

State Management Data Flow

1. Introduction

This document provides a detailed Data Flow Diagram (DFD) representation of the State Management React application.

It explains how context API, state updates, and UI changes are handled dynamically.

2. System Overview

The React application consists of the following key components:

- App.js: The main file that initializes the ThemeProvider and renders the ThemeSwitcher component.
- ThemeContext.js: Manages the theme state and provides context to child components.
- ThemeSwitcher Component: Allows users to toggle the theme dynamically.
- useState Hook: Maintains the current theme state.
- useEffect Hook: Updates the UI whenever the theme changes.

3. Data Flow Diagram (DFD)

Level 0 (Context Diagram)

At the highest level, the system consists of external users interacting with the theme toggle button.

External Entities:

- User: Clicks the button to toggle the theme.

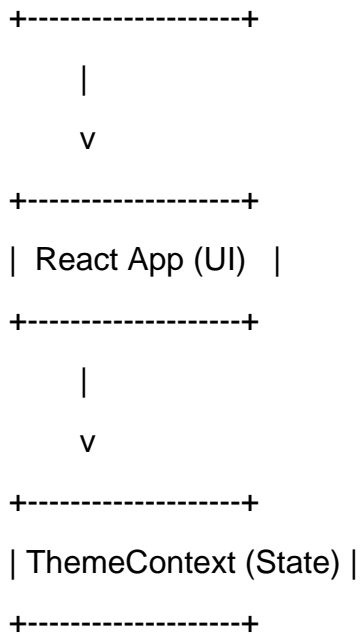
Processes:

- React Application: Handles state updates and re-renders UI components.

Data Stores:

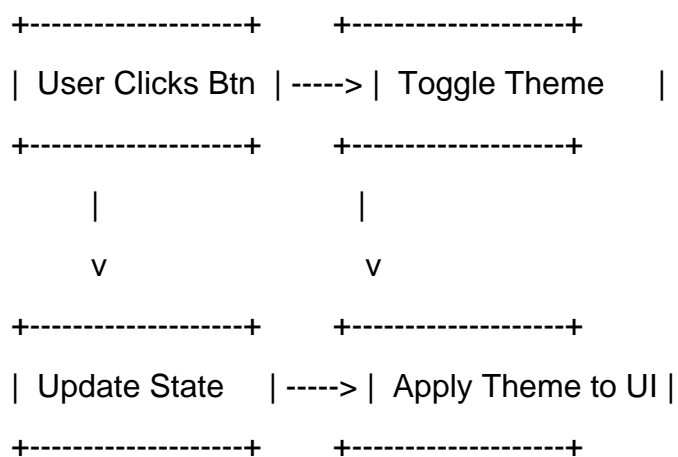
- ThemeContext: Stores the theme state.

```
+-----+
|  User  |
```



Level 1 DFD (Theme Switching Process)

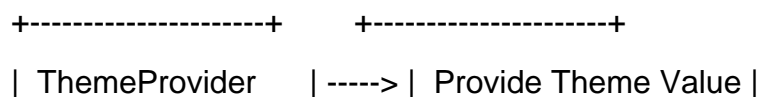
This level breaks down how user actions affect the theme state.

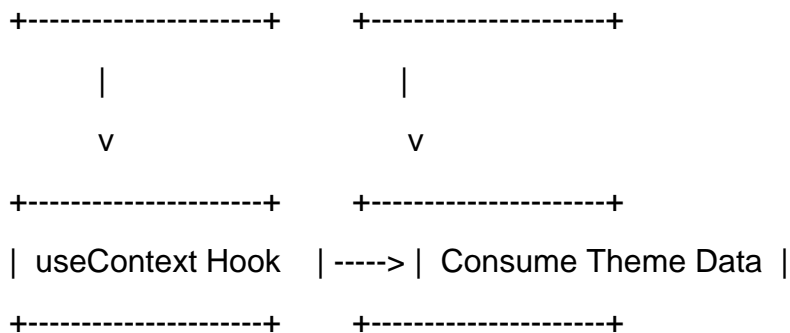


Process Explanation:

1. The user clicks the toggle button in the ThemeSwitcher component.
2. The `setTheme` function updates the theme state.
3. React detects the state change and applies the new theme to the UI.

Level 2 DFD (Context API Flow)





Process Explanation:

1. The ThemeProvider component provides theme state to all child components.
2. The ThemeSwitcher component consumes the theme value using `useContext`.
3. When the theme changes, all components using ThemeContext are re-rendered.

4. Explanation of Data Flow

1. The user interacts with the UI by clicking the toggle button.
2. The ThemeContext updates the state using `setTheme`.
3. The useEffect hook applies the new theme to the UI.
4. The ThemeSwitcher component re-renders to reflect the current theme.

5. Conclusion

This document outlines the detailed data flow in a React-based theme management system. The application efficiently updates and applies UI themes using React Context API and hooks.