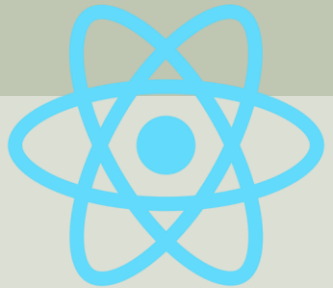
https://www.halvorsen.blog



Getting Started with React

React is a front-end library for creating UI

Hans-Petter Halvorsen

Contents

https://www.halvorsen.blog

Getting Started with React

Introduction to React

Hans-Petter Halvorsen

Table of Contents

React

- The React library is an open-source JavaScript library you can use to create dynamic and appealing applications with rich UI (User Interfaces).
- Basically, React is a front-end library for creating UI.
- React is a frontend (client-side) JavaScript library, but you can also use TypeScript.
- React is a UI library that uses a virtual DOM to manipulate the web application's user interface.

React

- React is a **front-end JavaScript library** for building user interfaces in Web Applications.
- React is also known as “React.js” or “ReactJS”.
- Developed by Facebook in 2011. React is now free and open source.
- React mainly focus on the user interface and rendering components to the DOM.
- Note! React is a “front-end Library“ and not a “Framework”.
  - So, you typically need a Framework like Next.js or ASP.NET Core, etc.
- Homepage: https://react.dev.

DOM

- DOM - Document Object Model.
- The Document Object Model (DOM) connects web pages to scripts or programming languages by representing the structure of a document - such as the HTML representing a web page - in memory.
- The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. DOM methods allow programmatic access to the tree. With them, you can change the document's structure, style, or content.
- The backbone of an HTML document is tags. According to the Document Object Model (DOM), every HTML tag is an object. Nested tags are “children” of the enclosing one. The text inside a tag is an object as well.
- JavaScript can change/manipulate the DOM.
- Instead of manipulating the web browser DOM directly, React creates a virtual DOM in memory.

https://en.wikipedia.org/wiki/Document_Object_Model

JSX

- React uses something called JSX.
- JSX stands for “JavaScript XML”.
- JSX allows us to write HTML in React.
- JSX makes it easier to write and add HTML in React.
- You don’t need to use JSX, but JSX makes it easier to write React Applications.

React Resources

- React Homepage: https://react.dev.
- React Quick Start: https://react.dev/learn.
- React Tutorial: https://www.w3schools.com/react/.
- Tutorial: Create an ASP.NET Core app with React in Visual Studio: https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-asp-net-core-with-react.

https://www.halvorsen.blog

Getting Started with React

Hans-Petter Halvorsen

Table of Contents

Getting Started with React

- You can start using React directly in HTML.
  - This can be used getting started and for testing.
- For creating real React Applications you should install a React Environment on your computer.
  - More about that later in this tutorial.
  - You can then use NPM (Node Package Manager) which is included with Node.js.
  - You can also use a build tool like Vite.
  - You can also use Visual Studio to create an React App.

React Example

```
..

function MyApp() {
    return <h1>Hello, world!</h1>;
}

const container = document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(<MyApp />);

..
```

Test out React – Hello World

- To use React in production, you need **npm** which is included with **Node.js**.
- To start learning React you can write React directly in your HTML files, as you see in this example.

You need to include these 3 CDNs:

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World</title>
    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>
    <script type="text/babel">

      function MyApp() {
        return <h1>Hello, world!</h1>;
      }

      const container = document.getElementById('root');
      const root = ReactDOM.createRoot(container);
      root.render(<MyApp />);

    </script>
  </body>
</html>
```

Content Delivery Network (CDN)

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World</title>
    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>
    <script type="text/babel">

      function MyApp() {
        return <h1>Hello, world!</h1>;
      }

      const container = document.getElementById('root');
      const root = ReactDOM.createRoot(container);
      root.render(<MyApp />);

    </script>
  </body>
</html>
```

Babel

- Babel is a JavaScript compiler that can translate markup or programming languages into JavaScript.
- React uses Babel to convert JSX into JavaScript.
- JSX is a syntax extension for JavaScript and JSX is used to convert HTML tags into react elements.

React Basics

- JSX – JSX is a syntax extension for JavaScript and JSX is used to convert HTML tags into react elements.
- Components - Components are reusable UI elements
- Props - Props (or Properties) are arguments passed into React Components

React **JSX**

- JSX is a syntax extension for JavaScript that lets you write HTML-like markup inside a JavaScript file.
- Basically, JSX converts HTML tags into react elements.
- JSX is short for “JavaScript XML”.
- JSX looks a lot like HTML, but it is a bit stricter and can display dynamic information.
- JSX and React are two separate things. They’re often used together, but you can use them independently of each other.
  - JSX is a syntax extension for JavaScript, while React is a JavaScript library.
- JSX is an extension of the JavaScript language and is translated into regular JavaScript at runtime.

https://www.halvorsen.blog

Getting Started with React

React Components and Props

Hans-Petter Halvorsen

Table of Contents

React Example

```
..

function MyApp() {
    return <h1>Hello, world!</h1>;
}

const container = document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(<MyApp />);

..
```

React **Components**

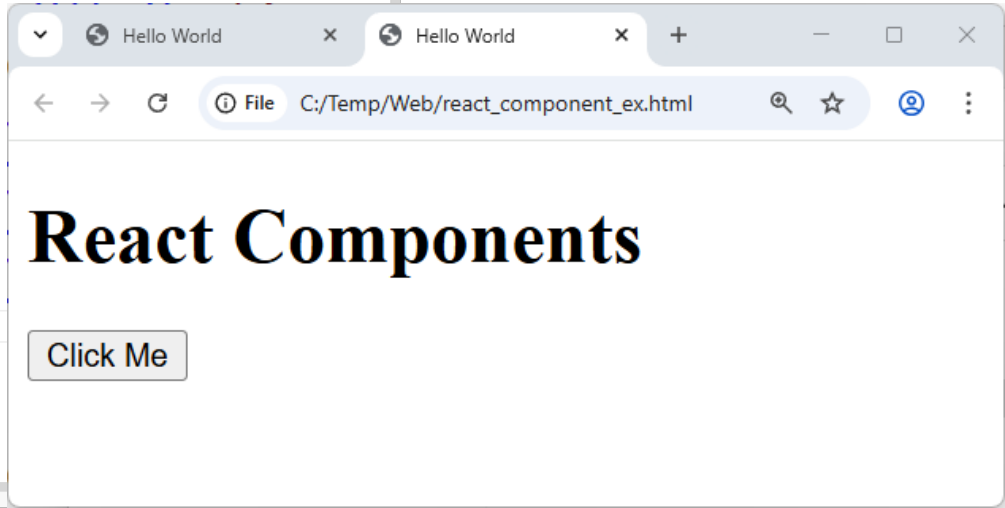
- In React you create and use Components.
- You create and use Components to build your User Interfaces (UI).
- In a React app, every piece of UI is a Component.
- Components are independent and reusable pieces of code.
- Basically, Components are reusable UI elements for your React Web Application.
- Components serve the same purpose as JavaScript functions but work in isolation and return HTML.
- React Components are regular JavaScript functions, but their names must start with a capital letter, or they will not work!

