Desarrollo Web Mayo- Agosto 2024 Ing. Ana Sol Arteaga



Introducción al desarrollo Web con Ant Design **Table y getProducts**

git branch getProducts

Como primer paso establece el endpoint *PRODUCTS: "api/products"*, en tu archivo de constantes.

En la carpeta de **services**, crea el archivo **products.js** para crear las solicitudes al endpoint establecido en la instrucción anterior.

Realiza los imports necesarios (axios, ENV) y construye la primera solicitud con el método get para obtener todos los productos.

Para continuar, en tu componente Productos (pages/index.jsx) implementa la vista, yo utilice el componente Table de Antd, ve a su documentación y observa los imports necesarios para una tabla de estructura básica con un select y paginación incluida. Importa el componente *import { Divider, Table } from 'antd';* e incluye una constante comlumns que será un arreglo de objetos que servirá para nombrar cada una de las columnas de la tabla, esto lo haremos de acuerdo a los parámetros que obtengamos de la solicitud. Aquí tienes un ejemplo, pero recuerda que la estructura depende de las características de API y los datos que quieras mostrar. El render se usa en la columna de imgURL porque necesitas renderizar una imagen en lugar de solo mostrar una URL.



También incluiremos la función rowSelection para seleccionar cada fila, esta viene directo con el componente de Antd.

```
// rowSelection object indicates the need for row selection
const rowSelection = {
    onChange: (selectedRowKeys, selectedRows) => {
        console.log(`selectedRowKeys: ${selectedRowKeys}`, 'selectedRows: ', selectedRows);
},

getCheckboxProps: (record) => ({
        disabled: record.name === 'Disabled User',
        // Column configuration not to be checked
        name: record.name,
}),
};

};
```

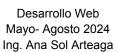
Para comenzar con la solicitud, no te olvides de importar la función getProducts *import {* getProducts } from '../../services/products'; dentro del componente crea el estado para products (incluye selectionType de antd).

Construye la estructura de useEffect que permite que la solicitud se realice una vez se cargue el componente, y no siga rehaciendo la solicitud, creamos la constante **data** que almacenará la respuesta de getProducts de manera asíncrona, se utiliza el método map para iterar sobre cada producto.

En esta iteración, se crea un nuevo objeto "producto" que incluye todas las propiedades del producto original (...product) y se añade una nueva propiedad key que se establece como el valor de _id del producto y se actualiza el estado.

Por último manejamos el error y ejecutamos esta función.

```
const Productos = () => {
   const [products, setProducts] = useState([]);
   const [selectionType, setSelectionType] = useState('checkbox');
   useEffect(() => {
        const fetchProducts = async () => {
                const data = await getProducts();
                const productsWithKey = data.map(product => ({
                    ...product,
                    key: product._id,
                }));
                setProducts(productsWithKey);
             catch (error) {
                console.error('Error al obtener los productos', error);
        };
        fetchProducts();
      []);
```





Ahora si en nuestro componente Table podemos pasarle las propiedades columns y datasource con los valores que acabamos de construir.

Observa que se añade la propiedad scroll cuando la tabla llega a una altura de 400. Los demás componentes que se muestran son parte de la UI.

```
return (
        <div>
            <Nav />
            <CrudButton />
            <Divider />
            <div className="products-container">
                <Table
                    rowSelection={{
                        type: selectionType,
                         ...rowSelection,
                    columns={columns}
                    dataSource={products}
                    scroll={{ y: 400 }}
            </div>
        </div>
export default Productos;
```

Por ejemplo, el componente CrudButton renderiza los botones necesarios para implementar el Crud.

Al final obtendrás una vista como esta.

