Life cycle of react js

<https://www.codingame.com/playgrounds/8747/react-lifecycle-methods-render-and-componentdidmount>

import React from "react";

import ReactDO from "react-dom";

// const App = () => {

// return <div>Hi Hello </div>;

// };

class App extends React.Component {

constructor(props) {

super(props);

this.state = { lat: null,ErrorMessage:'' };

window.navigator.geolocation.getCurrentPosition(

position => {

this.setState({ lat: position.coords.latitude });

},

err => {

this.setState({

ErrorMessage: err.message

});

}

);

}

render() {

if (this.state.ErrorMessage && !this.state.lat) {

return <div>ErrorMessage:{this.state.ErrorMessage}</div>;

}

if (!this.state.ErrorMessage && this.state.lat) {

return <div>latitude:{this.state.lat}</div>;

}

return <div>loading</div>;

}

}

ReactDO.render(<App />, document.querySelector("#root"));

Creating object and passing value for object at run time after Extract that object

const seasonConfig = {

summer: {

text: "beach",

iconNmae: "sun"

},

winter: {

text: "chil climate",

iconNmae: "snowflake"

}

};

const season = getSeason(props.latitude, new Date().getMonth());

const {text, iconNmae}= seasonConfig[season];

<i className={`${iconNmae} icon`} />

<h4>{text}</h4>

Defaultprops:

Spinner.defalutProps = {

message: "loading......"

};

const Spinner = (props)=>{

return (

<div>

<div class="ui active dimmer">

<div class="ui text loader">{props.message}</div>

</div>

</div>

);

TypeError: Cannot read property 'state' of undefined

onFormSubmit(event){

event.preventDefault();

console.log(this.state.term);

}

After preventDefault();

Using preventdefalu we can’t use this keyword

onFormSubmit=(event)=>{

event.preventDefault();

console.log(this.state.term);

}

<form onSubmit={this.onFormSubmit} className="ui form">

Request the Api and return the response:

onSearchSubmit(term){

axios.get('https://api.unsplash.com/search/photos', {

params: {

query: term

},

headers: {

Authorization:

' 3963ddc3e99a11241a747d513734b635de9279f8a3398ec5aa665326016825bf'

}

}).then((response)=>{

console.log(response.data.results);

(Or)

});

console.log(term);

}

Normal java script map() function:

var array1 = [1, 4, 9, 16];

// pass a function to map

const map1 = array1.map(x => x \* 2);

console.log(map1);

// expected output: Array [2, 8, 18, 32]

const images= props.images.map((image)=>{

return <img src={image.url.regular}/>

});

Defraction objects:

const images= props.images.map((image)=>{

return <img alt={image.description} key={image.id} src={image.url.regular} />;

});

To

const images = props.images.map(({ description, id, urls})=>{

return <img alt={description} key={id} src={urls.regular} />;

});

To solve 400 error install this

npm install [axios@0.18.0](mailto:axios@0.18.0)

Instead of getting props directly we can get the props variable

In child call back function

<VideoList videos={this.state.videos}/>

const VideoList = ({ videos})=>{

return (

<div>

<div>VideoList</div>

<div>{videos.length}</div>

</div>

);

}

Redux

Const ourDeparment=combineReducers({

Accounting:Accounting,

claimHistory:claimHistory,

policy:policy

});

Cost store=createstore(ourDeparment)

export default combineReducers({

song: songReducer,

selectedsong: selectedsongReducer

});

const mapStateToProp = state => {

console.log(state);

return { songs: state.song};

};

Asyn with await

export const fetchedData= async () =>{

const response = await JsonPlaceholder.get('/posts')

return{

type:'FETCHED\_DATA'

};

Thunk middleware in redux:

return async function(dispatch,getState){

const response = await JsonPlaceholder.get('/posts')

dispatch({ type: "FETCHED\_DATA", payload: response });

// return {

// type: 'FETCHED\_DATA',

// payload: response

// };

}

javaScript find method:

const user = this.props.users.find(user=>user.id===this.props.userId

mapStateToProp(state,ownpro) and destructure:

const mapStateToProps = (state,ownProps) => {

return {

user: state.users.find(user => user.id === ownProps .userId

) };

}

const {user} =this.props;

Db.json server (own l;ocal json server )