

## Life Cycle Methods

### App.js

```
import './App.css';
import LifecycleMethods from './components/LifecycleMethods';

function App() {
  return (
    <div>
      <LifecycleMethods value={10} />
    </div>
  );
}

export default App;
```

### LifeCycleMethods.js

```
import React, { Component } from "react";
import DummyComponent from "./DummyComponent";

export class LifecycleMethods extends Component {
  constructor(props) {
    super(props);
    console.log('constructor')
    this.state = {
      data: [],
      value: 0,
    };
    this.handleClick = this.handleClick.bind(this);
  }

  static getDerivedStateFromProps(props, state){
    console.log('getDerivedStateFromProps', props.value)
    return (prevState => ({...prevState, value: props.value}))
  }

  getData = () => {
    fetch('https://jsonplaceholder.typicode.com/users')
      .then(response => response.json())
      .then(res => {
        this.setState({data: res});
      })
      .catch(err=> console.log(err))
  }
}
```

```

    handleClick(event){
      console.log('Clicked ');
      this.setState({value: this.state.value + 1});
    }
    componentDidMount() {
      console.log('componentDidMount')
      this.getData();
    }

    shouldComponentUpdate(){
      console.log('shouldComponentUpdate')
      return true;
    }
    getSnapshotBeforeUpdate(prevProps, prevState){
      console.log('getSnapshotBeforeUpdate')
      document.getElementById('prev-state').innerHTML=` Previous State
${prevState.value}`
      return prevState.value;
    }
    componentDidUpdate(){
      console.log('componentDidUpdate')
      document.getElementById('curr-state').innerHTML=` Current State
${this.state.value}`
    }
    componentWillUnmount(){
      console.log('Unmounting Component');
    }
    render() {
      return <div>
        <h1 id='prev-state'>H1 Text</h1>
        <h2 id='curr-state'>H2 Text</h2>
        <button onClick={this.handleClick}>Click Me</button>
        {this.state.data.map((item)=>{
          return (
            <td key={item.id}>{item.id}</td>
          )
        })}
        {this.state.value%2 !== 0 ? <DummyComponent/> : null}
      </div>;
    }
  }
}

```

```
export default LifecycleMethods;
```

Dummy Component.js

```
import React, { Component } from 'react'

export class DummyComponent extends Component {
  componentWillUnmount(){
    console.log('Unmounting Dummy component');
  }
  render() {
    return (
      <div>DummyComponent</div>
    )
  }
}

export default DummyComponent
```