Component Example

```
function Button(props) {
    return <button>Click Me</button>;
}

const container =
document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(<Button />);
```

Component Example

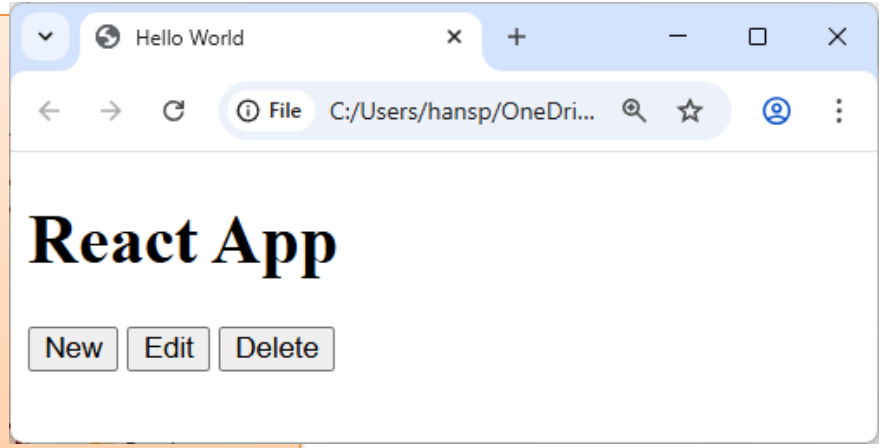
```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World</title>
    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>
    <h1>React Components</h1>
    <div id="root"></div>
    <script type="text/babel">

      function Button() {
        return <button>Click Me</button>;
      }

      const container = document.getElementById('root');
      const root = ReactDOM.createRoot(container);
      root.render(<Button />);

    </script>
  </body>
</html>
```



Return Multiple Elements

To return multiple elements you wrap it inside a ():

```
function Buttons() {
    return (
        <div>
         <button>New</button> 
         <button>Edit</button> 
         <button>Delete</button>
        </div>
    )
}

const container = document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(<Buttons />);
```



React **Props**

- Props (or Properties) are arguments passed into React Components.
- Props are passed to Components as HTML Attributes.
- Props are like function arguments in JavaScript and attributes in HTML.

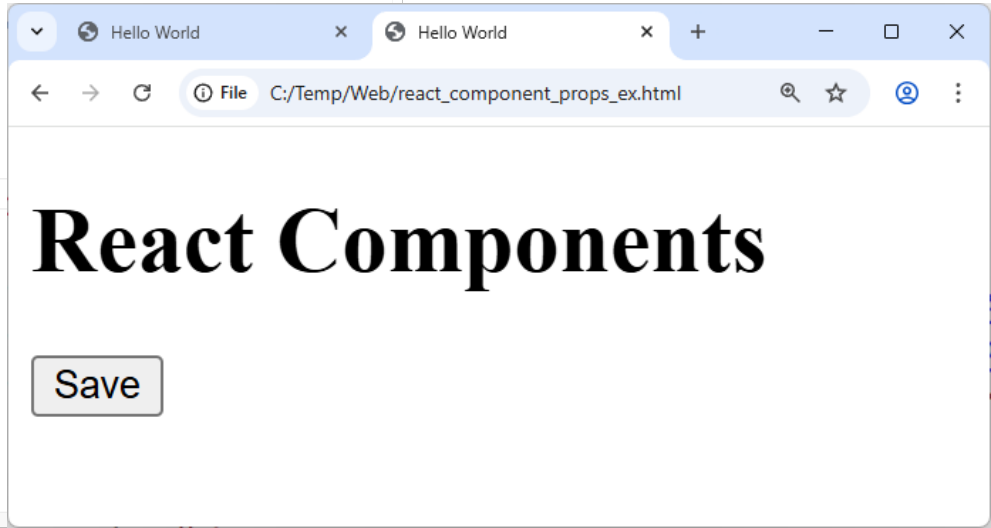
Props Example

For props you use curly brackets {}:

```
function Button(props) {
    return <button>{props.name}</button>;
}

const container =
document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(<Button name="Save" />);
```

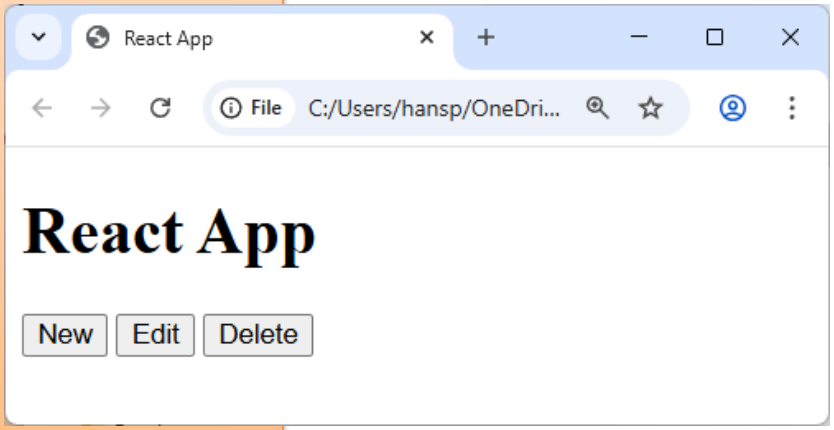
Props Example

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello World</title>
    <script src="https://unpkg.com/react@18/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  </head>
  <body>
    <h1>React Components</h1>
    <div id="root"></div>
    <script type="text/babel">

      function Button(props) {
        return <button>{props.name}</button>;
      }

      const container = document.getElementById('root');
      const root = ReactDOM.createRoot(container);
      root.render(<Button name="Save" />);

    </script>
  </body>
</html>
```



Props Example 2

```
function Button(props) {
    return (
        <button>{props.name}</button>
    )
}

const container = document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(
     <div>
     <Button name="New" />
      
     <Button name="Edit" />
      
     <Button name="Delete" />
     </div>
);
```



Getting Started with React

React Events

Hans-Petter Halvorsen

Table of Contents

Events

- React lets you add event handlers to your Components.
- Event handlers are your own functions that will be triggered in response to user interactions like clicking, hovering, focusing on form inputs, and so on.
- React Events vs HTML Events:
  - Use camelCase:
    onClick instead of onclick.
  - Use curly brackets {}:
    onClick={showMessage} instead of onclick="showMessage()"

Event Example

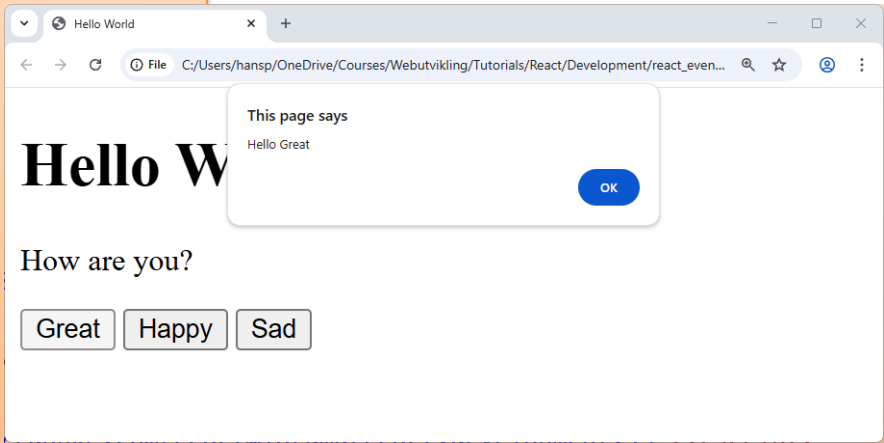
```
function Button(props) {
   function showMessage(){
      alert(`Hello ${props.name}`);
   }
   return (
      <button onClick={showMessage}>{props.name}</button>
   )
}

const container = document.getElementById('root');
const root = ReactDOM.createRoot(container);
root.render(
    <div>
    <Button name="Great" />
     
    <Button name="Happy" />
     
    <Button name="Sad" />
    </div>
);
```

Note the following:

```
alert(`Hello ${props.name}`);
```

You can also write like this:

```
alert("Hello " + props.name);
```



Handling Strings in JavaScript

- In JavaScript, you can choose **single quotes (')**, **double quotes (")**, or **backticks (`)** to wrap your strings in.
- Strings declared using single quotes (') and strings declared using double quotes (") are the same, and which you use is down to personal preference - although it is good practice to choose one style and use it consistently in your code.
- Strings declared using backticks (`) are a special kind of string called a template literal. In most ways, template literals are like normal strings, but they have some special properties.
- Inside a template literal, you can wrap JavaScript variables or expressions inside ${ }, and the result will be included in the string.

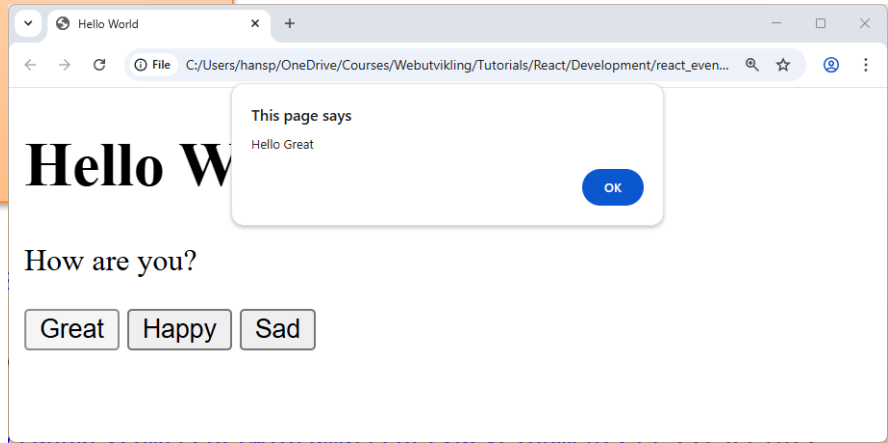
```
const name = "Donald";
const greeting = `Hello, ${name}`;
console.log(greeting); // "Hello, Donald"
```

https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/Strings

Event Example Alt. Solution

