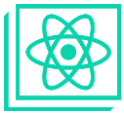
Lab6 –Create React App



The aim of this lab is to understand using the create-react-app command, share code in JS files, through import and export, and to structure components as modules.

1. What is Create-React-App (CRA)?

To run React in your browser, you need:

- Use `react.js` and `react-dom.js` from a CDN (Content Delivery Network).
- Use `Babel` to translate JSX code into JS at runtime.

But there's a better way, through CRA.
CRA is a scripting utility to facilitate the development of web applications based on React.

1.1 Installing Create-React-App:

CRA is written in Node and requires Node 6+ Node.js® is a JavaScript runtime environment built on Chrome's V8 JavaScript engine.
Node has a package ecosystem called NPM (Node Package Manager), which is the largest Open-Source JavaScript library. To write JS applications, we often use packages from this library,

> ***Note:*** *Before installing create-react-app, you must first install node and npm on your machine (installation details on Lab3). You can check the installed version using the following command in your terminal:*

```
node -v
```

You only need to install create-react-app once, from the command prompt:

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

Then, to create a react project, put it in a separate directory with a chosen name, e.g. `myrep`, as follows:

```
$ create-react-app myrep
```

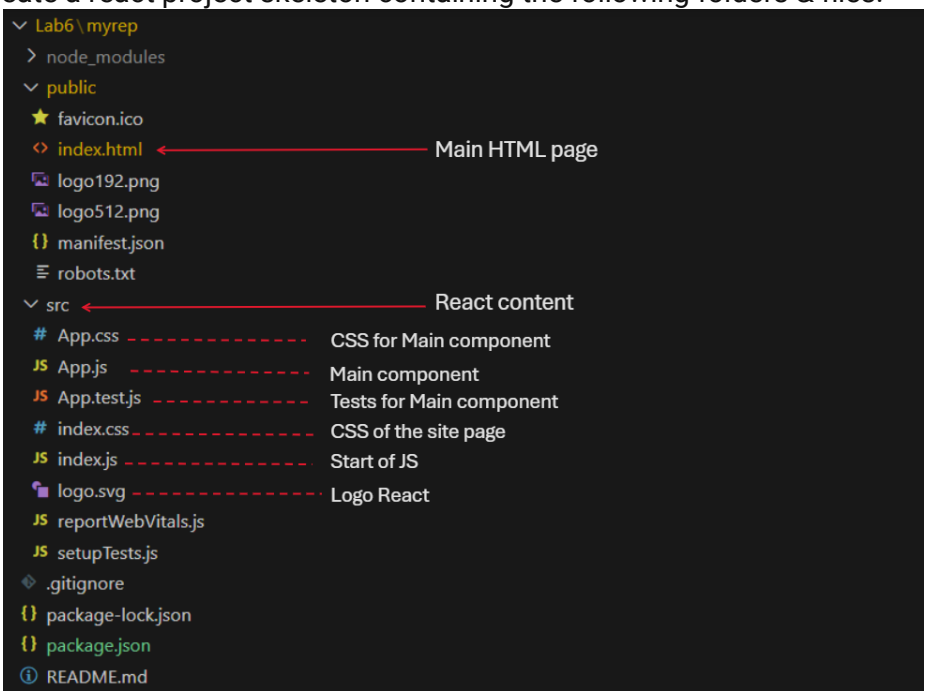
> ***Note:*** *If it returns an error message (UnauthorizedAccess)*

```
PS F:\NTIC_Local\2023-2024\S01\CAW\My Course\Labs\Lab6>  create-react-app myrep
create-react-app : File C:\Users\cheka\AppData\Roaming\npm\create-react-app.ps1 cannot be loaded because running scripts is disabled on this system. For more
information, see about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:2
+  create-react-app myrep
+  ~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
create-react-app : File C:\Users\cheka\AppData\Roaming\npm\create-react-app.ps1 cannot be loaded because running scripts is disabled on this system. For more
information, see about_Execution_Policies at https:/go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ create-react-app myrep
+ ~~~~~~~~~~~~~~~~
    + CategoryInfo          : SecurityError: (:) [], PSSecurityException
    + FullyQualifiedErrorId : UnauthorizedAccess
```

Run this command on Windows PowerShell (as an administrator):

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned
```

This will create a react project skeleton containing the following folders & files:



This content is created by default, but we can customize it to write our own components using very simple modifications (mainly acting on the content of the **`App.js`** file).

The most important files are: **`src/index.js`**, **`public/index.html`** and **`src/App.js`**.

