

Context API

Manage State Efficiently in React Applications with Context API

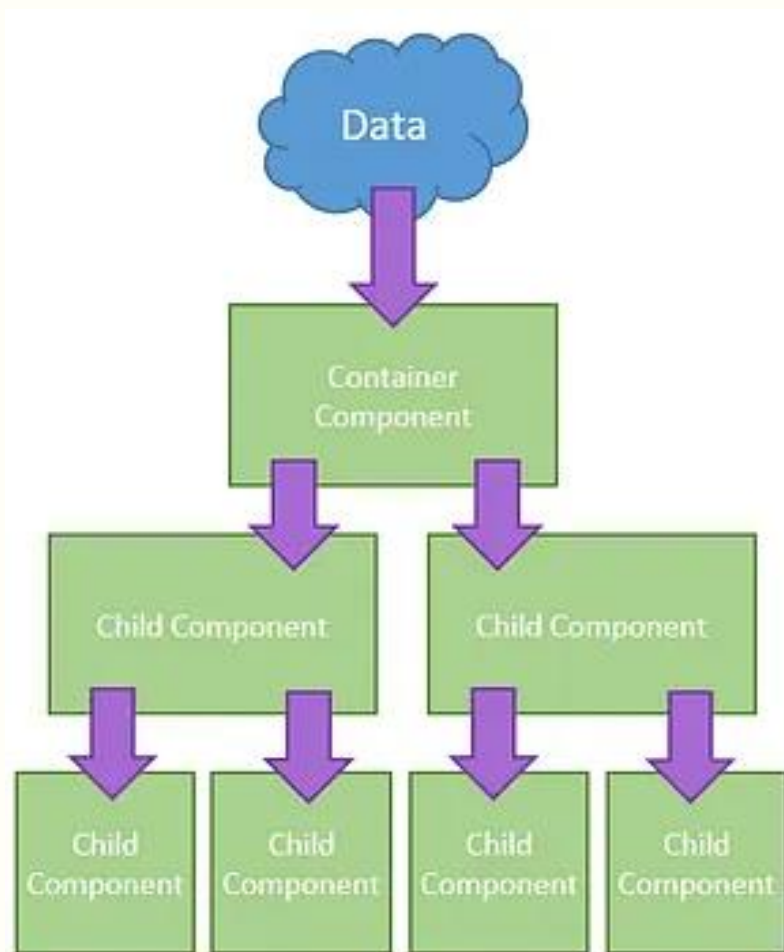


Introduction to Context API. What is Context API?

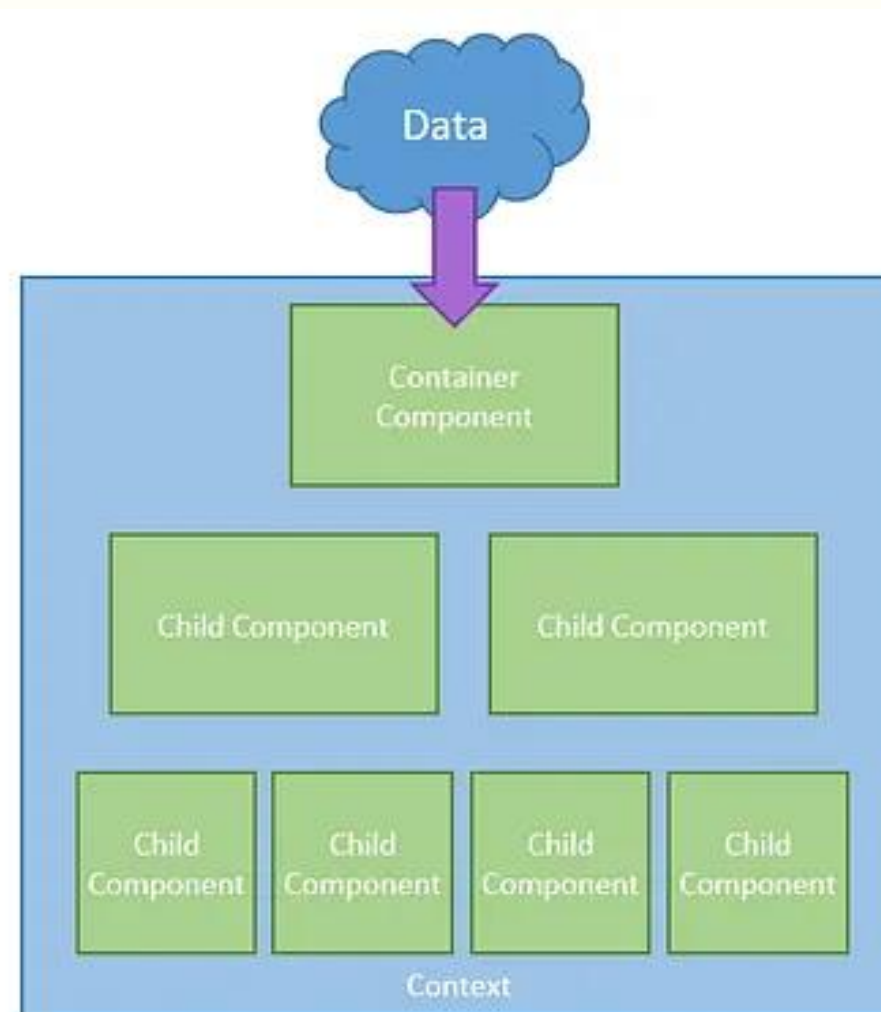


- A way to pass data through the component tree without having to pass props down manually at every level.
- Useful for sharing data like themes, user information, and settings.

Context API is used to fix prop drilling



prop drilling



context API

When to Use Context API



- Ideal Use Cases
 - Theming
 - User authentication
 - Locale settings
 - Avoiding prop drilling for deeply nested components
- When Not to Use
 - Small applications where prop drilling is minimal
 - Situations requiring frequent updates (performance concerns)

Creating Context



```
1  // Create a context
2  const MyContext = React.createContext();
3
4  // Default values can be passed to createContext
5  const MyContext = React.createContext(defaultValue);
6
```

Provider and Consumer



```
1  // Wrap your component tree with the Provider
2  <MyContext.Provider value={/* some value */}>
3    <App />
4  </MyContext.Provider>
5
6  // Consume the context value in a child component
7  <MyContext.Consumer>
8    {value => (
9      /* render something based on the context value */
10    )}
11 </MyContext.Consumer>
```

Using Context in Functional Components



```
1  // Using useContext hook
2  import React, { useContext } from 'react';
3  import { MyContext } from './MyContextProvider';
4
5  function MyComponent() {
6    const value = useContext(MyContext);
7
8    return <div>{value}</div>;
9  }
10
```


Using Context in Class Components




```
1  // Class component using contextType
2  import React, { Component } from 'react';
3  import { MyContext } from './MyContextProvider';
4
5  class MyComponent extends Component {
6    static contextType = MyContext;
7
8    render() {
9      const value = this.context;
10     return <div>{value}</div>;
11   }
12 }
13
```

Best Practices



- Avoid Overusing Context
 - Only use it for state that needs to be shared globally.
- Separate Contexts
 - Create separate contexts for unrelated state to avoid unnecessary re-renders.
- Default Values
 - Utilize default values to make context easier to use and test.
- Performance
 - Be mindful of performance. Updates to context value can trigger re-renders in all consuming components.



The shift begins with

you
Trainer name

Trainer

Assistant name

Assistant

trainer@mail.com
assistant@mail.com