```
function Button(props) {
    const showMessage = () => {
        alert(`Hello ${props.name}`);
    }
    return (
        <button
onClick={showMessage}>{props.name}</button>
    )
}

..
```

You can also use the arrow => function.



https://www.halvorsen.blog

Getting Started with React

Installing React Environment

Hans-Petter Halvorsen

Table of Contents

Install React

- To start learning React you can write React directly in your HTML files as shown in the example.
- You can create a React App using Node Package Manager (NPM):

```
npx create-react-app myreactapp
```

- NPM is part of the Node.js installation.
- You can use a build tool like **Vite**.
- Here you find more information about installation and creating React Apps https://react.dev/learn/installation
- How to Install React – A Step-by-Step Guide: https://www.freecodecamp.org/news/how-to-install-react-a-step-by-step-guide/
- You can also use Visual Studio to create a React App.

Getting Started with React

Node.js and NPM

Hans-Petter Halvorsen

Table of Contents

Node.js

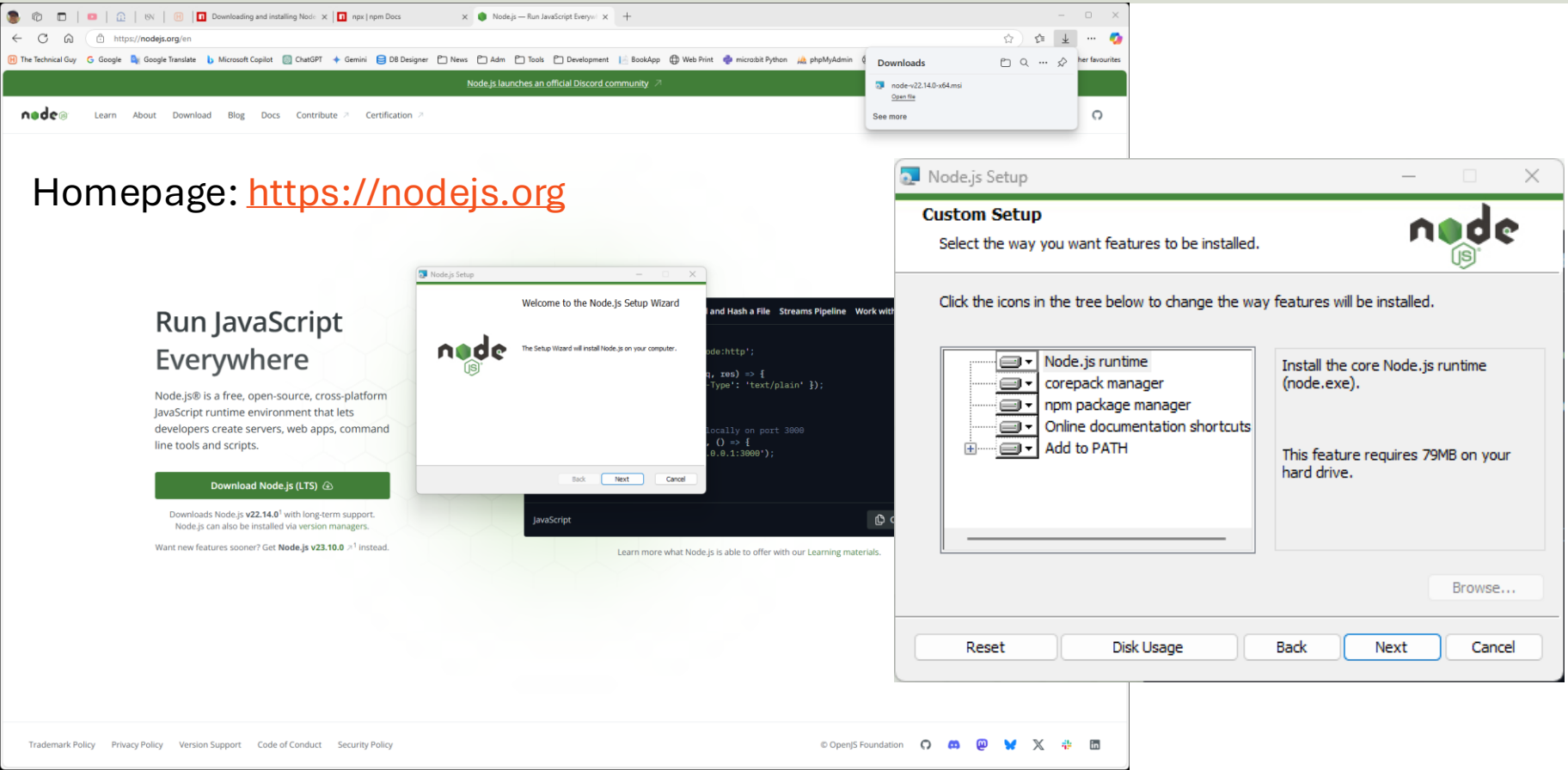
- Node.js allows you to run JavaScript on the server (server-side, backend).
- Node.js is a free, open-source and cross-platform
- **npm** is a package manager for Node.js packages/modules
- Homepage: https://nodejs.org
- npm is used to install many other frameworks/libraries like TypeScript, React, etc.

Node.js Resources

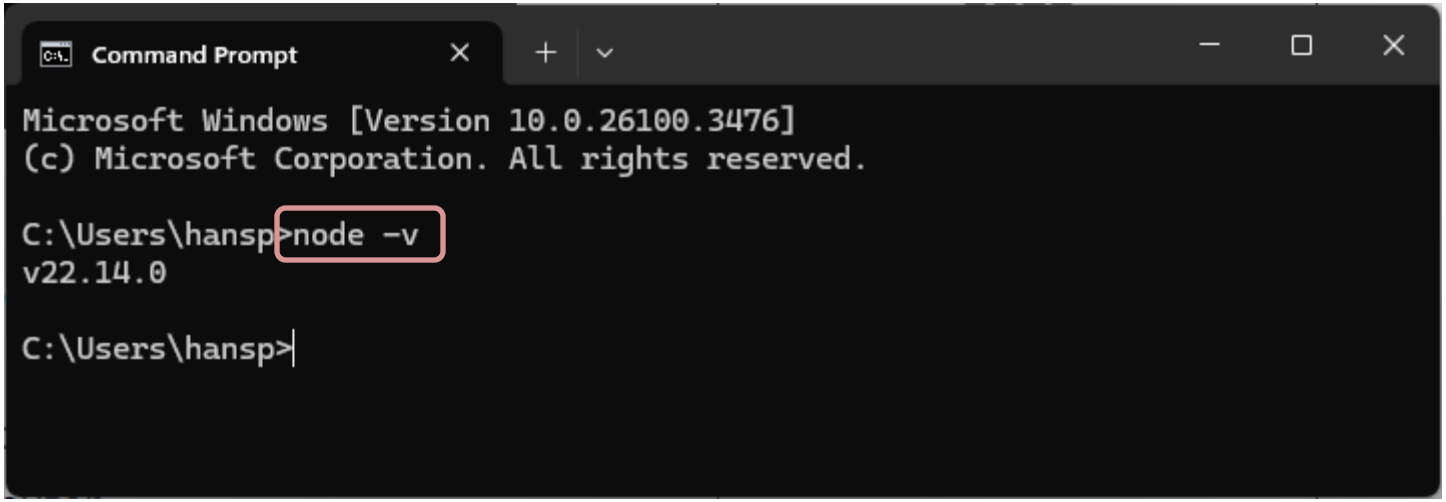
- Node.js Homepage: https://nodejs.org
- Node.js Tutorial w3school: https://www.w3schools.com/nodejs
- What Exactly is Node.js? Explained for Beginners: https://www.freecodecamp.org/news/what-is-node-js/

Install Node.js

Homepage: https://nodejs.org



Node.js



After installation of Node.js you can open the Command Prompt to confirm that Node has been successfully installed. If you enter “node -v in”, the installed version of Node.js will be shown.

NMP and NPX

- **NPM** is a package manager, and it is installed as part of Node.js.
- NPM is a package manager used to install, delete, and update JavaScript packages on your machine.
- NPM is short for Node Package Manager.
- NPX is a package executer, and it is used to execute JavaScript packages directly, without installing them.
- NPX is short for Node Package eXecute.

NPM

- npm is the world's largest Software Library.
  - Homepage: https://www.npmjs.com
- npm is also a Software Package Manager and Installer.
- npm is free to use.
- npm includes a CLI (Command Line Client) that can be used to download and install software:
  - C:\>npm install <package>
- The name npm (Node Package Manager) comes from Node.js since it was originally created as a package manager for Node.js.
- For most of the JavaScript packages you need to use npm to install them.
- npm is installed with Node.js
  - This means that you must install Node.js to get npm installed on your computer.
  - Download: https://nodejs.org

Install React using CRA

- You can create a React App using Node Package Manager (NPM):

