Uncontrolled Components

To write an uncontrolled component, instead of writing an event handler for every state update, you can use a ref to get form values from the DOM. In uncontrolled input we get data from input fields like traditional HTML form data handling.

An example of uncontrolled component

Getting data from an uncontrolled input

import React, { Component } from 'react'

import ReactDOM from 'react-dom'

class App extends Component {

firstName = React.createRef()

handleSubmit = (e) => {

e.preventDefault()

console.log(this.firstName.current.value)

}

render() {

return (

<div className='App'>

<form onSubmit={this.handleSubmit}>

<label htmlFor='firstName'>First Name: </label>

<input

type='text'

id='firstName'

name='firstName'

placeholder='First Name'

ref={this.firstName}

/>

<button type='submit'>Submit</button>

</form>

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)

Getting multiple input data from form

We can grab multiple input data from DOM. We are not directly targeting the DOM but React is getting data from DOM using ref.

import React, { Component } from 'react'

import ReactDOM from 'react-dom'

class App extends Component {

firstName = React.createRef()

lastName = React.createRef()

country = React.createRef()

title = React.createRef()

handleSubmit = (e) => {

// stops the default behavior of form element specifically refreshing of page

e.preventDefault()

console.log(this.firstName.current.value)

console.log(this.lastName.current.value)

console.log(this.title.current.value)

console.log(this.country.current.value)

const data = {

firstName: this.firstName.current.value,

lastName: this.lastName.current.value,

title: this.title.current.value,

country: this.country.current.value,

}

// the is the place we connect backend api to send the data to the database

console.log(data)

}

render() {

return (

<div className='App'>

<h3>Add Student</h3>

<form onSubmit={this.handleSubmit}>

<div>

<input

type='text'

name='firstName'

placeholder='First Name'

ref={this.firstName}

onChange={this.handleChange}

/>

</div>

<div>

<input

type='text'

name='lastName'

placeholder='Last Name'

ref={this.lastName}

onChange={this.handleChange}

/>

</div>

<div>

<input

type='text'

name='country'

placeholder='Country'

ref={this.country}

onChange={this.handleChange}

/>

</div>

<div>

<input

type='text'

name='title'

placeholder='Title'

ref={this.title}

onChange={this.handleChange}

/>

</div>

<button className='btn btn-success'>Submit</button>

</form>

</div>

)

}

}

const rootElement = document.getElementById('root')

ReactDOM.render(<App />, rootElement)

Most of the time we use controlled input instead of uncontrolled input. In case if you want to target

some element on the DOM you will use ref to get the content of that element. Don't touch directly using pure JavaScript. When you develop a React application do not touch the DOM directly because React has its own way of handling the DOM manipulation.