

Hands-on Lab: Hello World with React



Estimated Time: 20 minutes

In this lab, you will be creating your first React page and your first React application. As the first step to learning anything new in the coding world, you will render a 'Hello World' page.

Learning Objectives

After completing this exercise, you should be able to perform the following tasks:

- Create a simple React page
- Create, run, and render a React application
- Set properties for a React component

First React application

1. From the menu on top of the lab, go to Terminal and open a new terminal.
2. In the terminal, paste and run the following command to go to the `home/project` directory.

```
cd /home/project
```

3. Run the following command to create a React application named **myfirstapp**. It is mandated that the React application name cannot contain any uppercase letter.

```
npx create-react-app myfirstapp
```

4. It requires you to confirm that the packages required can be installed. Confirm by pressing `y` on your keyboard to get started with the installation.

```
theia@theia-lavanyas:/home/project$ npx create-react-app myfirstapp
Need to install the following packages:
  create-react-app@5.0.1
Ok to proceed? (y) █
```

Note: If you encounter an error related to the `react 19.0.0` version, please update the `react` and `react-dom` versions in the `package.json` and `package-lock.json` files as shown below.

```
"react": "18.0.0",
"react-dom": "18.0.0",
```

5. After the installation is over, the application is created inside a folder with the name specified by you for the application. Change to that folder by running the following command.

```
cd myfirstapp
```

6. Install the web-vitals package by running the following command:

```
npm install web-vitals
```

7. create-react-app creates the boiler plate code which is required for you to run a basic React app. To check the React app that has been created, run the following command.

```
npm start
```

This will start the React development server on port 3000.

8. Click Launch Application below or the SkillsNetwork Toolbox logo and from the options select OTHER and click Launch Application. Enter the port number 3000 and click the arrow mark to launch the application.

Launch Application

The screenshot displays the Theia IDE interface. At the top is a menu bar with 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. Below the menu is a sidebar with icons for file management, search, and other tools. The main area is titled 'Launch Your Application' and contains a dialog for launching an application. The dialog has a title bar with 'expWithAuth.js' and 'Your Application'. It includes a section for 'Application Port' where '3000' is entered. A 'Launch' button is visible. Below the dialog is a terminal window showing the command 'node expWithAuth.js' and the output 'Server is running'.

File Edit Selection View Go Run Terminal Help

S... ? expWithAuth.js Your Application Launch Application

> DATABASES

> BIG DATA

> CLOUD

OTHER

Launch ...

Open Li...

Open C...

Open C...

Sugges...

About ...

Launch Your Application

To open any application in the browser, please select

Application Port

3000

Your Application

Problems theia@theia-lavanyas: /home/project × t

```
theia@theia-lavanyas:/home/project$ node expWithAuth.js
express-session deprecated undefined resave option
express-session deprecated undefined saveUninitialized
expWithAuth.js:35:9
Server is running
```

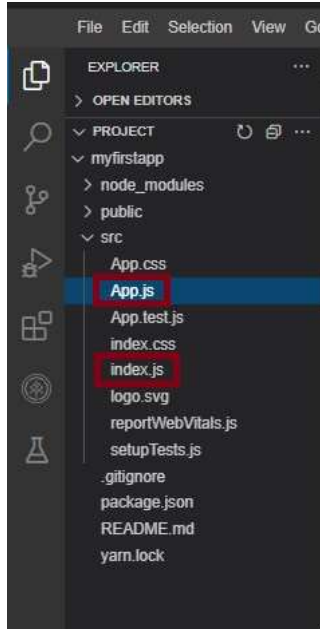
9. This will launch the application in a new browser tab.



Press Ctrl+C at the command prompt to stop the React development server.

Modify the Hello World App

1. Within the src folder of the myfirstapp project folder, you will find **App.js** and **index.js**.



2. Replace the following code in index.js inside the src folder. All the styling will be removed as a result, leaving you with the basic needs.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <App />
);
```

3. Replace the content of App.js inside the folder src, with the following code:

```
function App() {  
  return (  
    <h1>Hello World</h1>  
  );  
}  
export default App;
```

4. Remove all the files which you do not require for this lab, by running the following command:

```
rm src/App.css src/App.test.js src/index.css src/logo.svg src/reportWebVitals.js src/setupTests.js
```

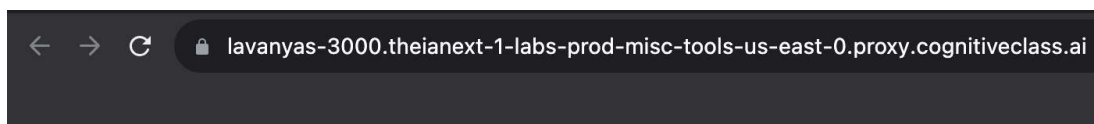
5. Now start the server again and follow the steps mentioned earlier to launch and view the application on the browser.

```
npm start
```

6. Click Launch Application below.

Launch Application

You will see the page render with the message "Hello World".



Hello World

Hello World with Time

1. Change the code in App.js, by pasting the code shown below. It will render the timestamp along with the Hello World message in the browser page.

```
function App(props) {  
  const currDate = new Date();  
  return (  
    <div>  
      <h1>Hello, world!</h1>  
      <h2>The time now is {currDate.toLocaleTimeString()}</h2>  
    </div>  
  );  
}  
export default App;
```

2. Now you will see the time along with "Hello World". Refresh the page if you don't see the update.



Hello, world!

The time now is 12:52:13.

3. To ensure the time is kept current and is constantly changing, change the code in index.js by pasting the following code in it:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
const root = ReactDOM.createRoot(document.getElementById('root'));
function ref() {
  root.render(<App/>);
}
setInterval(ref, 1000);
```

4. Now start the server again, if it is not running and follow the steps mentioned earlier to launch and view the application on the browser.

```
npm start
```

Launch Application

Practice exercise:

1. Replace the "Hello World" message to display your name instead.
2. Replace the code to display the date along with the time.

▼ Click here for a hint

Use `currDate.toLocaleDateString()` to get the date string.

▼ Click here for the solution

```
function App(props) {
  const currDate = new Date();
  return (
    <div>
      <h1>Hello, world!</h1>
      <h2>It is {currDate.toLocaleDateString()} and the time now is {currDate.toLocaleTimeString()}</h2>
    </div>
  );
}
```

```
</div>  
);  
}  
export default App;
```

Author(s)

[Lavanya T S](#)

© IBM Corporation. All rights reserved.