```
npx create-react-app myreactapp
```

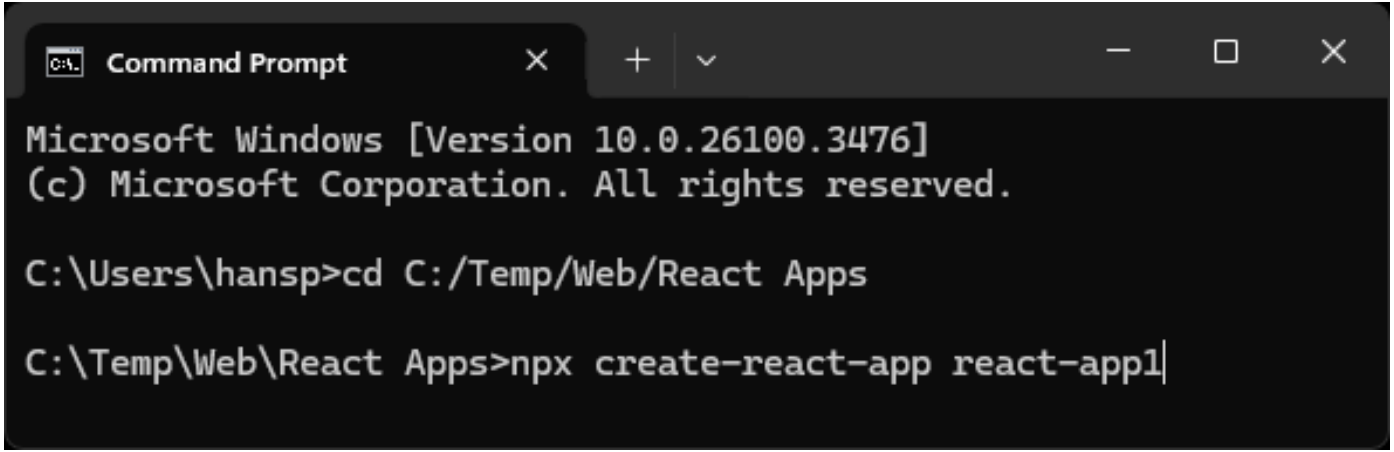
- “CRA” is basically short for the command “`create-react-app`”

“CRA” –> “Create React App”

Install React using CRA

Steps:

1. Create a Folder where you want to create the React App.
2. Open Command Prompt
3. Navigate to the directory that you want to use in creating your React App using cd …
4. Write `npx create-react-app myreactapp`



Command Prompt

```
  run `npm fund` for details
Git repo not initialized Error: Command failed: git --version
    at genericNodeError (node:internal/errors:983:15)
    at wrappedFn (node:internal/errors:537:14)
    at checkExecSyncError (node:child_process:882:11)
    at execSync (node:child_process:954:15)
    at tryGitInit (C:\Temp\Web\React Apps\react-app1\node_modules\react-scripts\scripts\init.js:46:5)
    at module.exports (C:\Temp\Web\React Apps\react-app1\node_modules\react-scripts\scripts\init.js:276:7)
    at [eval]:3:14
    at runScriptInThisContext (node:internal/vm:209:10)
    at node:internal/process/execution:449:12
    at [eval]-wrapper:6:24 {
  status: 1,
  signal: null,
  output: [ null, null, null ],
  pid: 30448,
  stdout: null,
  stderr: null
}

Installing template dependencies using npm...

added 18 packages, and changed 1 package in 6s

268 packages are looking for funding
  run `npm fund` for details
Removing template package using npm...


removed 1 package, and audited 1342 packages in 3s

268 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

Success! Created react-app1 at C:\Temp\Web\React Apps\react-app1
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd react-app1
  npm start

Happy hacking!

C:\Temp\Web\React Apps>
```

Command Prompt

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hansp>cd C:/Temp/Web/React Apps

C:\Temp\Web\React Apps>npx create-react-app react-app1
```

react-app1

Temp > Web > React Apps > react-app1 >

Search react-app1

New | Sort | View | Details

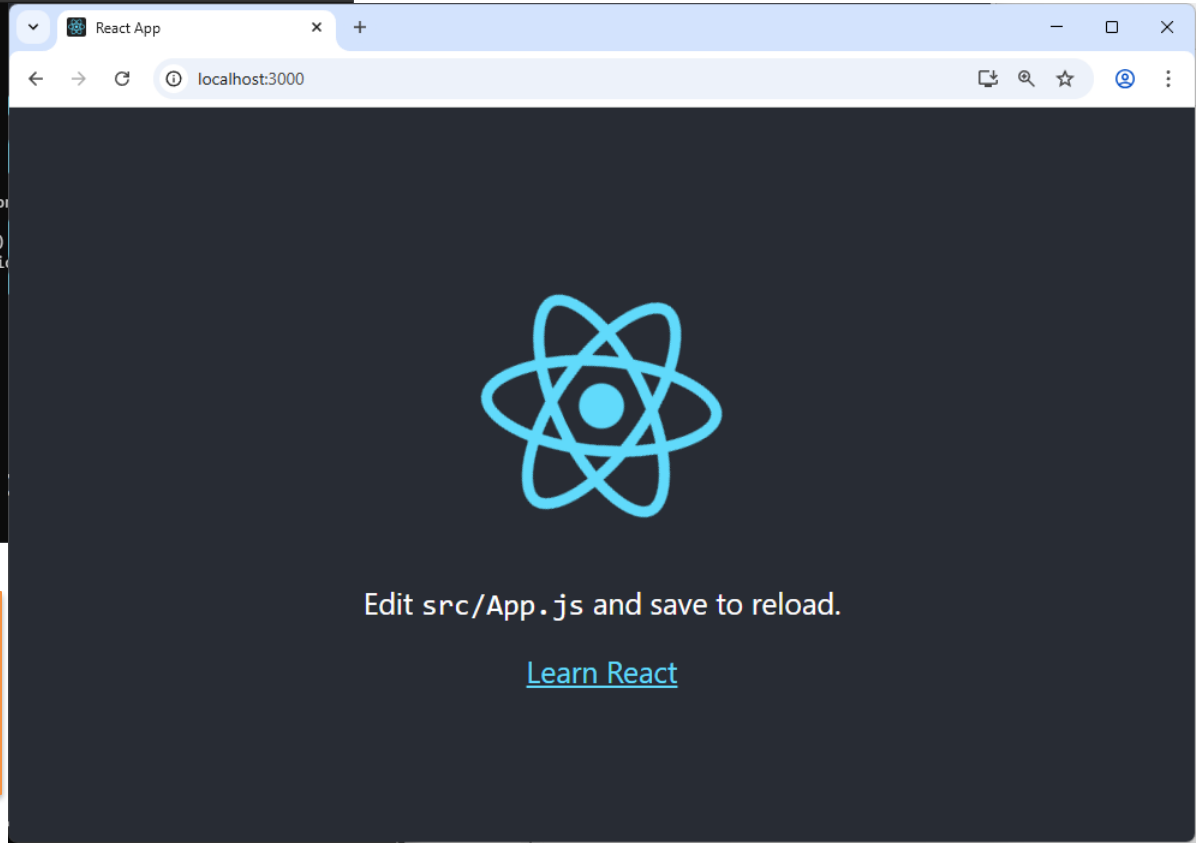
Home
Gallery
Hans-Petter - Personal
Desktop
Downloads
Pictures
Documents
Temp
Web

