

ReactJS - State Management

State management is one of the important and unavoidable features of any dynamic application. React provides a simple and flexible API to support state management in a React component. Let us understand how to maintain state in React application in this chapter.

What is state?

State represents the value of a dynamic properties of a React component at a given instance. React provides a dynamic data store for each component. The internal data represents the state of a React component and can be accessed using `this.state` member variable of the component. Whenever the state of the component is changed, the component will re-render itself by calling the `render()` method along with the new state.

A simple example to better understand the state management is to analyse a real-time clock component. The clock component primary job is to show the date and time of a location at the given instance. As the current time will change every second, the clock component should maintain the current date and time in it's state. As the state of the clock component changes every second, the clock's `render()` method will be called every second and the `render()` method show the current time using it's current state.

The simple representation of the state is as follows –

```
{
  date: '2020-10-10 10:10:10'
}
```

Let us create a new *Clock* component later in this chapter.

Here,

[State management API](#)

[Stateless component](#)

[State management using React Hooks](#)

[Component Life cycle](#)

[Component life cycle using React Hooks](#)

[Layout in component](#)

Pagination

Material UI

Revision #1

Created 14 December 2022 10:42:34 by Admin

Updated 14 December 2022 10:43:19 by Admin