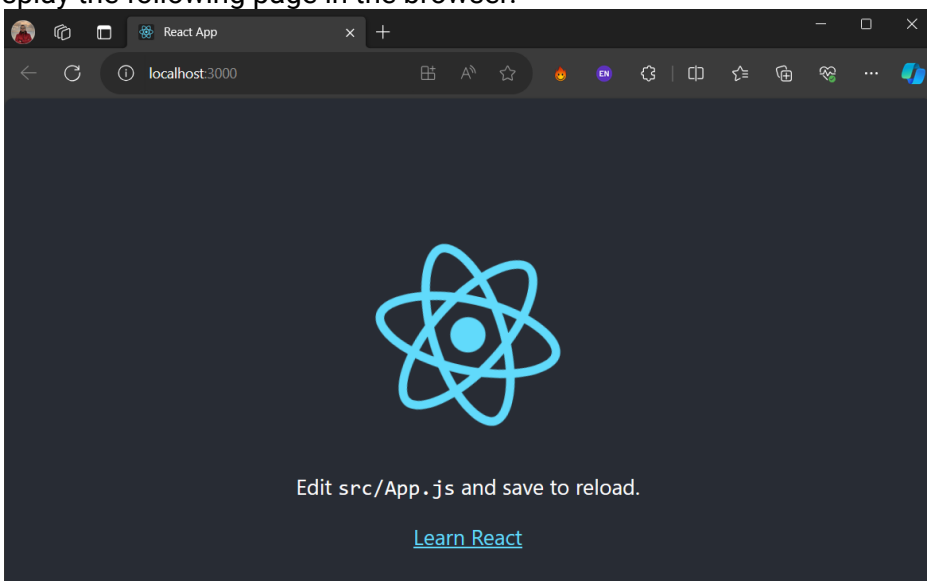
- If you want to change something in the html code, use **`public/index.html`**.
- If you want to change the main component to be displayed, use **`src/index.js`**.
- The **`src/App.js`** file is the component whose content will be displayed in **`index.js`**.

1.2 Launching the React application:

To launch the application, simply type the following command in the directory you've created (`cd myrep`):

```
$ npm start
```

This will display the following page in the browser:



Role of Webpack:
CRA is based on Webpack, a JS utility that lets you:

- Import/export modules:
  - Gather component-specific CSS code, images and JS files into a single file, making it easier for the browser.
  - Considerably reduce the number of HTTP requests, which improves performance.
- React Fast Refresh: The New React Hot Reloader, it is a feature introduced in React that allows you to get near-instant feedback for changes in your React components.
- Test & deploy the application more easily.

2. Module import/export

To share code between different components, you can utilize the import and export statements.

CRA uses ES2015 modules. Importing modules is a new standard version of invoking Node's **`require()`** method. It is similar to importing packages in Java.
You can import/export classes/data/functions between JS files.

All imported modules must be exported to the file containing them by:
`export [default] moduleName` : default is only used to export a main class or function within a module. It can be ignored. You don't use default if you want to export several things at once: variables, constants, functions, classes etc.

Example of a component:
`demo/my-app-name/src/ClickableButton.js`

```
import React, { useState } from 'react';
const ClickableButton = () => {
  const [isClicked, setIsClicked] = useState(false);

  const handleClick = () => {
    setIsClicked(true);
  };

  return (
    <div>
      <button onClick={handleClick}>ClickMe</button>
      {isClicked && <p>Clicked</p>}
    </div>
  );
};

export default ClickableButton;
```

2.1 Importing a module exported by default:

In general, a module exports its main function or component by default.
Example: `demo/import-export/mystuff.js`

```
function myFunc() {
  console.log("Hi");
 }

export default myFunc;
```

In the `demo/import-export/index.js` file where the `myFunc` function is imported:

```
import myFunc from './mystuff';
```

2.2 Importing an exported module: *(not by default)*

In general, when a module exports several elements (main function and other elements), it performs a "normal" export without necessarily being by default.
Example: in the `demo/import-export/mystuff.js` directory, we have a function `myFunc`

```
function myFunc() {
  console.log("Hi");
 }
 const msg = "wonderful" ;

 export default myFunc; // Default exportation
 export {msg} ; // Not by default exportation
```

in the `demo/import-export/index.js` file, which imports the `myFunc` function and the `msg` constant:

```
import myFunc, {msg} from './mystuff';
```

It is recommended to use a default export if there is a main function or component to export, otherwise use a "normal" export.

3. CRA and component conventions

For the best presentation of an application using CRA, the following guidelines should be observed:

- Each React component must be located in a separate file.
  - `src/Car.js` for the Car component, for example
  - `src/House.js` for the House component

- All components must extend the Component class (imputed from React).
  - Export any component as a default object.
- The application skeleton has App as its main component and must be located in the App.js file.

3.1. Convention for component CSS code:

CSS code for components must respect the following conventions:

- A CSS file for each React component, enabling it to be styled.
  - `House.css` for the House component
- Import this CSS file at the top of the `House.js` file.
  - CRA will automatically load this CSS file.
  - By convention, add the CSS class with the same name as the component in the first div of the corresponding component and use it as a prefix for the other classes of the elements contained in the component.

```
1   //Example: Componnt House
2  . . .
3  render (
4  <div className="House">
5   <p className="House-title">{ this.props.title }</p>
6   <p className="House-address">{ this.props.addr }</p>
7  </div>
8  )
9  export default House;
```

3.2 Convention for component images:

For component images, the following must be observed:

- place images in the `scr/` folder with components.
- If necessary, load them via Import, indicating the path and extension (.png, .gif etc).

```
1  import dol from "./dolly.jpg";
2
3  class Animal extends React.Component {
4   render() {
5   return (
6   <div>
7   <img src={dol} alt="my dolly!" />
8   </div>
9   );
10  }}
```

Classroom code

Google Classroom

aoa5lne

Drive QRcode

SCAN ME!