Name	Date modified
node_modules	02/04/2025 15:29
public	02/04/2025 15:29
src	02/04/2025 15:29
.gitignore	02/04/2025 15:28
package.json	02/04/2025 15:29
package-lock.json	02/04/2025 15:29
README.md	02/04/2025 15:28

7 items

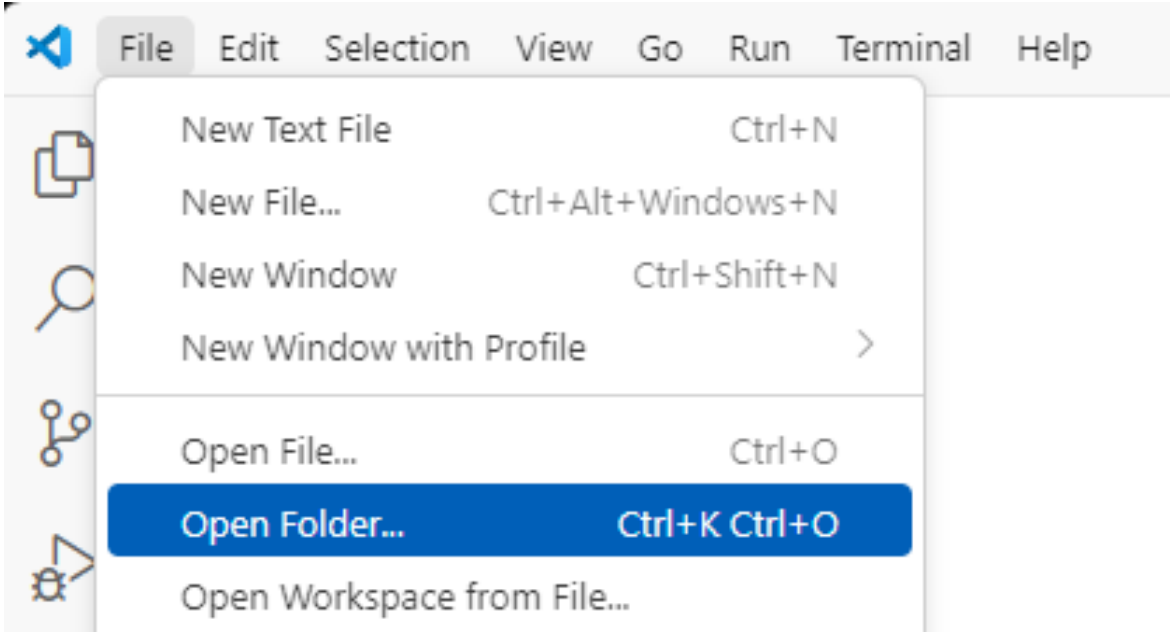
Start the React App

```
Windows PowerShell
Happy hacking!

C:\Temp\Web\React Apps>cd react-app1

C:\Temp\Web\React Apps\react-app1>npm start

> react-app1@0.1.0 start
> react-scripts start

(node:16064) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] Deprecatio
 option is deprecated. Please use the 'setupMiddlewares' option.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:16064) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] Deprecati
e' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view react-app1 in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.116.1:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```



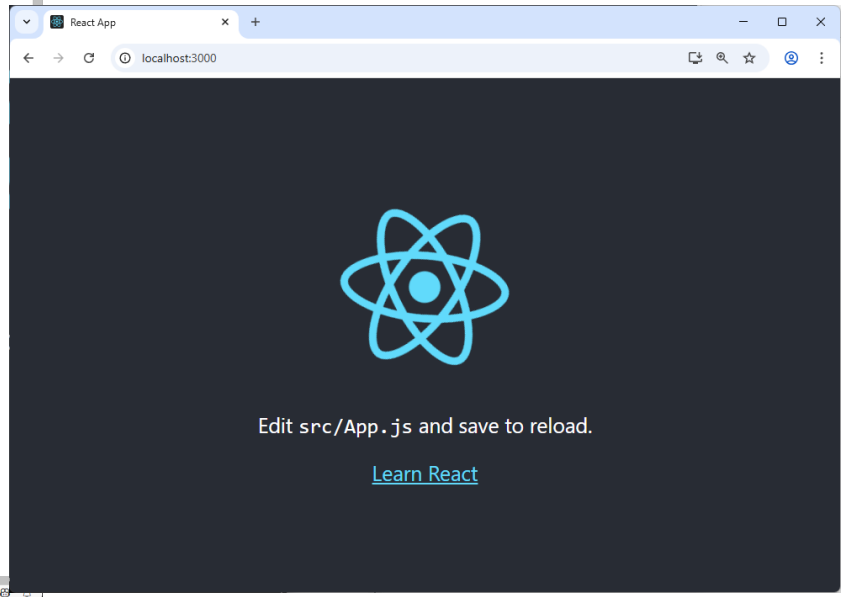

```
cd myapp
myapp>npm start
```

Open Code in Visual Studio Code



Open Code in Visual Studio Code

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```



Edit Code

File Edit Selection react-app1

EXPLORER

REACT-APP1
- 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

OUTLINE

TIMELINE

App.js

src > App.js > ...

```
1  import './App.css';
2
3  function App() {
4    return (
5      <div className="App">
6        <h1>Hello World</h1>
7      </div>
8    );
9  }
10
11 export default App;
```

0 0 Ln 2, Col 1 Spaces: 2 UTF-8 LF {} Jav

React App

localhost:3000

Hello World

Note!

- “create-react-app” has become deprecated and the React team doesn't recommend using it anymore.
- Other modern tools provide a faster front-end development experience, and you should use them to create React Apps.
- Such tools include Vite, Snowpack, Gatsby, Next.js, Percel, etc.

https://www.freecodecamp.org/news/how-to-install-react-a-step-by-step-guide/

https://www.halvorsen.blog

Getting Started with React

React + Vite

Hans-Petter Halvorsen

Table of Contents

Vite

- Vite is a frontend build tool and a local development webserver
- Just as we did when installing React using CRA, the first step is to make sure you have Node installed on your computer.
- Homepage: https://vite.dev
- How to Install React – A Step-by-Step Guide: https://www.freecodecamp.org/news/how-to-install-react-a-step-by-step-guide/

Create React App using Vite

```
> cd ..   (go to folder where you want to create the app)
> npm create vite@latest appname
```

C:\WINDOWS\system32\cmd.

```
C:\Temp\Web\React Apps>npm create vite@latest react-app2

> npx
> cva react-app2

|
*  Select a framework:
|     Vanilla
|     Vue
|   > React
|     Preact
|     Lit
|     Svelte
|     Solid
|     Qwik
|     Angular
|     Others
-
```

C:\WINDOWS\system32\cmd.

```
C:\Temp\Web\React Apps>npm create vite@latest react-app2

> npx
> cva react-app2

|
o  Select a framework:
|  React
|
*  Select a variant:
|     TypeScript
|     TypeScript + SWC
|   > JavaScript
|     JavaScript + SWC
|     React Router v7 ↗
-
```

Command Prompt

```
C:\Temp\Web\React Apps>npm create vite@latest react-app2

> npx
> cva react-app2

|
o  Select a framework:
|  React
|
o  Select a variant:
|  JavaScript
|
o  Scaffolding project in C:\Temp\Web\React Apps\react-app2...
|
-  Done. Now run:

   cd react-app2
   npm install
   npm run dev

C:\Temp\Web\React Apps>
```

Run the React App

C:\WINDOWS\system32\cmd.

