Controlled vs Uncontrolled Components

* Controlled component

In React, Controlled Components are those in which form’s data is handled by the component’s state. It takes its current value through props and makes changes through callbacks like onClick, onChange, etc. A parent component manages its own state and passes the new values as props to the controlled component.

**Example**: We are creating a simple form that comprises an input field with a label name and a submit button. We are creating an onSubmit function that triggers when we submit the form which shows the name that we have filled in the alert box.

We are creating a state name that stores the value we enter into the input field using the [useState](https://www.geeksforgeeks.org/reactjs-usestate-hook/) hook. We are creating an onChange function in the input field that stores the data we are entering in the input field to our state. There is another function handleSumit that gets triggered when we submit the form and it shows the name we have entered on the alert box.

**App.js**

|  |
| --- |
| import { useState } from 'react';  function App() {    const [name, setName] = useState('');    function handleSubmit() {      alert(`Name: ${name}`);    }    return (      <div className="App">        <h3>Controlled Component</h3>        <form onSubmit={handleSubmit}>          <label>Name:</label>          <input name="name" value={name} onChange={(e) => setName(e.target.value)} />          <button type="submit">Submit</button>        </form>      </div>    );  }  export default App; |

**Step to Run Application:** Run the application using the following command from the root directory of the project:

npm start

**Output:**



* UnControlled component

Uncontrolled Components are the components that are not controlled by the React state and are handled by the [DOM](https://www.geeksforgeeks.org/dom-document-object-model/) (Document Object Model). So in order to access any value that has been entered we take the help of refs.

For instance, if we want to add a file as an input, this cannot be controlled as this depends on the browser so this is an example of an uncontrolled input.

**Prerequisite:**

* Introduction and installation [ReactJs](http://www.geeksforgeeks.org/react-js-introduction-working/)
* ReactJS [useRef Hook](https://www.geeksforgeeks.org/react-js-useref-hook/)
* ReactJS[useState Hook](https://www.geeksforgeeks.org/reactjs-usestate-hook/)

**Creating React Application:**

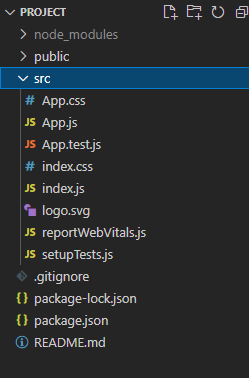
**Step 1:** Create the react app using the following command:

npm create-react-app project

**Step 2:** After creating your project folder(i.e. project), move to it by using the following command:

cd project

**Project Structure:**It will look like this.



**Example:**We are creating a simple form that comprises an input field with a label name and a submit button. We are creating an onSubmit function that triggers when we submit the form which shows the name that we have filled in the alert box. We are accessing the name we have filled using [useRef](https://www.geeksforgeeks.org/react-js-useref-hook/).

**App.js**

|  |
| --- |
| import React, { useRef } from 'react';  function App() {    const inputRef = useRef(null);    function handleSubmit() {      alert(`Name: ${inputRef.current.value}`);    }    return (      <div className="App">        <h3>Uncontrolled Component</h3>        <form onSubmit={handleSubmit}>          <label>Name :</label>          <input type="text" name="name" ref={inputRef} />          <button type="submit">Submit</button>        </form>      </div>    );  }  export default App; |

**Step to Run Application:**Run the application using the following command from the root directory of the project:

npm start

**Output:**



* Controlled vs Uncontrolled components in React JS

|  |  |
| --- | --- |
| Controlled  Component | Uncontrolled Component |
| The component is under control of the component’s state. | Components are under the control of DOM. |
| These components are predictable as are controlled by the state of the component. | Are Uncontrolled because during the life cycle methods the data may loss |
| Internal state is not maintained | Internal state is maintained |
| It accepts the current value as props | We access the values using refs |
| Does not maintain its internal state. | Maintains its internal state. |
| Controlled by the parent component. | Controlled by the DOM itself. |
| Have better control on the form data and values. | Has very limited control over form values and data. |