

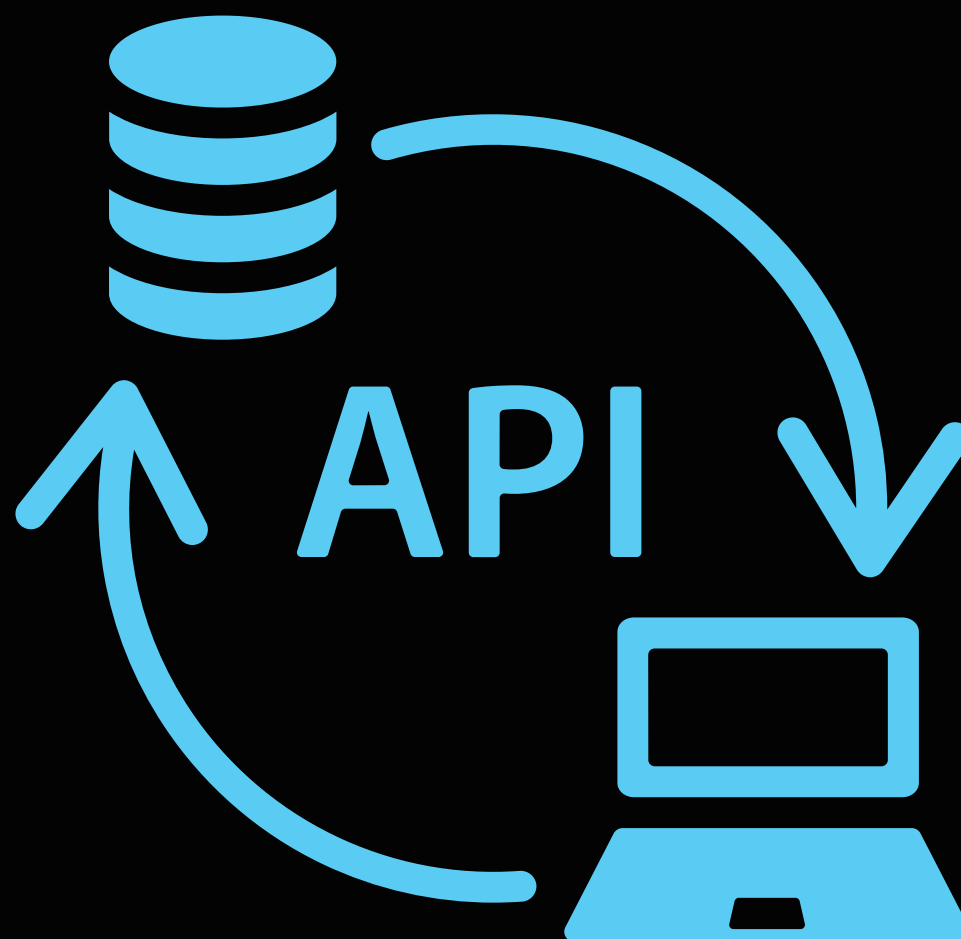
DON'T MISS IT

REACT CONTEXT API



Context API

- The Context API provides a way to pass data through the component tree without having to pass props down manually at every level.
- It is particularly useful for global state management, such as user authentication, themes, or settings.



Using Context API

- Start by creating a context object.

```
import React, { createContext, useState,
useContext } from 'react';

// Create a Context
const ThemeContext = createContext();

// Create a provider component
const ThemeProvider = ({ children }) => {
  const [theme, setTheme] = useState('light');

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

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

export { ThemeProvider, ThemeContext };
```



Using Context API

- Use the context in your components.

```
import React, { useContext } from 'react';
import { ThemeContext } from '../ThemeContext';

const ThemedComponent = () => {
  const { theme, toggleTheme } =
    useContext(ThemeContext);

  return (
    <div style={{ background: theme ===
      'light' ? '#fff' : '#333', color: theme ===
      'light' ? '#000' : '#fff', padding: '20px' }}>
      <h1>The current theme is {theme}</h1>
      <button onClick={toggleTheme}>Toggle Theme</button>
    </div>
  );
};

export default ThemedComponent;
```



Using Context API

- Ensure your components have access to the context by wrapping your app with the provider.

```
import React from 'react';
import ReactDOM from 'react-dom';
import { ThemeProvider } from './ThemeContext';
import ThemedComponent from './ThemedComponent';

const App = () => {
  return (
    <ThemeProvider>
      <ThemedComponent />
    </ThemeProvider>
  );
};

ReactDOM.render(<App />, document.getElementById('root'));
```