```
cd react-app2
npm install
npm run dev


C:\Temp\Web\React Apps>cd react-app2

C:\Temp\Web\React Apps\react-app2>npm install

added 150 packages, and audited 151 packages in 12s

30 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\Temp\Web\React Apps\react-app2>npm run dev

> react-app2@0.0.0 dev
> vite


  VITE v6.2.5  ready in 274 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h + enter to show help
```

Vite + React

localhost:5173

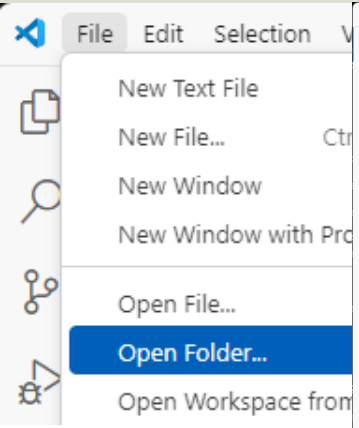
Vite + React

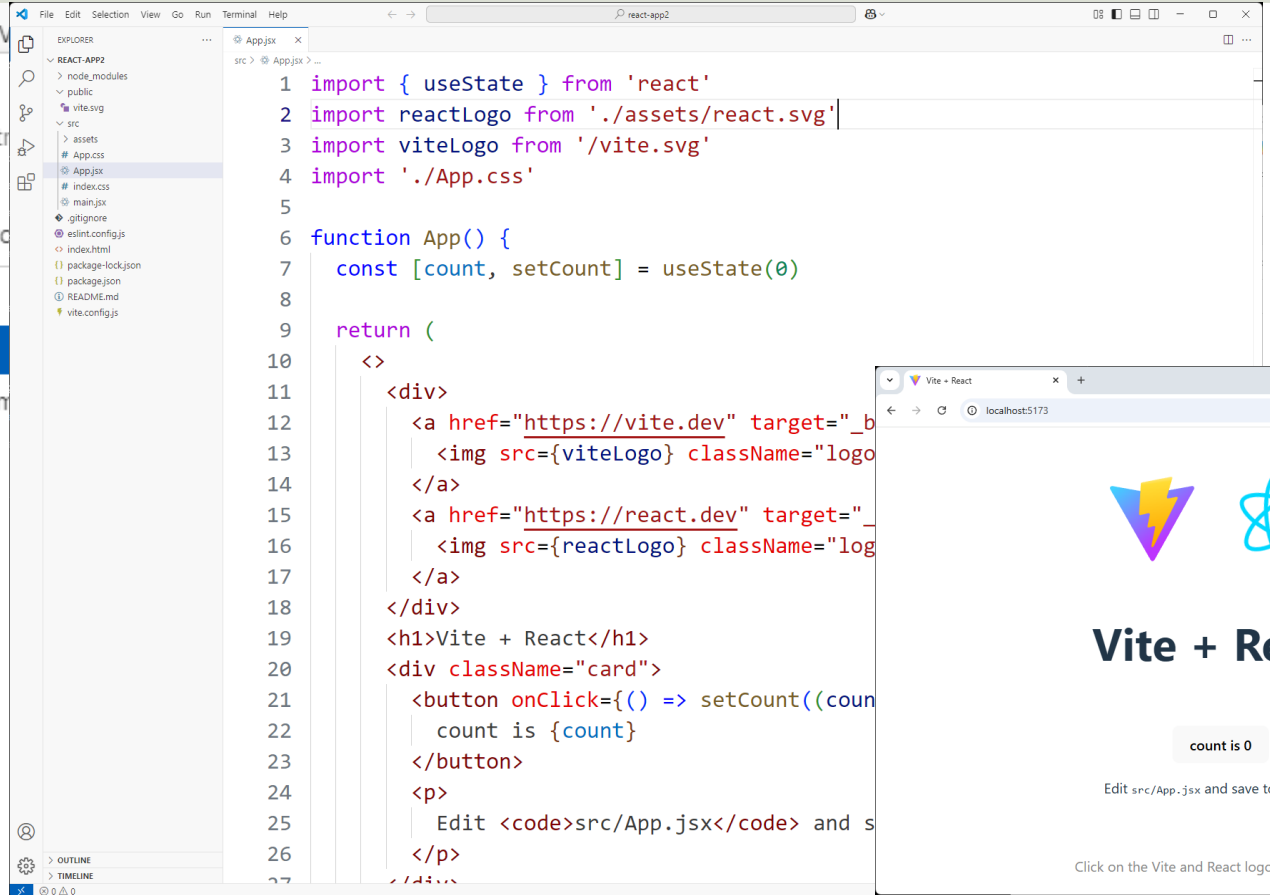
count is 0

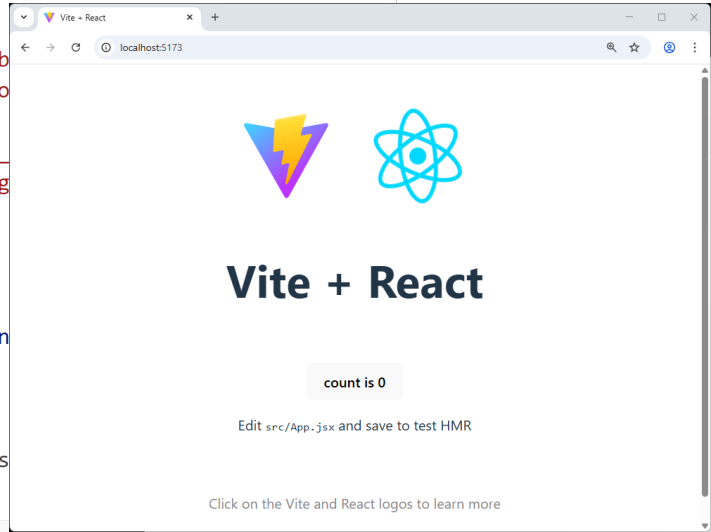
Edit `src/App.jsx` and save to test HMR

Click on the Vite and React logos to learn more

Open Code in Visual Studio Code

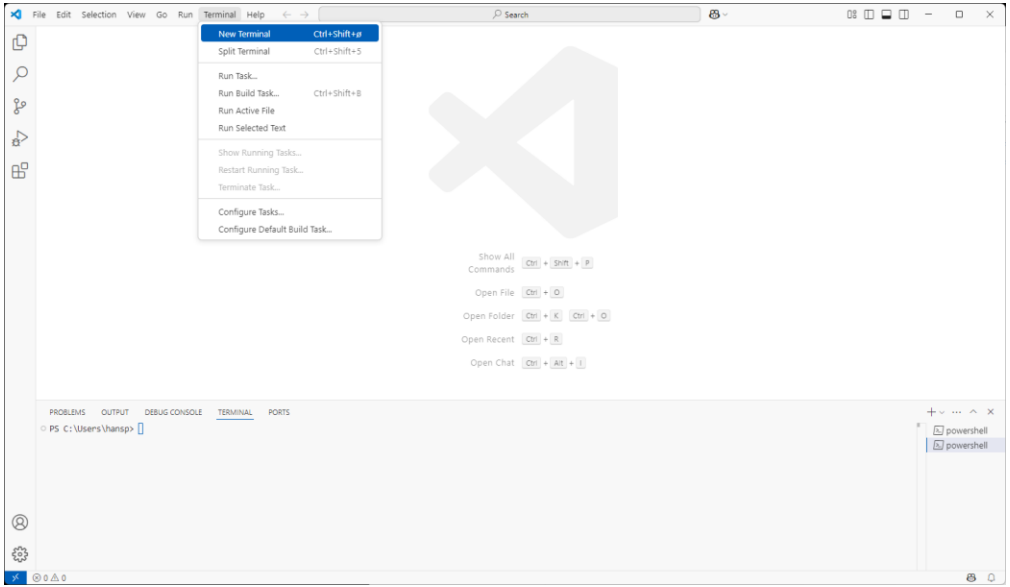


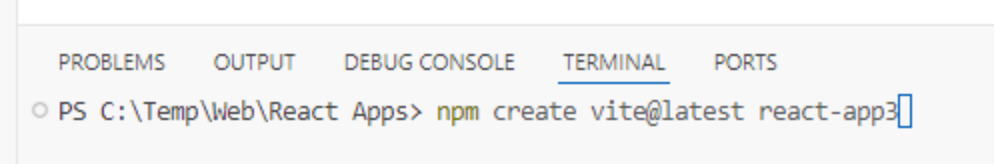




Terminal in Visual Studio Code

You can also use the terminal in Visual Studio Code:





Getting Error?

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Temp\Web\React Apps> npm create vite@latest react-app3
npm : File C:\Program Files\nodejs\npm.ps1 cannot be loaded. The file C:\Program Files\nodejs\npm.ps1 is not digitally signed. You cannot run this script on the current system. For more information about
running scripts and setting execution policy, see about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ npm create vite@latest react-app3
+ ~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Temp\Web\React Apps>
```

