

Good morning :))

React project --- node-modules , public src =====>React Container===>BUILD
|
|
|-----> 3 JS files and htmls --- polyfills.js , main.js,...

React Deployment ---

1. React project BUILD happens
2. React Build is copied in the server folder = DEPLOYMENT on the server
3. Request from browser <http://localhost:3000>
 - a. The React JS files and htmls land on the browser

Errors 2 types

1. Development and build time error --- seen on server side Terminal
2. After Request ----seen on browser console

1. JSX

- View
- { variable }
- { function }
- Conditional Rendering { flag ? Tag : "" }
- OR
- {flag && Tag }
- List Rendering --- arr.map ()

2. Event Handling --- register a handler

- event passing
- without writing a function we can use the handler
- OnClick = {(event) => {}}

3. PROPS --- passing data from parent to child
child to parent

4. State - function =useState
class = this.state , this.setState
Rerendering the components

5. Lifecycle of React Components ----

Who is the managing the lifecycle ? React Container

What is the meaning of managing life cycle ?

- 1. create the component
2. invoke callbacks
3. manage props and state data
4. Manage Rerendering

Component Created
Component Updated
Component Destroy

IF we want the container to **notify** us(Component) during different lifecycle activities
-----How ???

--- CALLBACK function ---- Lifecycle Functions / methods

Class component --- life cycle functions

componentDidUpdate ---- notify that the props state of the component got changed

componentDidMount ---- notify that the component got mounted on the DOM tree

componentWillUnmount -- notify that the component is about to be UNMOUNTED

Function component ---- **useEffect** () HOOK is used for lifecycle notification

Ex 1 -- create a folder day3 , write App2.js , add it to index.js

Write a class componet LifeCycleEx

Add it to App2.js

Observation --- componentDidMount and componentWillUnmount are called only Once in the life cycle
componentDidUpdate() ----- n number of times

useEffect(callback , [])	This callback is called only on MOUNTING
useEffect(callback , [props.msg, flag])	This callback is called on updation Of specified props or state
useEffect(()=>{ return callbackX }, [])	callbackX is called just before unmounting The component

Asynchronous functions

1. Promises

Promise = something to be done in the future

Async functions ---

callback function will return LATER as compared to current flow of execution

ISSUE = where to handle the result of the Delayed Callback Function

What can be the result of the Delayed Callback Function?

Promise is PENDING = it is yet to be executed

Promise is Successful ==>RESOLVES = there is function called as **"then()**

Promise Fails ==>REJECTS = there is function called as **catch()**

promise.then(callback that should be called WHEN promise RESOLVES)

Javascript Process ==

1. Program stack = highest priority
2. Heap
3. Callback Queue = 3rd highest priority
4. Job Queue --- Promises and the then and catch callbacks = 2nd highest priority

React AJAX Call

External Ajax library = axios

To get the library = npm install axios

Ex 3 --- AjaxEx functional component

Ex 4 - AjaxEx class component

When the page appears show all the firstnames in a dropdown list

Whenever a first

