

06. REACT Component API

In this chapter, we will explain React component API. We will discuss three methods: *setState()*, *forceUpdate* and *ReactDOM.findDOMNode()*. In new ES6 classes, we have to manually bind this. We will use *this.method.bind(this)* in the examples.

6.1. Set State

setState() method is used to update the state of the component. This method will not replace the state, but only add changes to the original state.

Example:

```
import React from 'react';

class App extends React.Component {

  constructor() {
    super();

    this.state = {
      data: []
    }

    this.setStateHandler = this.setStateHandler.bind(this);
  };

  setStateHandler() {
    var item = "setState..."
    var myArray = this.state.data.slice();
    myArray.push(item);
    this.setState({data: myArray})
  };

  render() {
```

```
        return (  
            <div>  
                <button onClick = {this.setStateHandler}>SET  
STATE</button>  
                <h4>State Array: {this.state.data}</h4>  
            </div>  
        );  
    }  
}  
  
export default App;
```

We started with an empty array. Every time we click the button, the state will be updated. If we click five times, we will get the following output.

Ouput:



6.2. Force Update

Sometimes we might want to update the component manually. This can be achieved using the *forceUpdate()* method.

Example:

```
import React from 'react';

class App extends React.Component {
  constructor() {
    super();
    this.forceUpdateHandler = this.forceUpdateHandler.bind(this);
  };
  forceUpdateHandler() {
    this.forceUpdate();
  };
  render() {
    return (
      <div>
        <button onClick = {this.forceUpdateHandler}>FORCE UPDATE</
button>
        <h4>Random number: {Math.random()}</h4>
      </div>
    );
  }
}

export default App;
```

We are setting a random number that will be updated every time the button is clicked.

Output:



6.3. Find Dom Node

For DOM manipulation, we can use **ReactDOM.findDOMNode()** method. First we need to import react-dom.

Example:

```
import React from 'react';
import ReactDOM from 'react-dom';

class App extends React.Component {

  constructor() {
    super();
    this.findDomNodeHandler = this.findDomNodeHandler.bind(this);
  };

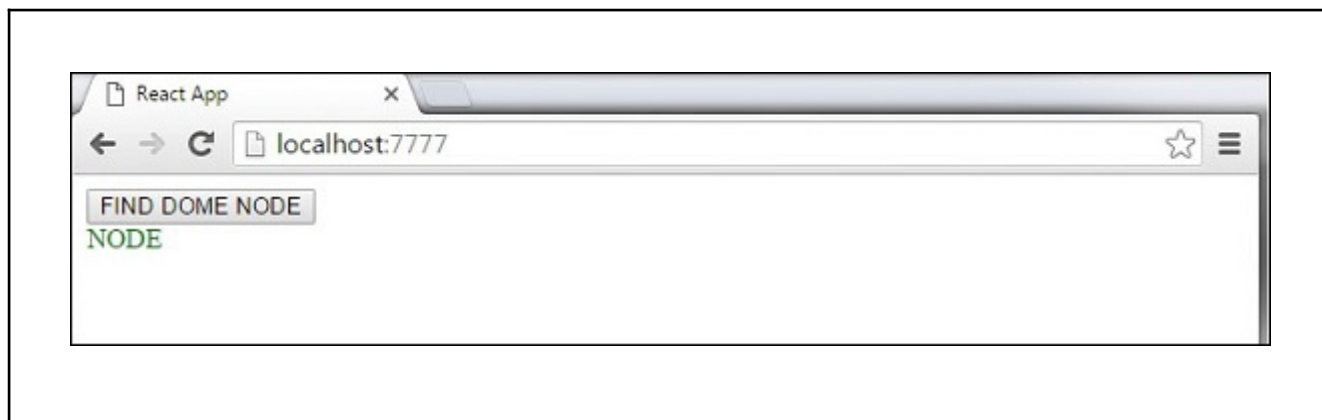
  findDomNodeHandler() {
    var myDiv = document.getElementById('myDiv');
    ReactDOM.findDOMNode(myDiv).style.color = 'green';
  }

  render() {
    return (
      <div>
        <button onClick = {this.findDomNodeHandler}>FIND DOME
        NODE</button>
        <div id = "myDiv">NODE</div>
      </div>
    );
  }
}

export default App;
```

The color of myDiv element changes to green, once the button is clicked.

Output:



Note – Since the 0.14 update, most of the older component API methods are deprecated or removed to accommodate ES6.