Introducción a React.Js

Módulo 05

Introducción a los Hooks

Durante este curso estuvimos utilizando distintos componentes para construir nuestra interfaz. Seguramente habrás notado que los componentes basados en clases son, por así decirlo, más completos (podemos gestionar el estado, podemos hacer llamadas a API).

Sin embargo, los componentes funcionales procuran un código mucho más limpio y ordenado.

¿Existe la forma de aprovechar la limpieza de código de los componentes funcionales y las características de los componentes basados en clases?

Introducción a los Hooks

Los “hooks” o “ganchos” de React permiten justamente eso. Los Hooks son funciones de React que permiten a un componente funcional “engancharse” a funcionalidades propias de un componente basado en clases.

Podemos acceder al estado de React y hacer llamadas a APIs desde un componente funcional.

Veamos estos dos casos

Introducción a los Hooks

```
1  import React, { useState } from "react";
2
3  export default function SubMenu(props) {
4      const visibleState = useState(false);
5      const getVisibleState = visibleState[0];
6      const setVisibleState = visibleState[1];
7
8      let panel = null;
9
10     if (getVisibleState) {
11         panel = <div>Hola</div>;
12     }
13
14     return (
15         <div>
16             <div onClick={() => setVisibleState(!getVisibleState)}>
17                 Haceme click
18             </div>
19             {panel}
20         </div>
21     );
22 }
```

1 - Los Hooks son una función nativa de React, por lo que debemos importarla. En este ejemplo usamos *useState*, que permite acceder al estado de React.
4 - La función useState recibe un valor. Ese valor será el valor inicial de esa porción del estado. Dicha función retorna un array. El primer elemento es el getter de esa porción, mientras que el segundo es el setter.
10 y 16 a 19 - En estas líneas utilizamos esta porción del estado para mostrar u ocultar un panel. El get se comportará igual que un this.state... de una clase; y el set se comportará igual que un setState. En pocas palabras, cada vez que cambie el valor (se llame a setVisibleState) ocasionará un re-render del componente.

Introducción a los Hooks

1 - En esta ocasión, utilizaremos el Hook *useEffect*. En React, muchos eventos del ciclo de vida (como componentDidMount) se usan para los llamados *efectos secundarios*. Estos “efectos secundarios” son todas las operaciones asincrónicas o tareas secundarias que van por fuera del componente, pero pueden afectar su estado. Son, justamente, “efectos secundarios” de este componente. El Hook *useEffect* condensa todos los eventos del ciclo de vida para el manejo de operaciones asincrónicas en una sola función (Entre ellas, llamadas a APIs)
5 - Otra sintaxis para el useState. Es necesario este Hook, sino ¿Donde guardamos la información de la API?
7 a 11 - useEffect es una función que recibe un callback. Al cargarse el componente React se crea una tarea “pendiente”, un “efecto”, que no bloquea el renderizado del componente. Ese efecto se ejecutará y podrá cambiar el estado cuando sea necesario (linea 10)

```
1  import React, { useState, useEffect } from "react";
2
3  export default function ListadoClientes(props) {
4      // Otra sintaxis
5      const [clientes, setClientes] = useState([]);
6
7      useEffect(() => {
8          fetch("/clients")
9              .then((r) => r.json())
10             .then((d) => setClientes(d));
11     });
12
13     return (
14         <div>
15             {clientes.map((cliente) => (
16                 <div>
17                     <h2>{cliente.nombre}</h2>
18                     <h4>{cliente.alta}</h4>
19                     <hr />
20                 </div>
21             ))}
22         </div>
23     );
24 }
```

Introducción a los Hooks

Los “hooks” son un gran tema de React. Puedes mantenerte al tanto en la documentación oficial de React.

https://es.reactjs.org/docs/hooks-intro.html

También este es un gran tema que se trata en el Curso de React Avanzado

Puedes, además, encontrar mucha información en la red.

¡Te deseamos muchos éxitos en tu carrera!

¡Muchas gracias!

¡Sigamos trabajando!