



WEB TECHNOLOGIES

React.JS – Stateful Components

Vinay Joshi

Department of
Computer Science and Engineering

React.JS – Stateful Components

Introduction



- So far, we have seen static or stateless Components. They don't undergo state changes.
- Components need to change based on
 - User actions (clicks, keyboard inputs, etc.)
 - Other triggers (responses received from server, timers, etc.)



React.JS – Stateful Components

Counter Component

- Let's consider this Component that shows the number of seconds the user has been on the page

122
seconds
since you loaded
the page



React.JS – Stateful Components

Counter Component...(cntd.)

- We consider two Components, CounterDisplay and Counter

```
ReactDOM.render(  
  <CounterDisplay/>,  
  document.querySelector("#container")  
);  
  
class CounterDisplay extends React.Component {  
  render() {  
    return (  
      <div>  
        <Counter/>  
        <h2>seconds </h2>  
        <h2>since you loaded the page</h2>  
      </div>  
    );  
  }  
}
```

```
class Counter extends React.Component {  
  render() {  
    return (  
      ...  
    );  
  }  
}
```



React.JS – Stateful Components

Counter Component...(cntd.)



- We use the ***constructor*** method to initialize the counter
- The Component also has the method ***componentDidMount*** that can be used to start the counter
- It also exposes the ***setState*** method to update the state (the counter)



React.JS – Stateful Components

Counter Component...(cntd.)

- We use these methods as follows

```
class Counter extends React.Component {  
  constructor(props, context) {  
    super(props, context);  
  
    this.state = {  
      seconds: 0  
    };  
  }  
  
  render() {  
    return (  
      <h1>{this.state.seconds}</h1>  
    );  
  }  
}
```

Add this code after the constructor:

```
componentDidMount() {  
  setInterval(this.timerTick, 1000);  
}  
timerTick() {  
  this.setState({  
    seconds: this.state.seconds + 1  
  });  
}
```



React.JS – Stateful Components

Counter Component...(cntd.)

- To bring in the stateful Component behaviour, that also ensures that state variable seconds is never out of sync

```
this.setState((prevState) => {  
  return {  
    seconds: prevState.seconds + 1  
  };  
});
```

- To ensure that the this.state works well in the timerTick method, add the following line to call that function with the context of the Component

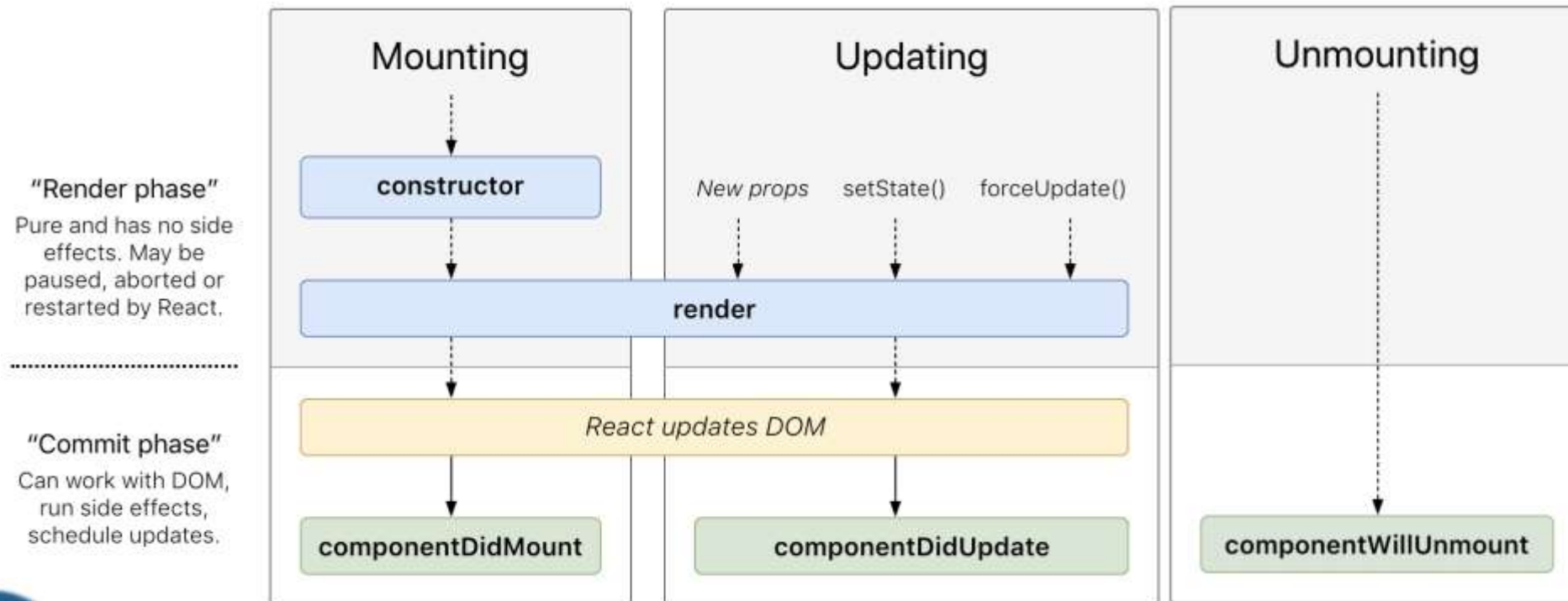
```
constructor(props, context) {  
  ...  
  this.timerTick = this.timerTick.bind(this);  
}
```



React.JS – Stateful Components

Component Life Cycle

- Components undergo these following life cycle states and their associated events





THANK YOU

Vinay Joshi

Department of Computer Science and Engineering

vinayj@pes.edu

+91 80 2672 6622