13. Context API Crash Course with 2 Projects

📘 Index of Key Topics

❓ Questions & In-Depth Answers

1. What is the Context API in React?

Q: What is the Context API, and why is it important?

A: The Context API is a feature in React that allows you to share values (like state) between components without having to explicitly pass props through every level of the component tree. It's particularly useful for global data such as themes, user authentication, or language settings.

Analogy: Think of the Context API as a public bulletin board in an office. Instead of sending memos to each employee individually, you post updates on the board, and everyone can see them.

---

2. How do you set up a Context in React?

Q: What are the steps to create and use a Context?

A: To set up a Context:

1. **Create a Context:**

```
import { createContext } from 'react';

const MyContext = createContext();
```

2. **Provide a Value:**

```
<MyContext.Provider value={/* some value */}>
  {/* components */}
</MyContext.Provider>
```

3. **Consume the Context:**

```
import { useContext } from 'react';

const value = useContext(MyContext);
```

Explanation: The `Provider` component makes the context value available to all components within its subtree, while `useContext` allows components to access the context value.

---

3. How is the Context API applied in the Theme Switcher project?

Q: Can you explain the implementation of the Theme Switcher project?

A: In the Theme Switcher project:

- A `ThemeContext` is created to manage the current theme.
- The `ThemeProvider` component wraps the application, providing the theme value.
- A `ThemeSwitcher` component allows users to toggle between light and dark themes.
- The `useContext` hook is used in components to access and apply the current theme.

Code Snippet:

```
const ThemeContext = createContext();
```

```
function ThemeProvider({ children }) {
  const [theme, setTheme] = useState('light');

  const toggleTheme = () => {
    setTheme(prevTheme => (prevTheme === 'light' ? 'dark' : 'light'));
  };

  return (
    <ThemeContext.Provider value={{ theme, toggleTheme }}>
      {children}
    </ThemeContext.Provider>
  );
}
```

---

4. How is the Context API utilized in the Todo App with Local Storage?

Q: How does the Todo App project utilize the Context API?

A: In the Todo App project:

- A `TodoContext` is created to manage the list of todos and the current filter.
- The `TodoProvider` component wraps the application, providing the todos and filter values.
- The `useContext` hook is used in components to access and update the todos and filter.

Code Snippet:

```
const TodoContext = createContext();

function TodoProvider({ children }) {
  const [todos, setTodos] = useState([]);
  const [filter, setFilter] = useState('all');

  useEffect(() => {
    const savedTodos = JSON.parse(localStorage.getItem('todos'));
    if (savedTodos) {
      setTodos(savedTodos);
```

```
    }
  }, []);

  useEffect(() => {
    localStorage.setItem('todos', JSON.stringify(todos));
  }, [todos]);

  return (
    <TodoContext.Provider value={{ todos, setTodos, filter, setFilter }}>
      {children}
    </TodoContext.Provider>
  );
}
```

5. What are best practices for using the Context API?

Q: What are some recommended practices when using the Context API?

A: Best practices include:

- **Avoid Overuse:** Use the Context API for global state that needs to be accessed by many components. For local state, it's better to use `useState` or `useReducer`.
- **Memoize Context Values:** Use `useMemo` to prevent unnecessary re-renders when the context value changes.
- **Separate Contexts:** Create separate contexts for different pieces of state to avoid unnecessary re-renders.

Example:

```
const value = useMemo(() => ({ theme, toggleTheme }), [theme]);
```

6. Where can I find more resources on React and the Context API?

Q: Where can I learn more about React and the Context API?

A: For a comprehensive understanding, consider exploring the following resources:

- Chai Aur React Series on GitHub: Offers source code and additional materials.
- Chai Aur React YouTube Playlist: Features video tutorials covering various React topics.

🎯 Learning Path Summary

1. **Grasp the Basics:** Understand the fundamental concepts of the Context API.
2. **Build Projects:** Apply your knowledge by building projects that utilize the Context API.
3. **Enhance User Experience:** Add features and optimizations to improve the application's usability and performance.
4. **Continue Learning:** Explore advanced topics and patterns in React to deepen your understanding.
