Rendering Content Conditionally:

* First way

// App.js

import React, { useState } from 'react';

import User from './User/User'

class App extends React.Component {

state = {

user: {

name: "Sagar",

age: 29

},

showUser: false

}

showUserHandler = () =>{

this.setState({

showUser: !this.state.showUser

});

}

render() {

return (

<div>

<button onClick={this.showUserHandler}>Show User</button>

{ this.state.showUser ? <User name={this.state.user.name} age={this.state.user.age} /> : null }

</div>

);

}

}

export default App;

* Second Way & Recomended (Use this on Realtime)

render() {

let user = null;

if(this.state.showUser){

user = (<User name={this.state.user.name} age={this.state.user.age} />);

}

return (

<div>

<button onClick={this.showUserHandler}>Show User</button>

{ user }

</div>

);

}

Outputting Lists:

class App extends React.Component {

state = {

users: [

{ name:'User One', age:10 },

{ name:'User Two', age:20 },

{ name:'User Three', age:30 }

]

}

render() {

return (

<div>

<div>

{ this.state.users.map(user =>{

return <User name={user.name} age={user.age} />

})}

</div>

</div>

);

}

}

Deleting a user from List:

* Note: Don’t directly delete a user from state use as follow
  + Create a copy of an array from state
  + Manipulate it based on your requirement
  + setState with new data
* You should always update state in immutable way

// App.js

import React, { useState } from 'react';

import User from './User/User'

class App extends React.Component {

state = {

users: [

{ name:'User One', age:10 },

{ name:'User Two', age:20 },

{ name:'User Three', age:30 }

]

}

deleteUserHandler = (userIndex) =>{

// Way one

const updatedUsers1 = this.state.users.slice();

// Way two

const updatedUsers2 = [...this.state.users];

updatedUsers2.splice(userIndex,1);

this.setState({

users: updatedUsers2

})

}

render() {

return (

<div>

<div>

{ this.state.users.map((user, index) => {

return <User

name={user.name}

age={user.age}

deleteUser={this.deleteUserHandler.bind(this, index)}/>

})}

</div>

</div>

);

}

}

export default App;

// User.js

import React from 'react';

const User = (props) =>{

return (

<div>

<p>I am {props.name}, My age is {props.age}</p>

<p onClick={props.deleteUser}>Delete</p>

</div>

);

}

export default User;

Lists & Key:

* When ever data changes react need to re draw DOM
* If we wont use “key” it will re draw whole DOM
* So, use unique value in key
* **Note: Don’t use array index**

<User key={user.id} name={user.name} age={user.age}/>

Useful Resources & Links:

* Conditional Rendering: <https://reactjs.org/docs/conditional-rendering.html>
* Lists & Keys: <https://reactjs.org/docs/lists-and-keys.html>