Complete Java Classes

by Kunal Sir

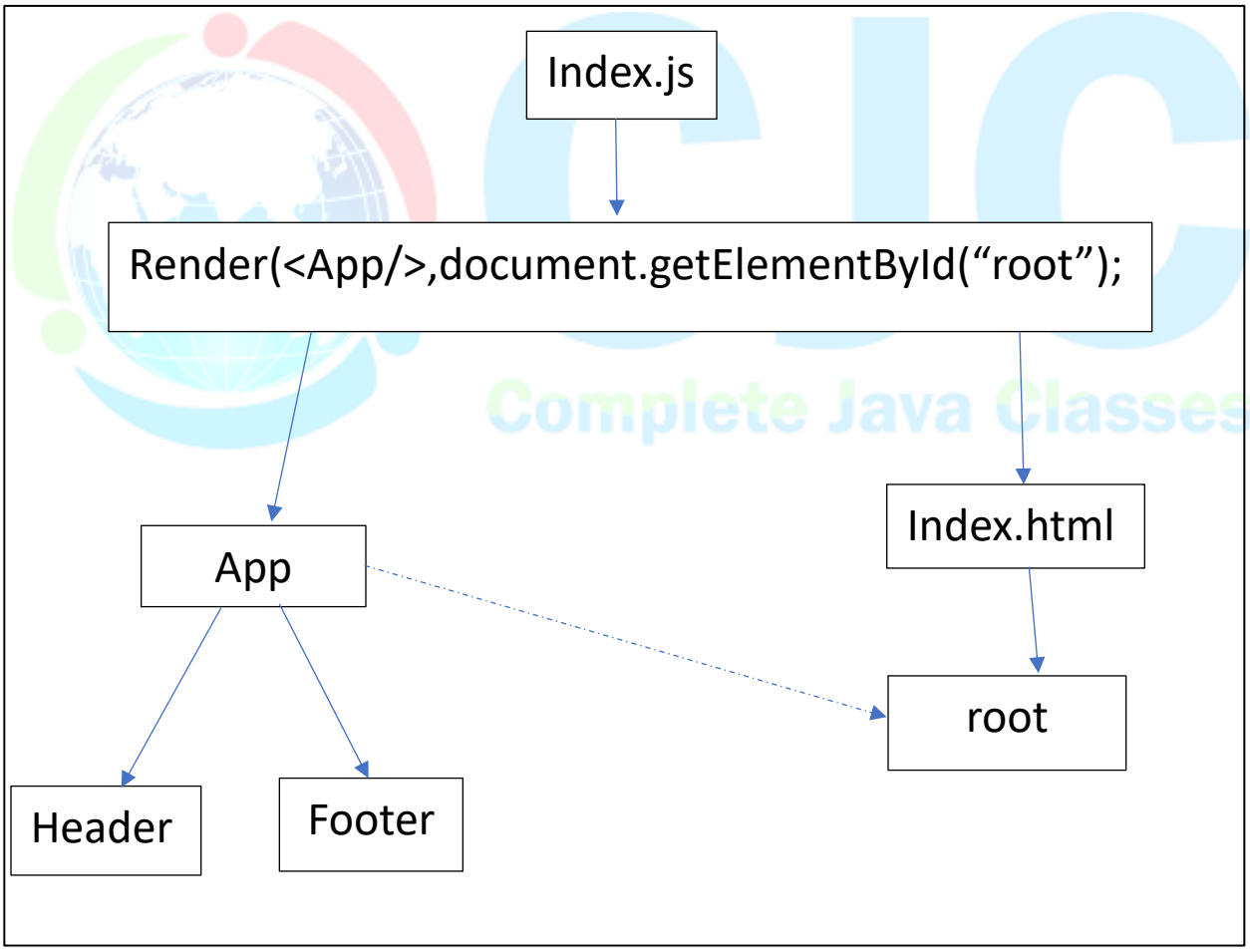
React Application Execution Flow

React application execution flow starts with index.js file.

Once render( ) method get compiled. React will come to know which component (App) needs to render. And where it will be render (“root” element).

If App component has include any custom component, then its JSX code will be added at the place of its selector.

Then all the JSX code will be convert into react elements by babel compiler internally, then those react elements will be injected into root element.



by Kunal Sir

React Components Relationship

React Components forms a tree structure to create the layout of our application.

It means we have to render some component in another components.

The component which render / include another component in it. Is called parent component. And the component which is render by other component is called child component.

Parent components can shear the data to its child component with the help of props.

But it is not convenient to shear the data from parent component to its grand child component or its grate grand child components with the help of props.

Context API

The Context API is a mechanism that allow you to share specific information (like state or functions) with multiple child components eliminating the need of props drilling.

The react context API is a powerful tool for efficient state management. Offering a cleaner alternative to props drilling and enhancing overall code organization.

In short, whenever it is required to shear the data from parent component to grand child component or to multiple components. Then we must go with context API.

by Kunal Sir

Context API provides a common place, where parent can store the data. And all child and grand child can get the data, stored by parent.

Using createContext() function we can create the context in react.

Using useContext() function we can get/receive the data in child component.

To set the data in context we use `value` attribute of predefine component `provider`.

We need to call provider component explicitly.

Steps To Implement Context API:-

1. Import CreateContext from react library.

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

2. Create The context in Parent component, and export it as a named export.

```
export const context=createContext();
```

3. In parent component call the Provider component and pass Child Component as a children on Provider component, and using value attribute set the data in context Provider.

```
function Parent()
{
    const  course={
        courseName:'ReactJS',
        time:'2:00 PM',
        fees:95000
    }

    return <div className='ParentContainer'>
            <h1>Parent Component Work</h1>
            <context.Provider value={course}>
                    <Child/>
            </context.Provider>
    </div>
}
```

4. Import useContext( ) hook in child component or Demo.js.

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

5. Import the context from Parent.js in Child.js or Demo.js

```
import { context } from './../parent/Parent.js'
```

6. Using useContext() hook get the data of context in child (same for Demo).

```
function Child()
{
 const data  =useContext(context);

   return <div className='ChaildContainer'>
        <h1>Child Component Work.</h1>
          <h3>{data.courseName}</h3>
          <h3>{data.fees}</h3>

        <Demo/>
    </div>
}
```