Éléments de correction

Méthode 1 :

import React, { Component } from "react";

import { Message } from "semantic-ui-react";

import Recherche from "./components/Recherche";

class App extends Component {

state = {

data: '',

error: '',

crypto: '',

monnaie: '',

resultat: ''

}

onSearch = async (crypto, monnaie) => {

if (crypto && monnaie) {

try {

let response = await fetch(`https://api.coingecko.com/api/v3/simple/price?ids=${crypto}&vs\_currencies=${monnaie}`);

let donnees = await response.json();

this.setState({

data: donnees,

error: '',

crypto: crypto,

monnaie: monnaie,

resultat: Object.values(Object.values(donnees)[0])[0]

})

}

catch (e) {

this.setState({ error: "Erreur lors de la recherche" });

}

}

else {

this.setState({ error: "SVP Choisisez un crypto et une monnaie" });

}

}

onEmpty = () => {

this.setState({

data: '',

error: '',

crypto: '',

monnaie: '',

resultat: ''}

)

}

render() {

return (

<div className="App">

<h1>Convertisseur crypto</h1>,

<Recherche onSearch={this.onSearch} onEmpty={this.onEmpty} />

{this.state.error ? <Message warning>Erreur:{this.state.error}</Message> : undefined}

{this.state.data ? <Message warning> 1 {this.state.crypto} = {this.state.resultat} { this.state.monnaie} </Message> : undefined}

</div>

);

}

}

export default App;

Méthode 2 :

import React, { Component } from 'react';

import { Button, Container, Message, Select } from 'semantic-ui-react';

import '../Recherche.css';

class Recherche extends Component {

state = {

source: '',

destination: '',

data: '',

error:''

}

onSourceChange = (e, data) => {

this.setState({

source: data.value

});

}

onDestinationChange = (e, data) => {

this.setState({

destination: data.value

})

}

onAppelApi = async () => {

if (this.state.source && this.state.destination) {

try {

let response = await fetch(`https://api.coingecko.com/api/v3/simple/price?ids=${this.state.source}&vs\_currencies=${this.state.destination}`);

let donnees = await response.json();

this.setState({

data: donnees,

error:''

})

console.log(this.state.data);

}

catch (e) {

this.setState({error:'APPEL API échoué'})

}

}

else {

this.setState({error:' il faut choisir la source et la destination'})

}

}

onVider = () => {

this.setState({

source: '',

destination: ''

});

}

render() {

console.log(this.state.source, this.state.destination);

const optionsSource = [

{ key: 'af', value: 'bitcoin', text: 'Bitcoin' },

{ key: 'ax', value: 'ethereum', text: 'Ethereum' },

{ key: 'al', value: 'cardano', text: 'Cardano' },

]

const optionsDestination = [

{ key: 'af', value: 'cad', text: 'Dollard Canadien' },

{ key: 'ax', value: 'usd', text: 'Dollard Americain' },

{ key: 'al', value: 'eur', text: 'Euro' },

]

return (

<Container >

<h1>Conversion de crypto monnaie</h1>

<div className="recherche">

<Select onChange={this.onSourceChange} placeholder="Choisissez le crypto" options={optionsSource} />

<Select onChange={this.onDestinationChange} placeholder="Choisissez la destination" options={optionsDestination} />

<Button onClick={this.onAppelApi} primary> Convertir </Button>

<Button onClick={this.onVider} secondary> Vider</Button>

</div>

{this.state.data [this.state.source] ? <Message> 1 {this.state.source} = {this.state.data[this.state.source][this.state.destination]} {this.state.destination}</Message>:undefined}

{this.state.error? <Message>{this.state.error}</Message>:undefined}

</Container>

);

}

}

export default Recherche;