

8. REACT Forms

In this chapter, we will learn how to use **forms** in React.

8.1. Form Simple Example

In the following example, we will set an input form with *value* = *{this.state.data}*. This allows to update the state whenever the input value changes. We are using *onChange* event that will watch the input changes and update the state accordingly.

Example: App.jsx

```
import React from 'react';

class App extends React.Component {

  constructor(props) {
    super(props);

    this.state = {
      data: 'Initial data...'
    }
    this.updateState = this.updateState.bind(this);
  };

  updateState(e) {
    this.setState({data: e.target.value});
  }

  render() {
    return (
      <div>
        <input type = "text" value = {this.state.data}
          onChange = {this.updateState} />
      </div>
    );
  }
}
```

```
        <h4>{this.state.data}</h4>
      </div>
    );
  }
}
export default App;
```

Example: main.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App.jsx';

ReactDOM.render(<App/>, document.getElementById('app'));
```

When the input text value changes, the state will be updated.

Output:



8.2. Form Complex Example

In the following example, we will see how to use forms from child component. *onChange* method will trigger state update that will be passed to the child input value and rendered on the screen. A similar example is used in the **Events** chapter. Whenever we need to update state from child component, we need to pass the function that will handle updating (*updateState*) as a prop (*updateStateProp*).

Example: App.jsx

```
import React from 'react';

class App extends React.Component {

  constructor(props) {
    super(props);

    this.state = {
      data: 'Initial data...'
    }
    this.updateState = this.updateState.bind(this);
  };

  updateState(e) {
    this.setState({data: e.target.value});
  }

  render() {
    return (
      <div>
        <Content myDataProp = {this.state.data}
          updateStateProp = {this.updateState}></Content>
      </div>
    );
  }
}
```

```
    }  
  }  
  
  class Content extends React.Component {  
  
    render() {  
  
      return (  
  
        <div>  
          <input type = "text" value = {this.props.myDataProp}  
            onChange = {this.props.updateStateProp} />  
          <h3>{this.props.myDataProp}</h3>  
        </div>  
  
      );  
  
    }  
  
  }  
  
  export default App;
```

Example: main.js

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import App from './App.jsx';  
  
ReactDOM.render(<App/>, document.getElementById('app'));
```

This will produce the following result.

Output:

