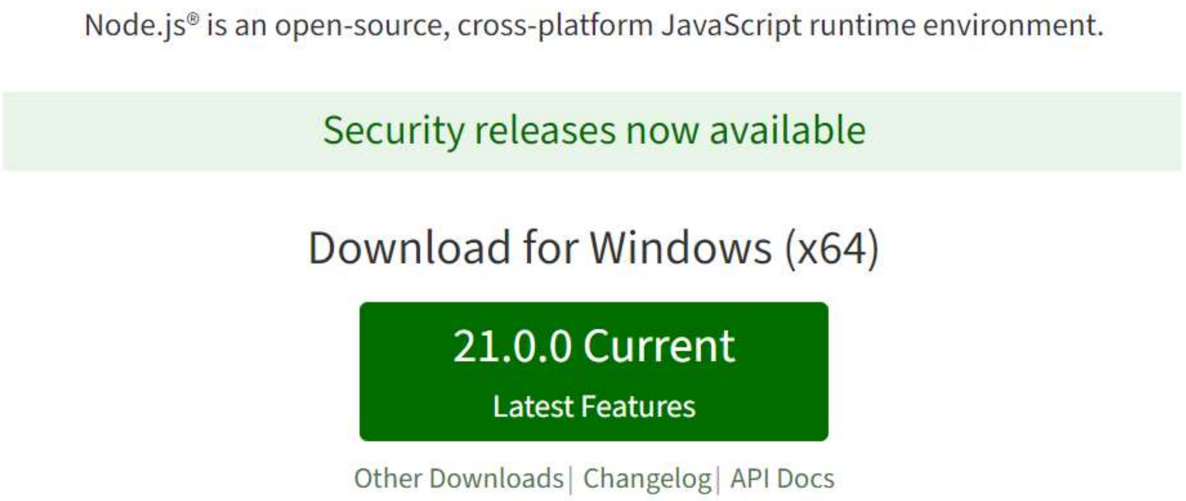
- ✓ React is the UI library.
- ✓ React library given by facebook.
- ✓ React acting as UI Layer in web applications.
- ✓ ReactJS can interact with the Angular,VueJS,.....
- ✓ we will develop ReactJS Applications by using JSX.
- ✓ "JSX" stands for JavaScript + XML.
- ✓ in general, browsers won't understand XML.
- ✓ so, we must convert XML to Equalent JavaScript.
- ✓ React simplifies the Complex UI with the help of Components.
- ✓ React Applications are Component Based Applications.
- ✓ React Components are Reusable.
- ✓ "babel" is the tool used to convert the XML to Equalent JavaScript.
- ✓ "React" Applications are faster applications, because of virtual DOM.

ReactJS Environmental Setup

1) download and install NodeJS

- ➢ React Installation Depending on Node Server.
- ➢ "npm" is the tool available in NodeJS, helps to install the React.
- ➢ "npm" stands for node packaging manager.

```
website : https://nodejs.org/en/

file    : node-v14.4.0-x64.msi
```



Node.js® is an open-source, cross-platform JavaScript runtime environment.

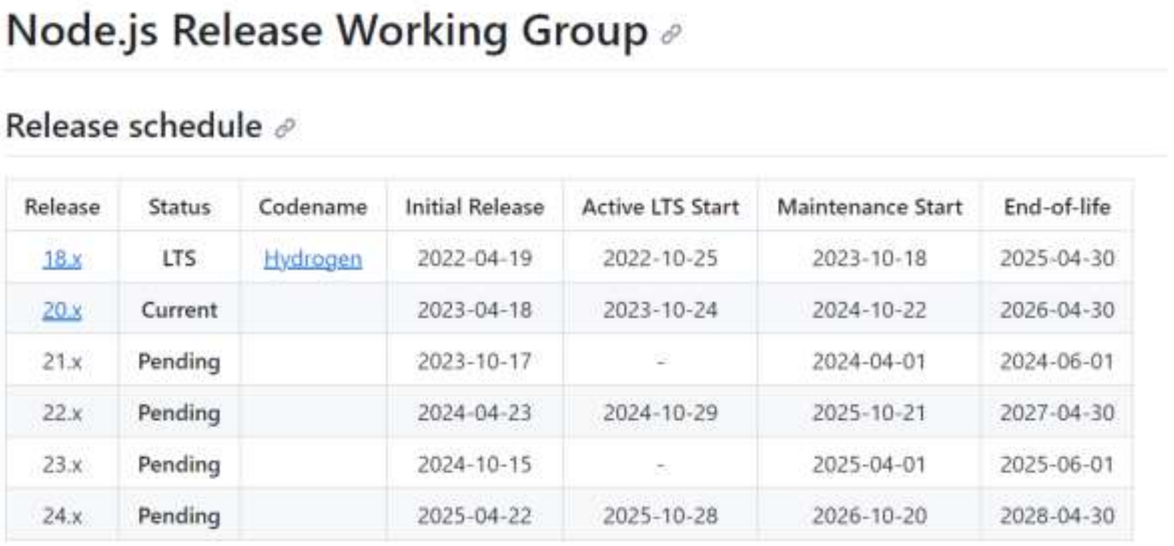
Security releases now available

Download for Windows (x64)

