Componentes mutables (práctica)

Tiempo estimado: 15min

El objeto de esta práctica es afianzar, reforzar y consolidar los conocimientos teóricos presentados en la lección.

Al finalizarla, el estudiante:

- Habrá creado un componente mutable.
- Habrá actualizado el estado de un componente mutable.
- Habrá refrescado la representación de un componente mutable.

Objetivos

El objetivo de esta práctica es crear un componente que muestre una tabla de datos mutable. Cada 3 segundos, una función va añadiendo una nueva fila de datos a su estado con objeto de ver cómo se actualiza su representación.

Por un lado, tenemos el componente mutable que representa la tabla. Y por otro lado, el inmutable que representa cada fila.

Preparación del entorno

Para comenzar, crearemos un proyecto de aplicación React mediante el generador de Justo:

1. Abrir una consola.
2. Crear el directorio de la práctica e ir a él.
3. Invocar el generador de Justo:

   ```
   > justo -g react
   ```
4. Responder a las preguntas realizadas por el generador, teniendo en cuenta:
   - Responder con N a la pregunta sobre si usar React Router.
5. Instalar las dependencias del proyecto:

   ```
   > npm install
   ```
6. Mostrar el catálogo de tareas automatizadas del proyecto:

   ```
   > justo -c
   ```
7. Invocar la tarea build del catálogo del proyecto para construir la aplicación:

   ```
   > justo build
   ```
8. Abrir el archivo dist/index.html con el navegador.

   Debe aparecer el mensaje Hello World!

Creación de la constante de datos

Primero, vamos a crear los datos de la tabla:

1. Crear el archivo app/data/bands.js:

   ```
   export default [
     {
       name: "La Dama se Esconde",
       year: 1985,
       origin: "San Sebastián"
     },
     {
       name: "Lori Meyers",
   ```



```
    year: 1998,
    origin: "Loja, Granada"
  },
  {
    name: "La Habitación Roja",
    year: 1994,
    origin: "L'Eliana, Valencia"
  },
  {
    name: "The Sunday Drivers",
    year: 1999,
    origin: "Toledo"
  },
  {
    name: "Second",
    year: 1997,
    origin: "Murcia"
  }
];
```

Creación del componente inmutable

Ahora, vamos a crear el componente inmutable que tiene como objeto mostrar una fila de datos:

1. Ir a la consola.
2. Crear el componente mediante el generador de Justo: inmutable, simple, mediante función y con el nombre Band.

```
> justo -g react component
```

3. Editar el archivo app/components/Band.jsx y con su función de reproducción como sigue:

```
export default function Band(props) {
  return (
    <tr>
      <td>{props.name}</td>
      <td>{props.year}</td>
      <td>{props.origin}</td>
    </tr>
  );
}
```

4. Guardar cambios.

Creación del componente mutable

A continuación, vamos a crear el componente mutable que tiene como objeto mostrar las filas de datos. Lo consideramos mutable porque su contenido se irá generando poco a poco y deseamos mostrárselo al usuario a medida que los cambios se producen.

1. Ir a la consola.
2. Crear el componente mediante el generador de Justo: mutable, simple y con el nombre Bands.

```
> justo -g react component
```

3. Editar el archivo app/components/Bands.jsx.
4. Definir el estado inicial. Los datos de las filas se pasarán a través de la propiedad rows:

```
constructor(props) {
  super(props);

  //initial state
  this.state = {
    rows: this.props.rows || []
  };
}
```

5. Importar el componente Band:

```
//imports
import React from "react";
```



```
import Band from "./Band";
```

6. Definir la representación del componente:

```
render() {
  return (
    <table>
      <thead>
        <tr>
          <th>Banda</th>
          <th>Año</th>
          <th>Origen</th>
        </tr>
      </thead>

      <tbody>
        {
          this.state.rows.map(function(row) {
            return <Band key={row.name} {...row} />;
          })
        }
      </tbody>
    </table>
  );
}
```

7. Guardar cambios.

Modificación del estado y actualización de reproducción

Antes de finalizar, hay que instanciar el componente mutable e ir cambiando su estado cada 3 segundos añadiéndole una fila cada vez:

1. Editar el archivo app/views/App.jsx.
2. Importar el componente Bands y los datos:

```
//imports
import React from "react";
import Bands from "../components/Bands";
import rows from "../data/bands";
```

3. Definir el método render() como sigue:

```
render() {
  return <Bands ref="bands" rows={rows.slice(0, 1)} />;
}
```

Supongamos que la tabla inicialmente sólo tiene una fila de datos.

La propiedad ref es especial de React. La veremos detenidamente en una lección posterior. En resumen, es una propiedad que sirve para dar un nombre único a la instancia componente a partir del cual obtenerla fácilmente, tal como veremos a continuación.

4. Definir el método componentDidMount() como sigue:

```
componentDidMount() {
  const self = this;
  var last = 1;

  addBandToStateDynamically();

  //helper
  function addBandToStateDynamically() {
    if (last < rows.length) {
      setTimeout(function() {
        self.refs.bands.setState({rows: rows.slice(0, ++last)});
        addBandToStateDynamically();
      }, 3000);
    }
  }
}
```

5. Guardar cambios.

6. Ir a la consola.
7. Reconstruir la aplicación:

```
> justo build
```

8. Abrir dist/index.html con el navegador.
9. Esperar a que las filas se vayan mostrando, unos 15 segundos en total.

