**React Js Fundamental**

1. reactjs, node js, npm (node package manager), vs code, axios(third party)

2. redux

---------------------

1. How it work

a)it create a virtual DOM and get the changed data from dom.

2. Setup ReactJs

a) using CDN Link bellow mention

b) copy the 2 link which is mention in react js officeal website

c) Babel - modern js code convert uding ES6()

babeljs officeal website->setup->in the browser-> copy 2 link

Note- this is not recommended because we are manually creating the project structure because of that we use NPM

Without Using these

1. it take lot of code to make react js structure, so many time we have to create a nested element if nested div is required

Requirement

1. NPM

2. Webpack- it build the dependency graph which maps evry module your project needs and generate one or more module

3. Babel-ECMAScript 2015 + code into a backwards compatible version of javaScript in curret and older browseres

4. JSX- its similar to html code but not HTML

NPM(Node Package Manager)

-----

1. NPM is distributed with Node.js which means that when you download Node.js , you

automatically get npm installed on your computer

2. Check if already NPMis installed

Ex- cmd> npm -v

3. Installation of Node.js LTS support version(long time support)

all package we can find into npmjs.com website

4. to update the npm

Ex- cmd> npm install npm@latest -g (gloably)

5. auto setup a project or existing project

Ex- npm init (it will generate the package.json file with the project structure)

a) create a directory for project

b) get the path of directory and run the npm init command (Shift + right click)

Package.json -> 1. we can give the package Name only when we want to publish the package

2. version when we need to publish

3. git reposotory url

6. npm configuration

npm gets its config setting from the command line, env variabls, npmrc files,

and in some cases, the package.json file.

Ex. cmd> npm config , npm config list, npm config list -l

7. setting the configuration using npm

Ex. npm config set init-author-name "Any Name"

in the package.json file automatic default value will set as auther name.

8. get the configuration using npm

Ex. npm config get init-author-name

9. edit the configuration using npm

Ex. npm config edit init-author-name , npm config edit -global(gloablly edit)

10. install the package like Jquery , reactjs, module

Ex. npm install lodash (get all the lodash related library we can using the command from npmjs website )

it will put the dependency into package.json file as dependency key and also it will create package-lock.json file(it will have specific version even though in future if any update version will come it wont reflact into lock.json file)

11. if want to publish the project without libraries folder as it is very big

Ex. cmd> node index --it will run throw error because required folder is not there.

Reson --> we have to install the pacakge first

Ex. cmd> npm install

Ex cmd> npm install lodash@1.23.1 (for specific version install)

EX. cmd> nmp install lodash@letest (using tag name)

conclusion

----------

1. if we install any package dependency should be there in both the file package.json as well package-lock.json file other wise it will throw the error if any other person will use after publish the project.

12. Uninstall or update the package

Ex. cmd> npm uninstall <packageName> --save or npm remove <packageName> --save or npm rm <packageName> --save

cmd> npm update <packageName> --save

13 if we want to see the doc before installing the package

Ex cmd> npm docs lodash --> it will open into web browser

14. to run npm start for default package

EX. into package.json file --> "script"--> "start":-->"file name as index file"

15. Install VS code

a) official website

b) install nodejs LTS

c) react developer tools- extantion

d) create React App for babel & webpack--> npx create-react-app my-app

short cut

----------

Ctrl+g - jump line no

Ctrl+Shft+t - reopen the recently closed tab

Ctrl+\ - split the editor

Ctrl+, go to setting

Alt+ down - move the line code down and up

Alt+Shft+down - copy the code

Ctrl+/ multiple line comment the code

16. create project using NPM

a) cmd> npx create-react-app my-app -->project is getting created

b) to start developement server--> go to cd my-app --> npm start--> it will give the local server with port number

c) prettier code formatter-- install

d) ES7 react/redux/graphQL/react-Native snippets --install

17. read the link from public folder using webpack because webpack never read the data from public folder

Ex. we can use this line of code to read <link rel ="shortcut icon" href ="%PUBLIC\_URL%/favicon.ico">

Ex. if we have more image file then we can keep those images into public folder but if only less image is there then we can keep into src folder

npm start --to start server

npm run build - when we build the project

npm test - when we test the project

Components

-----------

1. Defination of components

a) components are reusable , and also we nested the components i.e one components can have another component.

b) components describe a part of user interface

2. Type of component

a) stateless funtional component

b) stateful class component

3. creting every component need to import the react when we creating a component using funtion

Ex. import React from 'react'

4. and also need to export the component which we created

Ex. export default Hello

a) we can export in two ways 1. name export 2. default export(we can use any name while importing the components

class component

1. we need to import two things

Ex. import React, {Component} from 'react'

2. need to extends the Component

Ex. class Welcome extends Component{ render(){}}

3. Funtional Vs Class components

a) simple funtions

b) solution without using state

c) Absence of 'this' keyword

d) Use Func components as much as possible

e) stateless/dumb/presentational

f) using ES6 const student=()=> return <h1> hello</hello> or funtion student(){ return <h1> hello</h1>

g) funtion student(props){ return <h1>hello {props.name}</h1>

1) More feature rich

2) Maintain their own private data- state

3) complex UI logic

4) provide lifecycle hooks

5) stateful/smart/container

6) when we use class component we have to use props using this keyword(this.props) because it point current address

7) class student extends Component{ render(){ return <h1>hello</h1>}} -->without props

8) class student extends Component{ render(){ return <h1>hello {this.props.name}</h1>}} -->using props

4. New feature React V16.7.0

1. Hooks

2.