21.0.0 Current

Latest Features

Other Downloads | Changelog | API Docs



Node.js Release Working Group

Release schedule

Release	Status	Codename	Initial Release	Active LTS Start	Maintenance Start	End-of-life
18.x	LTS	Hydrogen	2022-04-19	2022-10-25	2023-10-18	2025-04-30
20.x	Current		2023-04-18	2023-10-24	2024-10-22	2026-04-30
21.x	Pending		2023-10-17	-	2024-04-01	2024-06-01
22.x	Pending		2024-04-23	2024-10-29	2025-10-21	2027-04-30
23.x	Pending		2024-10-15	-	2025-04-01	2025-06-01
24.x	Pending		2025-04-22	2025-10-28	2026-10-20	2028-04-30

2) install yarn tool

- "yarn" tool provided by facebook
- "yarn" tool used to download the libraries.
- we will install yarn tool by using command.

```
> npm install -g yarn@latest
```

   - "npm" stands for node packaging manager

   - "-g" stands for global installation

3) install "create-react-app" tool

- "create-react-app" tool used to create the "react applications"
- "create-react-app" tool provided by facebook.
- we will install "create-react-app" tool by using npm.

```
> npm install -g create-react-app
```

For developing a React application, the setup of tools can be a time-consuming and daunting task. In any React application, there are plenty of moving parts. You may want to set up Babel to transpile JSX into browser-ready code.

Also, the need may arise to configure Webpack to pack your project assets. For all such tasks, you may have to invest a lot of time and effort.

But the Create React App presents an easier approach to provide you with great convenience. Especially for a react developer, this tool proves to be practical. The present post familiarizes you with everything you need to know about Create React App, so let's get started:

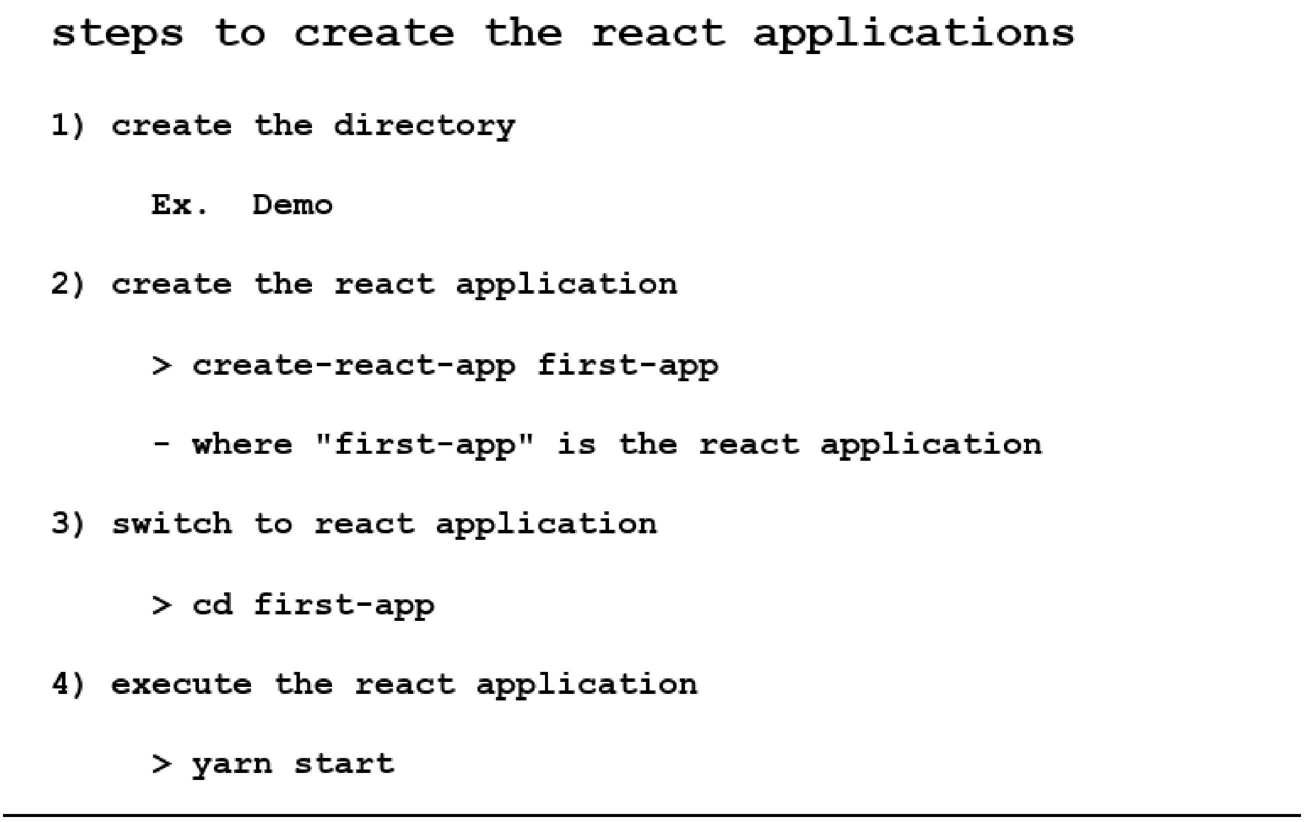
What is Create a React App?

