Exercise 5 – Component Lifecycle Methods

Objective

To understand when component lifecycle methods are called.

Overview

This exercise is designed to show you exactly when the differing component lifecycle methods are called at various points in an application. A simple counter application will be set up to display the count and a button to increase it. After it has been increased a pre-determined number of times, the component will be removed.

Part 1 – Project Setup

1.1. This project has already been set up for you, so there is no need to run through the project set-up.

Part 2 – The main.js file

2.1. In a suitable text editor, navigate to the **EG05_ComponentLifecycle/starter** folder and create a **main.js** file.

2.2. Add `imports` for `React`, `ReactDOM` and `App`.

2.3. Add the `ReactDOM.render` method that takes the arguments of an `App` component and the element with the `id` of `content`. The **App** component should have an attribute of `defaultProp` set to a string of `'Default Prop from main.js'`.

Part 3 – The App Component

3.1. Create the **App.jsx** file in the scripts folder and add the import for **React** and **ReactDOM**.

3.2. Add the `class` declaration, remembering its `export`.

3.3. Create a `constructor` for the class that:

a. Has `props` passed in

b. Calls `super`, passing in `props`

c. Sets **state** to have a `count` property set to **0**

d. Uses `console.log` statements to display

e. `'Constructor has been called'`

f. `'Initial Count is: ' + this.state.count`

3.4. Create a function called `increase` that will increase the value of `count` by **1** when called.

3.5. Create a function called `componentWillMount` that uses `console.log` to display `'componentWillMount: Component is about to mount'`.

3.6. Create a function called `componentDidMount` that uses `console.log` to display `'componentDidMount: Component just mounted'`.

3.7. Create a function called `componentWillUpdate`, taking `newProps` and `newState` as arguments, that uses `console.log` to display `'componentWillUpdate: Component is about to update'`.

3.8. Create a function called `componentDidUpdate`, taking `currentProps` and `currentState` as arguments, that uses `console.log` to display `'componentDidUpdate: Component just updated'`.

3.9. Create a function called `componentWillUnmount` that uses `console.log` to display `'componentWillUnmount: Component is about to be removed'`.

3.10. Create a function called `componentWillReceiveProps`, taking `newProps` as an argument, that uses `console.log` to display `'componentWillReceiveProps: Component will get new props! '`.

3.11. Create a function called `shouldComponentUpdate`, taking `newProps` and `newState` as arguments, that uses `console.log` to display `'shouldComponentUpdate?'`. Add a conditional statement to the function to:

a. See if the value of `newState.count` is less than **5**

b. If it is, use `console.log` to display `'Condition met: Component should update'` and `return true`.

i. If it isn't, use a `console.log` displaying `'Condition not met: Component should NOT update and has been removed'` and then add the following line of code:

```
ReactDOM.unmountComponentAtNode(content)
```

followed by `return false`.

3.12. Create a `render` function and enter the following styling objects:

```
     var backgroundStyle = {
        padding: 50,
        border: "#333 2px dotted",
        width: 250,
        height: 100,
        borderRadius: 10,
        textAlign: "center"
      };

      var numberStyle = {
        fontSize: 24
      }
```

3.13. The `return` part of the function should:

a. Have an enclosing `<div>` with the `style` set to `{backgroundStyle}`

b. A paragraph with a `style` of `{numberStyle}` that displays the current value of `count`.

c. A paragraph that has the text `'Please inspect the console'`.

d. A `<button>` element who's `onClick` function is set to the ***<u>bound</u>*** `increase` function.

3.14. Save the file and then start the application.

3.15. Open the console and click the button to observe the lifecycle methods being called

3.16. What do you notice about the calls when the component is removed?