5. JSX -

a) JavaScript XML

b) write XML-like code for element and components

c) JSX tags have a tag name , attributes and children

d) JSX ultimati

7. Composing Components -> Component can refer to other components in their output

Note: read about redux, props, component,state

6. render() ->The render() method is the only required method in a class component, it examines this.props and this.state

a) Arrays and fragments -> it is used to return multiple elements from render

b) portals - it is used to render children into a diffrent DOM subtree

c) string and number - these are rendered as text nodes in the DOM

d) booleans or null - it renders nothing

e) creating a element - React.createElement(type, props, children);--> we can create more nested element into children for that onlt we use JSX

f) ReactDOM.render(element, DOMnode); -> it takes a React element and render it to a DOM node

7. Fragment -> It used to group a list of children without adding extra nodes to the DOM.

1) <React.Fragment>

2) <React.Fragment key={item.id}> <h1>{item.title}</h1> <p>{item.description}</p> </React.Fragment>

8. Specifying Attributes with JSX

1. const e1=<h1 attribute="value"></h1>

2. className

3. htmlFor

Note: when ever we write any JSX code Babel will compile and convert into normal js code and

9. Object creation using JSX

Ex. const e1={

type:'h1',

props:{

className:'bg'

children:'hello'

}

};

10. Props ->

Whether you declare a component as a function or a class, it must nevr modify its own props.

ALl React component must act like pure funtions with respect to their props

Ex. Pure Funtion

funtion sum(a,b){

return a+b;

}

Ex. Impure Funtion

funtion withdraw(account, amount){

acount.total-=amount; //modifying the props value , it should not recommended to change the props because it read only

}

11. Type checking with props type

1. go to npmjs.com website

2. search for prop-type -> install info

3. we can import or use CDN link or we can install the package

Ex. cmd> npm install prop-types --> to install the package

How to import

Ex. import PropType from 'prop-types';

How to use

Student.propType={

name.propTypes.string

name.propTypes.string.isRequired --> if any prop which is required

}

Default value for PropType

Student.defaultProps={

name:"Default Value"

}

Note: npmjs.com for more prop-types

12. {this.props.children} --> we can use while rendering the component we can close the Student tag using closing node and in between we can provide any vlue , and that value we can access as children.

13 Install React Router

Ex. npm install react-router-dom

To import router

1. impo

14. to install axios for HTTP request

Ex. npm install axios

15. React Hooks

1. Why Hooks? -> more related to java script rather then react itself

-> reuse statefull logic without changing componet hirerchy

-> data fetching componentDidMount and componentDidUpdate

-> Even listeners componentDidMount adn componentWillUnmount

2. Use -> V16.8 or higher

3. feature ->cant use hooks inside of a class component

4. Rules -> only call hooks at top level , only call hooks form react funtions , dont call inside conditions or nested funtions

16. Redux in reactJs

Ex. npm install redux --save

Ex. npm install redux-logger --save

Ex. npm install react-redux --save and import as Provider(it work as bridge because both react and redux are standlone)

4 steup for Redux

a) Store

1. reducer

2. state

b)Reducer

1. state

2. action

c) Subscribe

d) Dispatch (some action)

Redux middleware?

-> It is a good library

->

1.

If use react and redux together then separation of concern is there. Means, there are two types of components.

Dumb Components(Presentational)

Smart Component(Containers)

Dumb components render the data, and they did not care about any logic. They have nothing to do with a redux store.

Smart components are concern about logic and directly connected to the store.

https://jsonplaceholder.typicode.com/users

https://alligator.io/react/axios-react/

https://github.com/Pau1fitz/react-spotify

https://www.youtube.com/watch?v=OSSpVLpuVWA

reactive form :eslint

\_https://www.npmjs.com/package/react-reactive-form

Design Documentation UI Design Componets

Actions

Reducers

Services

BE Design Handlers

Plugins

DTO

Mappers

React Hooks

-----------------

React version - 16.8

what - without writing a class we can create the component that call react hooks

why - reason 1

need to understand the this keywork works in java script,

remember how to bind event handler in class component

class component dont minify very well and make hot reload very unreliable

reason 2