Built by developers at Facebook, Create React App is a tool (built by developers at Facebook) that provides you with a huge dominance while building React apps. You will get rid of tedious setup and configuration.

All you need to do is run a single command and the Create React App will set up the tools you require to begin your React project.

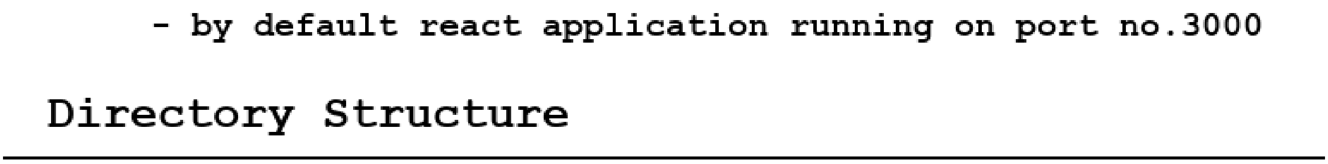
Whether you want to develop apps or learn React, Create React App proves to be a comfortable environment. It works as the best approach for starting building a new single-page app in the React environment.

Moreover, it deploys your development environment to let you use the cutting-edge functionalities of JavaScript features



steps to create the react applications

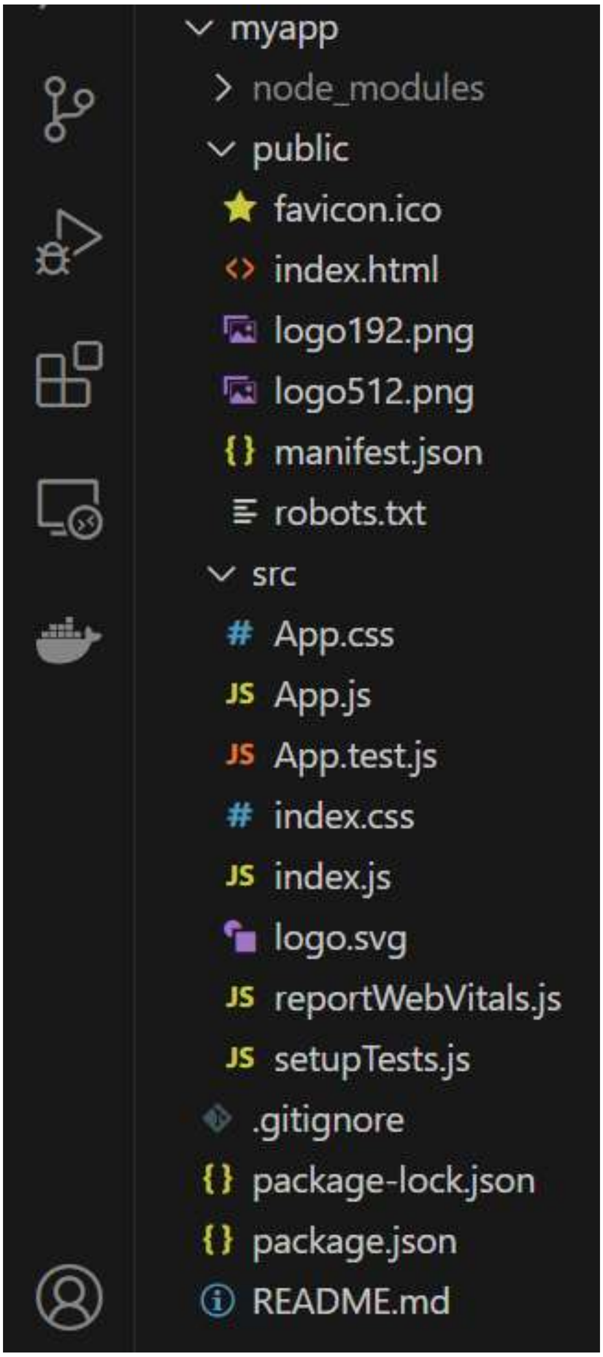
```
1) create the directory

     Ex.  Demo

2) create the react application

     > create-react-app first-app

     - where "first-app" is the react application

3) switch to react application

     > cd first-app

4) execute the react application

     > yarn start
```



```
     - by default react application running on port no.3000
```