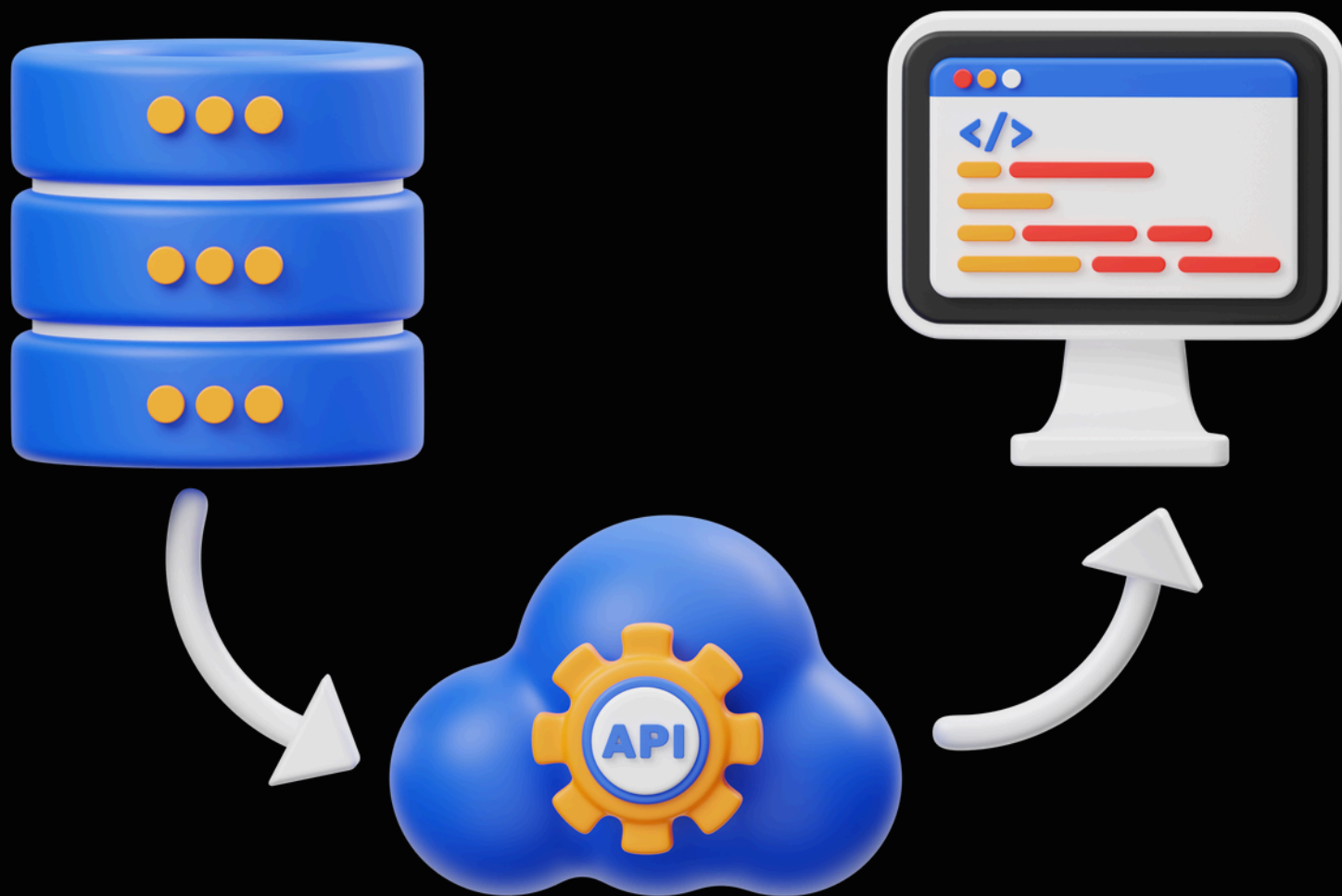
How It Works

- We create a ThemeContext using createContext and a ThemeProvider component that provides the context value to its children.
- The ThemeProvider component manages the theme state and provides a toggleTheme function to toggle between light and dark themes.
- The ThemeProvider component wraps its children with the ThemeContext.



How It Works

- Provider and passes the current theme and toggleTheme function as the context value.
- The ThemedComponent component uses the useContext hook to consume the context value and access the theme and toggleTheme function.



Benefits of the Context API

- No need to pass props through intermediate components.
- Easily manage global state in a centralized way.
- Context values can be consumed by any component, making your application more modular and reusable.



DID YOU FIND IT HELPFUL ?



Share this with a friend who needs it!



VINCENT RAJA

FULL STACK WEB DEVELOPER