React Hooks 2

Tessema Mengistu (Ph.D.)
Department of Computer Science
Virginia Tech
Mengistu@vt.edu

React Hooks - Recap

- React Hooks
  - Special functions that allow developers to hook into state and life cycle of React components
  - JavaScript functions that accept arguments and return specific values
  - Should be imported before use
  - Example:
    - useState
    - useEffect
    - useReducer
    - useParam
    - . . .

React Hooks – useEffect()

- useEffect()
  - A react hook that performs side effects
    - Fetching data
    - Reading from local storage
    - . . .
  - Effects are always executed after rendering
    - Can be skipped
  - Takes two arguments:
    - The setup function
      - The function with the Effect’s logic
    - The dependency array
      - Optional

React Hooks – useEffect()

- **Syntax:**

```
useEffect(
    () => {
            //Execute side effect
        },
         [
                   // 0 or more entries
         ]
   )
```

React Hooks – useEffect()

```
 . . .
import { useEffect } from 'react';
...
function App() {
  useEffect(() => {
    axios.get('/api/items');
  }, []);
```

React Hooks – useEffect()

- The dependency array is an array of variables that React uses to determine whether it should execute the function supplied to useEffect
- React uses shallow comparisons
    - Reference equality

React Routers - Recap

- React Router
  - A routing library for React apps that gives navigational capability
  - Responsible for selecting what to show in the app for a requested path
  - Is part of a package called *`react-router-dom`*
    - Should be installed
  - Different kinds of routers
    - BrowserRouter
    - HashRouter
    - …

React Routers - Recap

- Routing allows users to interact with your app and enhances the overall user experience
- **Nested Routes**
  - Nesting routes is allowed in React

```
<Routes>
  <Route path="/" element={<Home />} />
  <Route path="/items" element={<ItemList />} >

     <Route path=":itemId" element={<ItemList />}/>
  <Route path="*" element={<Page Not Found />} />
</Routes>
```

React Routers

- Dynamic Segment - A segment of a path pattern that is dynamic, meaning it can match any values in the segment.
  - For example, the pattern `/items/:itemId` will match URLs like `/items/123`
- URL Params - The parsed values from the URL that matched a dynamic segment
  - A parameter has two parts:
    - The colon, which you use to identify a parameter to React Router
    - The parameter name (`itemId`, in this case), which follows the colon

React Hooks – useParams()

- Parameters are dynamic information that you can access inside your component
- React Router provides a hook called useParams that returns an object containing all parameters from the current URL
- useParams
  - Returns an object containing a key/value pair of the dynamic params from the current URL
    - The value passed as the key
  - Using object destructuring, you can immediately get the value of id

```
const { id } = useParams();
```

References

- http://react.dev/reference/react
- https://blog.logrocket.com/useeffect-hook-complete-guide/
- https://reactrouter.com/en/main/start/concepts