Directory Structure

React Project Structure



```
myapp
  node_modules
  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
    reportWebVitals.js
    setupTests.js
  .gitignore
  package-lock.json
  package.json
  README.md
```

Here's an overview of each file and directory in the

This structure is set up by Create React App (CRA) to give developers a quick start with React

1. **node_modules:** This directory contains all the dependencies and libraries required for your project. These get installed when you run **npm install.**

2. **public:** Holds static files and assets that don't go through Webpack.

- **favicon.ico:** The small icon displayed next to your website's title in browser tabs.
- **index.html:** The main HTML file. React will inject its components into a div with an id of "root" inside this file.
- **logo192.png & logo512.png:** Logo images in different resolutions, often used for app icons on various devices and in the browser.
- **manifest.json:** Provides information about the application (like name, author, description, and more) in a JSON format. It's used when you want to make your web app installable as a Progressive Web App (PWA).

- robots.txt: A file used by search engines to understand which pages or sections of your site shouldn't be crawled or indexed.

3. src: The primary directory where your React source code resides.

    - App.css: Contains the styles for the App component.
    - App.js: The main React component for your application.
    - App.test.js: Contains tests related to the App component, utilizing Jest, the testing tool that comes with Create React App (CRA).
    - index.css: Contains global styles for your application.
    - index.js: The JavaScript entry point file. It initializes the React application and renders the App component.
    - logo.svg: A default SVG logo provided by CRA.
    - reportWebVitals.js: A utility for measuring and reporting web page performance.
    - setupTests.js: Configuration for setting up tests in the CRA environment.

4. **.gitignore:** Specifies which files or directories should be ignored by Git. For instance, **node_modules** is typically excluded from being tracked because it contains a large number of files and can be re-created by running `npm install`.

5. **package-lock.json:** Automatically generated by npm. It ensures that installations are consistent across different environments by detailing the exact version and configuration of installed packages.

6. **package.json:** Defines the project dependencies, scripts, and other metadata. It's a crucial file for npm and yarn to know which packages to install and how to run certain tasks.

7. **README.md:** A documentation file written in Markdown format. It usually provides an introduction to the application, how to run it, and other related information.

Key Concepts in React:

1. Components:

React applications are built using components. Components are reusable, self-contained building blocks that encapsulate the UI and its behavior. Components can be either functional or class-based, and they can be composed together to create complex user interfaces.

2. JSX (JavaScript XML):

JSX is a syntax extension for JavaScript that allows you to write HTML-like code within your JavaScript files. It makes it easier to define the structure and content of React components. JSX is transpiled to regular JavaScript using tools like Babel.

```
import React from 'react';

const App = () => {
  return (
    <div>
      <h1>Hello, React!</h1>
      <p>This is a simple example of JSX.</p>
    </div>
  );
};

export default App;
```

Explanation:

import React from 'react';

- This line imports the React library from the 'react' module. It's necessary to use React components.

const App = () => {

- Here, a functional component named '**App**' is being defined using an arrow function.

return (

- This starts the return statement for the '**App**' component.

<div>

- This is the start of a **JSX** div element. JSX allows you to write HTML-like syntax within your JavaScript code.

<h1>Hello, React!</h1>

- This is a **JSX h1** element displaying the text 'Hello, React!'.

<p>This is a simple example of JSX.</p>

- This is a JSX paragraph element displaying the text about **JSX**.

</div>

- This closes the **JSX** div element.

);

- This closes the return statement.

};

- This closes the '**App**' functional component definition.

export default App;

This line exports the 'App' component as the default export from this module. This allows other modules to import and use the 'App' component

3. Virtual DOM:

React uses a virtual DOM (a lightweight representation of the actual DOM) to optimize rendering performance. When there are updates to the UI, React creates a virtual representation of the changes and then efficiently updates the real DOM to reflect those changes.

The Virtual DOM is a core concept in React that helps optimize rendering performance. It's a lightweight representation of the actual DOM and acts as an intermediary between your React components and the browser's DOM. Here's an example to illustrate how the Virtual DOM works: