Performance Optimization Techniques For ReactJs

Vishnu viswambharan

S7 CSA

Guided By:

Mr. Muhammed Ilyas H

Department of Computer Science Engineering

February 6, 2021

OVERVIEW

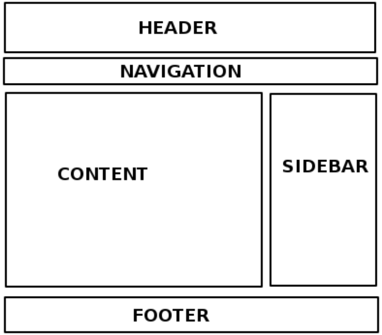
- INTRODUCTION TO REACT JS
- REACT COMPONENTS
- PROPS AND STATE
- LIVE DEMO
- OPTIMIZATION TECHNIQUES
- CONCLUSION

INTRODUCTION TO REACTJS

- React is a JavaScript library which is mainly used for building user interfaces.
- React is developed by Facebook
- React makes use of virtual DOM.
- Whenever any changes are made in the virtualDOM , it is compared with original DOM and the changes are updated in the original DOM.

REACT COMPONENTS

- Components are independent and reusable bits of code.
- They serve the same purpose as JavaScript functions, but work in isolation and returns HTML via a render function.



REACT COMPONENTS

Components can be defined in two ways

Class Components

- The component has to include the extends React.Component statement.
- The component also requires a render() method, this method returns HTML.

Functional Components

- A Function component also returns HTML, and behaves pretty much the same way as a Class component.

PROPS AND STATES

Props

- In a React component, props is an object passed to it by its parent component.
- Props are unchangeable.

State

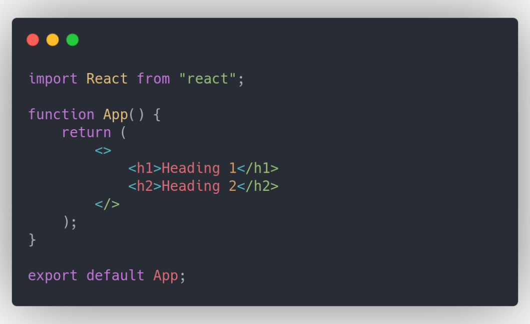
- States are variables managed within a component.
- States can be changed within the component.

LIVE
DEMO

OPTIMIZATION TECHNIQUES

Use React Fragments to Avoid Extra Tags

- Using fragments reduces the number of extra tags that are included, only to fulfill the requirement of having a common parent in the React component.



```
import React from "react";

function App() {
    return (
        <>
            <h1>Heading 1</h1>
            <h2>Heading 2</h2>
        </>
    );
}

export default App;
```

OPTIMIZATION TECHNIQUES

Reducing the number of State and Prop variables

- Every component that has props passed down will be rerendered when that prop is changed.
- Every change in state rerenders the component.
- Avoid unnecessary usage of state and props to reduce rerendering of components.

OPTIMIZATION TECHNIQUES

Do Not Use Inline Function Definition

- If an inline functions is used in a component, every time the “render” function is called,a new instance of the function is created.
- When React does the virtual DOM diffing, it finds a new function instance each time, so during the rendering phase, it binds the new function and leaves the old instance for garbage collection.
- So, directly binding the inline function leads to extra work on the garbage collector and new function binding to the DOM.

OPTIMIZATION TECHNIQUES

Avoid Using Inline Style Attribute

- With inline styles, the browser spends a lot more time scripting and rendering.
- A lot of time is spent on scripting because it has to map all the style rules passed to the actual CSS properties, which increases the rendering time for the component.

OPTIMIZATION TECHNIQUES

Using a Unique Key for Iteration

- Keys help in identifying the items that have been changed, added, or deleted.
- Keys give a stable identity to the element.
- A key should be kept unique for each element on the list.

OPTIMIZATION TECHNIQUES

Optimize Conditional Rendering in React

- Mounting and unmounting React components are costly operations.
- To ensure better performance of the application, the number of mounting and unmounting operations should be reduced.

OPTIMIZATION TECHNIQUES

Avoid Async Requests in componentWillMount()

- componentWillMount will be called right before the component is rendered.
- There is no benefit in calling Async functions like API calls in componentWillMount.
- As the API calls are asynchronous, the component does not wait for the API to return data before calling the render function.So, the component is rendered without any data in the initial rendering.

OPTIMIZATION TECHNIQUES

Immutable Data Structures for Components

- The state and props data in the React component should be immutable if a component must work consistently.
- The mutation of objects can result in inconsistent output..

OPTIMIZATION TECHNIQUES

Using Lazy Loading of Components

- In react, all main components and the external dependencies are merged into a single file and sent over the network to have the web application up and running.
- If this single file becomes a large file, it consumes lots of network bandwidth.
- The concept of code splitting can create multiple bundles for the application which can be dynamically loaded at runtime.
- Loading on runtime reduces the size of the initial bundle that is created.

OPTIMIZATION TECHNIQUES

Multithreading

- While a web page gets rendered, it needs to perform multiple tasks.
- Workers can be seen as an option to reduce the execution load on the main thread.
- Worker threads are the background threads that can enable the user to execute multiple scripts and JavaScript execution without interrupting the main thread.
- Whenever there are long executing tasks that require a lot of CPU utilization, those logical blocks can be executed on the separate thread using workers.

CONCLUSION

- ReactJs is a popular javascript library used for web development.
- It is necessary to optimize a react app for attaining maximum performance.
- React can be easily optimized by following the above methods when developing a React Application.

REFERENCE

1 https://reactjs.org/docs/optimizing-performance.html

2 https://medium.com/technofunnel/https-medium-com-mayank-gupta-6-88-21-performance-optimizations-techniques-for-react-d15fa52c2349d2b1

3 https://developer.mozilla.org/en-US/docs/Web/API/Web-Workers-API/Using-web-workers

Thank You

for Listening.