Solution:
https://stackoverflow.com/questions/78450758/npm-file-c-program-files-nodejs-npm-ps1-cannot-be-loaded-because-running-scri

First check your status with the command
Get-ExecutionPolicy
To check whether it's restricted or others, it should Restricted if you face this error
Then run **Set-ExecutionPolicy -Scope CurrentUser**
Type
RemoteSigned

Getting Started with React

React App

Hans-Petter Halvorsen

Table of Contents

Helo World App

Let's delete all unnecessary code and create a simple “Hello World” App

```
import './App.css'

function App() {

  return (
    <>
      <h1>Hello World</h1>
      <p>How are you?</p>
    </>
  )
}

export default App
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
11:12:07 [vite] (client) hmr update /src/App.jsx
```

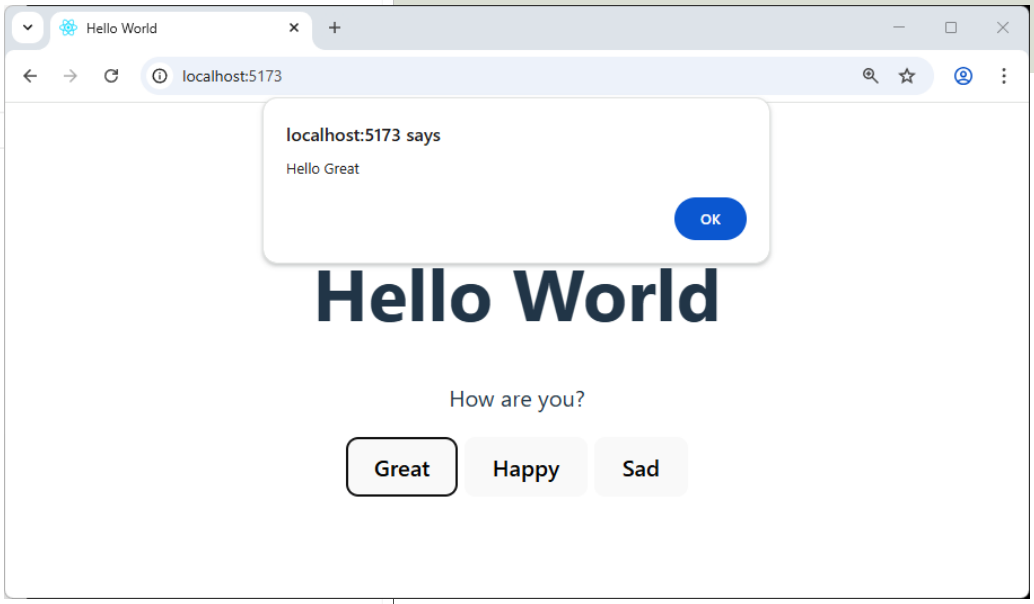
Hello World

localhost:5173

Hello World

How are you?

```jsx
import './App.css'

function App() {
  return (
    <>
      <h1>Hello World</h1>
      <p>How are you?</p>
      <Button name="Great" />
       
      <Button name="Happy" />
       
      <Button name="Sad" />
    </>
  )
}

function Button(props) {
  function showMessage(){
    alert(`Hello ${props.name}`);
  }
  return (
    <button onClick={showMessage}>{props.name}</button>
  )
}

export default App
```



We use the code created in the previous examples where we created a Button Component with Props and Events.

Put Component in separate File

```
import './App.css'
import Button from './Button.jsx'

function App() {
  return (
    <>
      <h1>Hello World</h1>
      <p>How are you?</p>
      <Button name="Great" />
       
      <Button name="Happy" />
       
      <Button name="Sad" />
    </>
  )
}

export default App
```

We create a new file called "Button.jsx"
And put the Button Component into that new file.

```
function Button(props) {
    function showMessage(){
      alert(`Hello ${props.name}`);
    }
    return (
      <button onClick={showMessage}>{props.name}</button>
    )
  }

  export default Button
```

https://www.halvorsen.blog

React in Visual Studio

Hans-Petter Halvorsen

Table of Contents

Node.js in Visual Studio

Modifying — Visual Studio Community 2022 — 17.10.0

Workloads | Individual components | Language packs | Installation locations

Web & Cloud (4)

- ASP.NET and web development — Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...
- Azure development — Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....
- Python development — Editing, debugging, interactive development and source control for Python.
- Node.js development — Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Desktop & Mobile (5)

- .NET Multi-platform App UI development — Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.
- .NET desktop development — Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
- Desktop development with C++ — Build modern C++ apps for Windows using tools of your
- Windows application development — Build applications for the Windows Platform using WinUI

Installation details

- Data science and analytical applicati...
- Node.js development
  - Included
    - Node.js development tools
    - JavaScript and TypeScript language support
    - JavaScript diagnostics
  - Optional
    - Web Deploy
    - Live Share
    - IntelliCode
    - GitHub Copilot
    - Connectivity and publishing tools
    - Developer Analytics tools
    - C++ core features
    - MSVC v143 - VS 2022 C++ x64/x86 build t...
- Individual components
- Extensions

Location
C:\Program Files\Microsoft Visual Studio\2022\Community

Remove out-of-support components

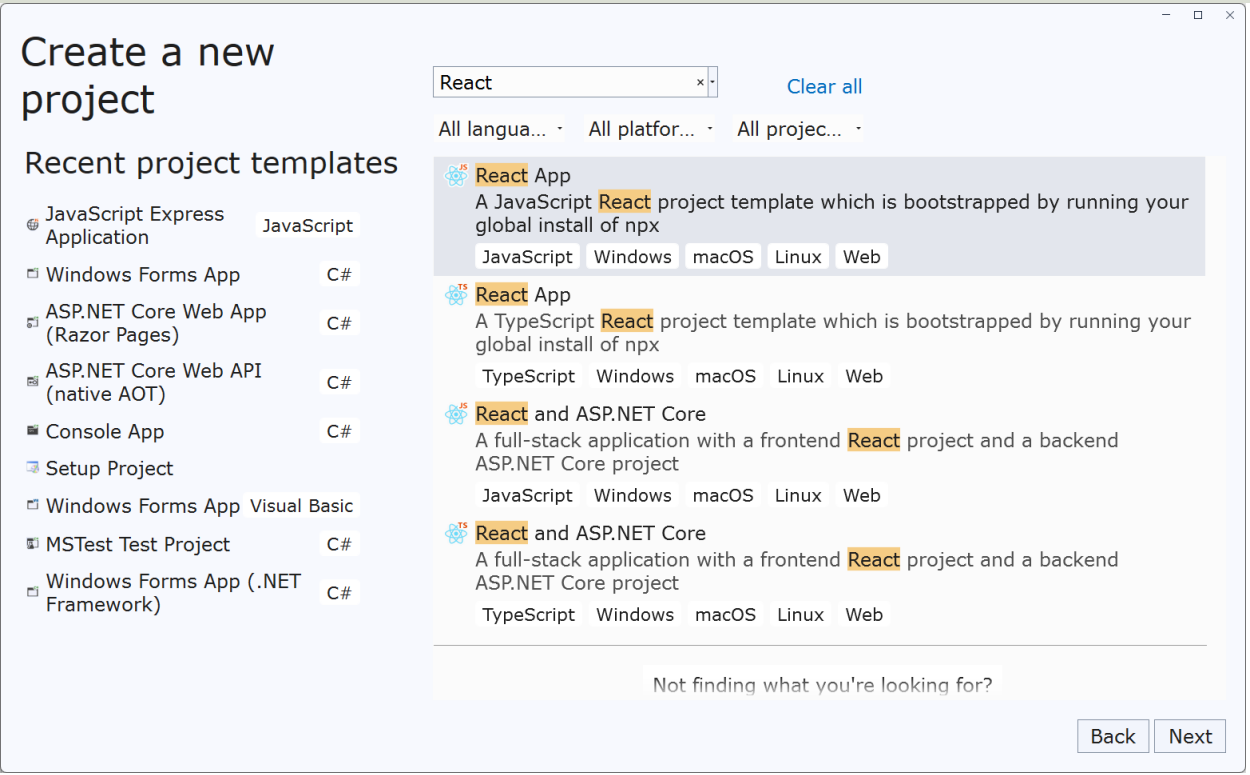
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 45 MB

Install while downloading | Modify

https://visualstudio.microsoft.com/vs/features/node-js/

React in Visual Studio



Different Templates:

- "React App"
- "React and ASP.NET Core" - Here, the ASP.NET Core project acts as an API backend and the React project acts as the UI.

