

React: Hacer fetch al servidor

Importaremos los módulos **useState** y **useEffect** del paquete **'react'**. Definimos una variable de estado, que será donde guardaremos la información que nos devuelve el servidor/api.

```
import React, {useState,useEffect} from 'react';

function App(){
let [nombre, setNombre] = useState('');

}
```

useEffect es una función que se ejecuta, como mínimo, cuando se pinta el componente por primera vez. A esta función le pasaremos dos parámetros: una función y un array. Dentro de la función pondremos el código que queremos que se ejecute. Dentro del array pondremos una variable de estado que será la que diga a **useEffect** si tiene que volver a ejecutar o no: cuando esa variable cambie de valor se ejecutará **useEffect**, si no cambia no se ejecutará.

```
useEffect(function () {
    //escribimos nuestro código
}, [nombre])
```

Al array del **useEffect** le pasamos la variable de estado **nombre**, por lo que cada vez que cambie su valor se ejecutará **useEffect** otra vez. Si cambia cualquier otra variable de estado, o **nombre** no cambia, **useEffect** no se ejecutará aún y cuando el componente se re-renderice.

Si queremos que un **useEffect** solo se ejecute una vez (la primera vez que se pinta el componente), pondremos un array vacío.

```
useEffect(function () {
    //escribimos nuestro código
}, [])
```

Sabiendo esto, podremos hacer un fetch al servidor al cargar el componente. El fetch lo haremos de la misma manera que lo hemos estado haciendo hasta ahora:

```
useEffect(function () {
```

```
    fetch('ruta')
      .then(respuesta =>
        respuesta.json()
      )
      .then(datos => setNombre(datos))
  }, [])
```

Estamos asignando a la variable de estado **nombre** los datos que nos envía el servidor/API. Y ya podremos trabajar y modificar el valor de la variable de estado para lo que queramos.