



香港中文大學
The Chinese University of Hong Kong

CSCI2720 2023-24 Term 1: Building Web Applications

Lab 6: React Router

Dr Colin Tsang

Outline

- Tooling ecosystem
- Accessing command line
- Basic of React Router
- URL parameters
- No-match route
- Functional vs class components

Tooling ecosystem

- We don't want to work with bare HTML/CSS/JS
- More and more tools are there to help us for
 - Safety net
 - E.g., checking errors in code
 - Transformation
 - E.g., transpiling JSX to plain JS
 - Post-development
 - E.g., testing and deployment

Tooling ecosystem

- To ensure easy incorporation of multiple tools, the command line interface (CLI) is the best choice
 - Consistency with clear syntax
 - Easily scriptable for automation
- Many tools provide CLI
 - npm, React CLI, Netlify CLI, ...

Accessing CLI on your computer

- For simplicity, our lab will demo using npm by cloud services, but you can try it on your own computer too.
- On Linux and macOS, the **Terminal** is ready for use
 - Just search for it on your computer
- On Windows, the **Command Prompt (cmd)** is a bit primitive
 - Just run **cmd** from the start menu: it is okay to use **cmd** in this stage
 - More preferred: **Powershell**
 - Advanced user: **Windows Subsystem for Linux (WSL)**
 - See: <https://learn.microsoft.com/en-us/windows/wsl/install>
- There are other possibilities, e.g., virtual machines, cloud services
 - We will try **AWS** later.

Basic commands

- While Linux and macOS are similar, Windows have a different set of commands
- More details:
<https://www.geeksforgeeks.org/linux-vs-windows-commands/>

	Linux/macOS	Windows
List dir contents	<code>ls -l</code>	<code>dir</code>
Change dir	<code>cd dir</code>	<code>cd dir</code>
Make new dir	<code>mkdir dir</code>	<code>mkdir dir</code>
Copy file	<code>cp fileA fileB</code>	<code>copy fileA fileB</code>
Move file	<code>mv fileA fileB</code>	<code>move fileA fileB</code>
Delete file	<code>rm file</code>	<code>del file</code>
Delete dir	<code>rm -rf dir</code>	<code>rmdir dir</code>
Show file contents	<code>cat file</code>	<code>type file</code>

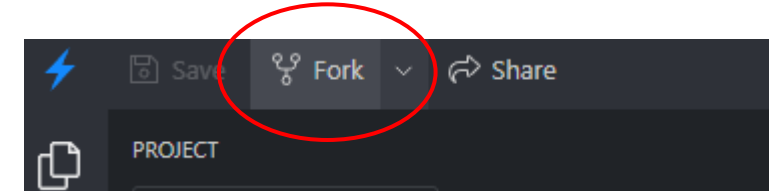
npm and npx

- npm is Node's package manager
 - It can be downloaded together with Node.js at <https://nodejs.org/>
- Other than the Node.js platform, it is also good for basic web development.
- More and more other tools are now provided as npm packages.
- npm allows easy management of packages
- npx lets you execute without installing
 - Newest version guaranteed.

Installing React router

- You can start with opening this link in Google Chrome:
 - <http://stackblitz.com/edit/node-inpmqg>
 - A react app is already created for you.
 - Fork the project to edit and save a copy. (you can do it with or without a StackBlitz account)
- In the terminal of StackBlitz, enter the following commands:

```
cd demo-app/ # enter app folder
npm install react-router-dom # install react-router-dom
npm start# start server
```
- After the command **npm start**, you should be able to see the web app rendered inside StackBlitz.



Basic components

- Look at the given file *src/index.js*
 - You will make your edits here
- The router
 - **<BrowserRouter>**
 - For modern browsers, supporting HTML5 history API with states, e.g., the *Back* button.
- Route matchers
 - **<Routes>** looks at children **<Route>** elements for the first match, and ignore others
 - **<Route>** matches URL against the **path="..."** *attribute/props*
- Route changers
 - **<Link>** allows specifying the *to* attribute

In a nutshell.....

- The given files has these features:
 - A list of links, and the linked component are displayed inside the component **App**
 - If the link *Home* is visited (URL becomes /), the **Home** component is shown
 - If the link *About* is visited (URL becomes /**about**), the **About** component is shown instead



Task 1: URL parameters

- A variable could be matched inside the URL
- Set up three more **<Link>**, pointing to */file/fileA*, */file/fileB*, and */file/fileC*
 - You can decide what labels they should take
- Under **<Routes>**, add more **<Route>**
`<Route path="/file/:id" element={<File />} />`
- Add this line in the top of the js file
`import { useParams , useLocation } from 'react-router-dom';`
- You cannot see the result yet, as **File** is not defined

Task 1: URL parameters

- Set up a new component File

```
function File() {  
  let { id } = useParams();  
  return (  
    <div>  
      <h3> ID: {id} </h3>  
    </div>  
  );  
}
```

- We try to use the *functional component* in this case. We will discuss it later.
- Now you can see the result, as **File** is well-defined.

- [Home](#)
- [About](#)
- [file A](#)
- [file B](#)
- [file C](#)

Home

Task 1: URL parameters

- In this task, we use the parameter **:id**. The string could be automatically captured for use with the **useParams()** hook
- This is especially useful for *pattern matching* in URL
- See: <https://reactrouter.com/en/main/start/concepts#matching>
- More on: <https://betterprogramming.pub/using-url-parameters-and-query-strings-with-react-router-fffdcea7a8e9>

Task 2: No-match route

- Traditionally, a web server would return status 404 with an error page to a URL not found on the server
- We can also do it here:
 - Add a wrong to URL with Link
 - Add a new Route at the end of the list

```
<Route path="*" element={<NoMatch />} />
```

- Set up a new NoMatch Component:

```
function NoMatch() {  
  let location = useLocation();  
  return (  
    <div>  
      <h3> No Match for <code>{location.pathname}</code></h3>  
    </div>  
  );  
}
```

- The `useLocation()` hook tell us what URL was brining to this page.

- [Home](#)
- [About](#)
- [file A](#)
- [file B](#)
- [file C](#)
- [This is a wrong link](#)

No Match for /wrongFile/wrongFile

Functional vs Class components

- Our original components were written in classes
- New components in the lab today are written in functions
- Which one is better?
 - *Class*: more traditional way to understand objects, clear use of *props* and *states*.
 - *Function*: cleaner code, shifting to the use of *hooks*.
- Either is fine, or even a mix of both, just like today lab
- Learn more about hooks of React Router:
 - See under “Hooks” on: <https://reactrouter.com/en/main/start/tutorial>
 -

Submission

- No submission is needed for labs
- What you have done could be useful for your further exploration or the upcoming assignment
- Please keep your own files safely