Saturday, February 5, 2022 9:54 AM

Two way communication between JS and DOM (Screen) -------->

We can show content in JS ==> screen {varname}

We can get the content from input tag on screen === > JS <input onSomeEvent={ (event)=>{varname = event.target.value} />

We can send parent component data ====> child component data in the parent <ChildCompo data="ttt" />
in the child props.data or this.props.data

To send a call from child component ====> method of parent component

REACT uses HAS A (Composition instead of Inheritance)
---Parent Child } we think of Inheritance

```
Render()  ………parent
<div>
 …..
 ….
  <ChildComponent />
</div>
```

___

Life Cycle Of React Class Components -------------

1. componentDidMount = React container calls the method after the component is MOUNTED On the DOM
   We can write code to initialize data here (similar to init() in servlet)
2. componentDidUpdate = React container calls the method when the component changes state
3. componentWillUnMount = React container calls the method when the component is about to be (before) UNMOUNTED from the DOM (similar to destroy in servlet)

___

Rendering Multiple React components ------using List

We will use list.map and generate the corresponding html/component for each element
For each component we must give a key , to avoid the warning (this key may be used by us or react container while re -rendering , the key must be unique)

FOR changing any object

```
    Arr[i] = new object
Arr[i].comment = changed comment
```

___

Access REST API from React

1. Use fetch API
2. Make sure that the controller is annotated by @CrossOrigin (we are on 3000 and we are connecting on 8080)
3. Promise

Promise object = Future object !!!

Var promise = fetch(………..)

promise.then(callback) // this callback is executed whenever request succeeds , we get response
promise.catch (callback) //this callback is executed when request fails , we get error

Anotherpromise = (jab server se data ayega).then(response =>{ return response.json() }

(jab json me convert hoga).then (jsondata => {   setState ({data: })

OR

We can use simple Javascript AJAX   XMLHttpRequest()

OR

Http libraries ---------------- axion library can be installed and used