Cheat Sheet: In-depth Understanding of Advanced React Functionality

Hooks and form management	Description	Code example

useState()	`useState()` hook can manage states of the React function component where you can declare any data type, for example, boolean, object, array, string.	`import React, { useState, useEffect } from 'react';`
		`function SideEffect() {`
		`  const [empId, setEmpId] = useState(100);`
		`  return (`
		`  <div>`
		`  <p>{empId}</p>`
		`  </div>`
		`  );`
		`}`
		`export default SideEffect;`

useEffect()	`useEffect` is a React hook that allows you to perform side effects in functional components. A side effect refers to any operation that you need to execute as soon as the page loads without calling these operations/functionalities separately, such as fetching data from an API.	`import React, { useState, useEffect } from 'react';`
		`function SideEffect() {`
		`  const [foods, setFoods] = useState([]);`
		`  useEffect(() => {`
		`  fetch('https://api.npoint.io/d5429d9d99f581ab3db9')`
		`  .then(response => response.json())`
		`  .then(data => {`
		`  console.log(data);`
		`  setFoods(data);`
		`  })`
		`  .catch(error => console.error('Error fetching users:', error));`
		`  },[]); // Empty dependency array means this effect runs only once when the component mounts`
		`  return (`
		`  <div>`
		`  <h3>Food List</h3>`
		`  <ul>`
		`  {foods.map((food)=>{`
		`  return (<>`
		`  <li><h1>{food.name}</h1></li>`
		`  <p>Food.description</p>`
		`  <p>Food.price</p>`
		`  <p>Food.category</p>`
		`  <p>Food.ingredients</p>`
		`  <img src={food.image_url} alt="" height='100px' width='100px' />`
		`  </>`
		`  )`
		`  })`
		`  </ul>`
		`  </div>`
		`  );`
		`}`
		`export default SideEffect;`

Custom hook	You can use custom hooks in any other component. In this code snippet, there is one function component known as `UseToggle`, which serves as a custom hook, and another function component `ToggleButton`, which will use this custom hook.	`//ToggleButton`
		`import { useState } from 'react';`
		`import UseToggle from './UseToggle';`
		`function ToggleButton() {`
		`  const [toggled, toggle] = UseToggle(false);`
		`  return (`
		`  <div>`
		`  <h3>Toggle Button</h3>`
		`  <button onClick={toggle}>`
		`  {isToggled ? 'ON' : 'OFF'}`
		`  </button>`
		`  </div>`
		`  );`
		`}`
		`export default ToggleButton;`
		`//UseToggle.jsx`
		`import { useState } from "react";`
		`function UseToggle(initialValue = false) {`
		`  const [value, setValue] = useState(initialValue);`
		`  const toggle = () => {`
		`  setValue(!value);`
		`  };`
		`  return [value, toggle];`
		`}`
		`export default UseToggle`

fetch api method	Fetch method can fetch data using API.	`const apiUrl = 'https://jsonplaceholder.typicode.com/posts';`
		`fetch(apiUrl)`
		`  .then(response => response.json())`
		`  .then(data => {`
		`  console.log(data);`
		`  })`
		`  .catch(error => {`
		`  console.error('There was a problem with the fetch operation:', error);`
		`  });`

axios api method	Axios method can fetch data using API.	`import axios from 'axios';`
		`const apiUrl = 'https://jsonplaceholder.typicode.com/posts';`
		`axios.get(apiUrl)`
		`  .then(response => {`
		`  console.log(response.data);`
		`  })`
		`  .catch(error => {`
		`  console.error('There was a problem with the fetch operation:', error);`
		`  });`

onChange	The onChange event attribute is often used in HTML and React to track when the value of an input field changes, like a text input. The onChange event occurs when a user writes something into an input field. This attribute lets you record and handle the changes.	`import React, { useState } from 'react';`
		`function FormData() {`
		`  const [empName, setEmpName] = useState('');`
		`  const handleChange = event => {`
		`  setEmpName(event.target.value);`
		`  };`
		`  const handleSubmit = event => {`
		`  event.preventDefault();`
		`  console.log('Form submitted:', empName);`
		`  };`
		`  return (`
		`  <div>`
		`  <h2>My Form</h2>`
		`  <form onSubmit={handleSubmit}>`
		`  <label>`
		`  Input:`
		`  <input type="text" value={empName} onChange={handleChange} />`
		`  </label>`
		`  <button type="submit">Submit</button>`
		`  </form>`
		`  </div>`
		`  );`
		`}`
		`export default FormData;`

Redux toolkit	Redux toolkit can be installed using npm	`npm install @reduxjs/toolkit.`