You can choose between JavaScript or TypeScript

https://learn.microsoft.com/en-us/visualstudio/javascript/create-react-app

https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-asp-net-core-with-react

React App in Visual Studio

React App

A JavaScript React project template which is bootstrapped by running your global install of npx

JavaScript | Windows | macOS | Linux | Web

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help | Search | reactapp

Deb | Any CPU | localhost (Edge) | GitHub Copilot

main.jsx | index.html

Miscellaneous | "main" | StrictMode

```
1  import { StrictMode } from 'react'
2  import { createRoot } from 'react-dom/client'
3  import './index.css'
4  import App from './App.jsx'
5
6  createRoot(document.getElementById('root')).render(
7    <StrictMode>
8      <App />
9    </StrictMode>,
10 )
11
```

No issues found

Solution Explorer

Search Solution Explorer (Ctrl+¨)

- Solution 'reactapp' (1 of 1 project)
  - reactapp
    - Dependencies
    - .vscode
    - public
    - src
      - assets
      - App.css
      - App.jsx
      - index.css
      - main.jsx
    - .gitignore
    - CHANGELOG.md
    - eslint.config.js
    - index.html
    - package.json
    - README.md
    - vite.config.js

GitHub Copilot Chat | Solution Explorer

Properties

Output

Show output from: Debug

```
C:\Users\hansp\AppData\Local\Temp\visualstudio-js-debugger.txt
[vite] connecting...
[vite] connected.
-[0m-[1mDownload the React DevTools for a better development experience: https://react.dev/link/react-devtools-[0m
The thread 0 has exited with code 0 (0x0).
The program 'localhost:53397 - JavaScript debug adapter' has exited with code 4294967295 (0xffffffff).
The program '' has exited with code 4294967295 (0xffffffff).
```

Developer PowerShell | Output

Ready | Add to Source Control | Select Repository

Vite + React

localhost:53397

Vite + React

count is 0

Edit `src/App.jsx` and save to test HMR

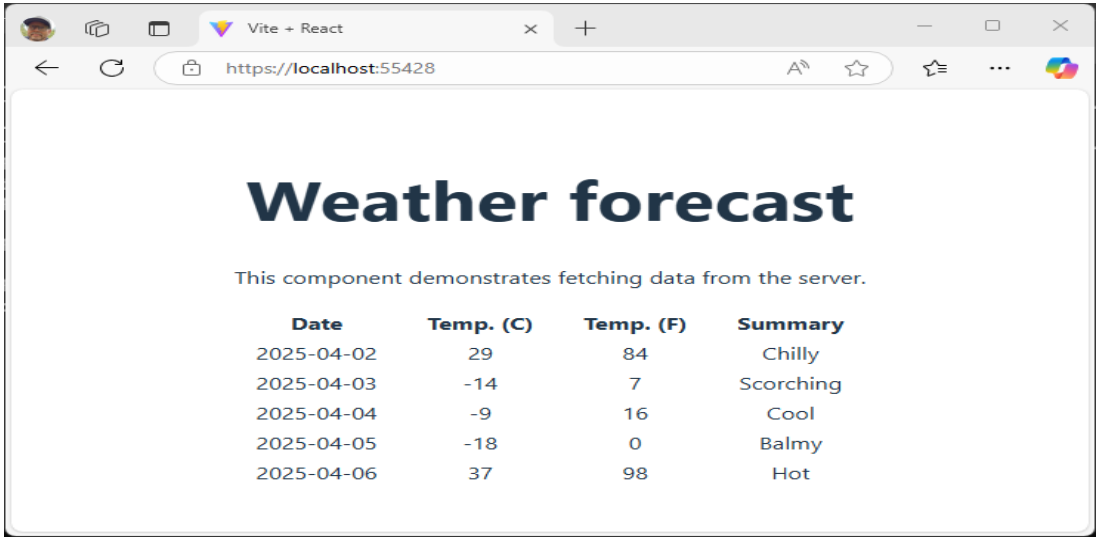
Click on the Vite and React logos to learn more

React and ASP.NET Core

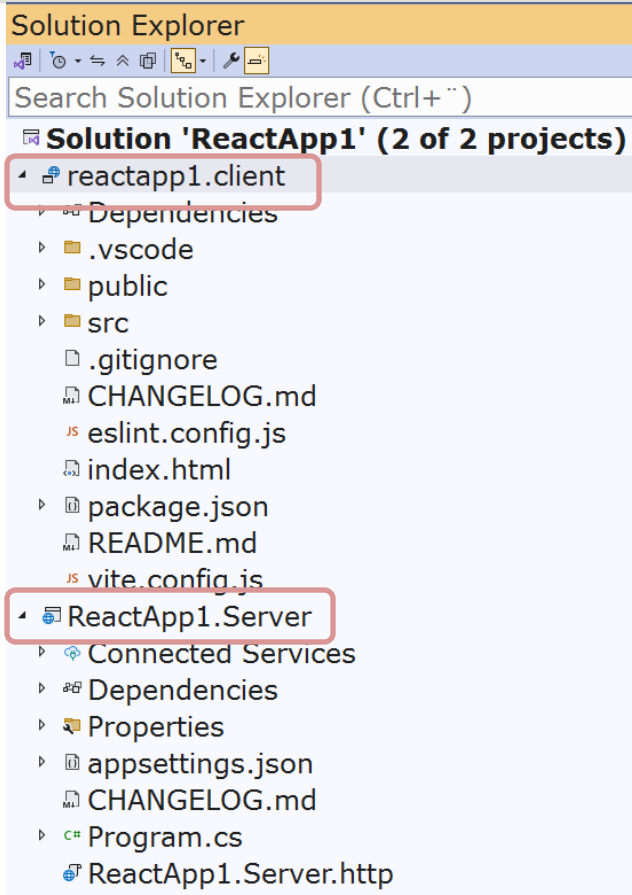
React and ASP.NET Core

A full-stack application with a frontend React project and a backend ASP.NET Core project

JavaScript Windows macOS Linux Web



Date	Temp. (C)	Temp. (F)	Summary
2025-04-02	29	84	Chilly
2025-04-03	-14	7	Scorching
2025-04-04	-9	16	Cool
2025-04-05	-18	0	Balmy
2025-04-06	37	98	Hot



Here, the ASP.NET Core project acts as an API backend and the React project acts as the UI.

React Visual Studio Resources

- Create a React app in Visual Studio: https://learn.microsoft.com/en-us/visualstudio/javascript/create-react-app
- Tutorial: Create an ASP.NET Core app with React in Visual Studio: https://learn.microsoft.com/en-us/visualstudio/javascript/tutorial-asp-net-core-with-react
- Create a full stack application by using React and minimal API for ASP.NET Core: https://learn.microsoft.com/en-us/training/modules/build-web-api-minimal-spa/

Getting Started with React

Other Matters

Hans-Petter Halvorsen

Table of Contents

Use Components made by others

- To make all your React Components from scratch may be time consuming.
- There exists different communities where you can get React Components made by others.
- Some Examples are:
- Chakra UI: https://chakra-ui.com
- Material UI: https://mui.com/material-ui/

Resources and References

- React Homepage: https://react.dev
- React Tutorial: https://www.w3schools.com/react/
- React Quick Start: https://react.dev/learn.
- Getting started with React: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Frameworks_libraries/React_getting_started
- ReactJS Tutorial: https://www.tutorialspoint.com/reactjs

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: https://www.halvorsen.blog