**REACTJS**

**PART 2**

**State & Lifecycle**

Function Clock(props) {

Return(

<div>

<h1>Hello, World</h1>

<h2>It is {props.date.toLocaleTimeString()}.</h2>

</div>

);

};

Function tick(){

ReactDOM.render(

<Clock date={new Date()} />,

Document.getElementById(‘root’)

);

}

setInterval(tick, 1000);

In this section we will learn how to make the clock component truly reusable and encapsulated. It will set up its own timer and update itself every second. The example above misses a crucial requirement: the fact that the clock sets up a timer and updates the UI every second should be an implementation detail of the clock. Ideally we want to write this once and the clock update itself:

ReactDOM.render(<Clock />, document.getElementById(‘root’));

To implement this, we need to add “state” to the Clock component. State is similar to props, but it is private and fully controlled by the component. Local state is a feature only available to classes (not functions).

Converting a function to a class

1. Create an ES6 class with the same name, that extends React.Component
2. Add a single empty method to it called render()
3. Move the body of the function in to the render() method
4. Replace props with *this.props* in the render() body
5. Delete the remaining empty function decleration

Example:

Class Clock extends React.Component{

Render(){

Return(

<div>

<h1>Hello, World!</h1>

<h2>It is {this.props.date.toLocaleTimeString()}.</h2>

</div>

);

}

}

This lets us use additional features such as local state and lifecycle hooks.

Adding local state to a class

//we will move the date from *props* to *state* in 3 steps:

1. Replace *this.props.date* with *this.state.date* in the render method
2. Add a class constructor that assigns the initial *this.state*:

Class Clock extends React.Component{

Constructor(props){

Super(props);

This.state = {date: new Date()};

}

Render(){

Return(

<div>

<h1>Hello, World!</h1>

<h2>it is, {this.state.date.toLocalTimeString()}.</h2>

</div>

);

}

}

//note how we pass *props* to the base constructor. Class components should always call the base constructor with *props*.

1. Remove the date from the <Clock /> element.

//next we will make the clock set up its own timer and update itself every second

Adding lifecycle methods to a class