Desarrollo Web
Mayo- Agosto 2024
Ing. Ana Sol Arteaga

Introducción al desarrollo Web con Ant Design

Creación de rutas

Para crear las rutas iniciales de nuestro proyecto, es necesario instalar la siguiente dependencia *npm install react-router-dom* e importar en el componente App ***useRoutes y BrowserRouter,*** existen varias maneras de crear las rutas en un proyecto con react pero para este ejemplo lo haremos creando un componente AppRoutes, dentro un arreglo de objetos que almacene todas las rutas que queremos añadir.

En la estructura de tu proyecto crea src/routes/index.jsx

```
src > routes > index.jsx > AppRoutes
1   import React from 'react';
2   import { useRoutes } from 'react-router-dom';
3   import Home from '../pages/Home';
4   import Login from '../pages/Login';
5
6   const AppRoutes = () => {
7       let routes = useRoutes([
8           { path: '/login', element: <Login /> },
9           { path: '/', element: <Home /> },
10      ])
11      return routes;
12  }
13
14  export default AppRoutes;
```

Para poder usar este componente desde cualquier componente de nuestra aplicación, debemos envolver el componente que creamos dentro de ***BrowserRouter*** en el componente App.

```
<BrowserRouter >
  <AppRoutes />
</BrowserRouter >
```

Prueba tus rutas desde la barra de navegación y verifica que sean funcionales, para probar que las rutas funcionan al accionar un evento, por ejemplo al darle clic en login vamos a crear una función dentro del componente FormLogin, dentro de un objeto asigna valores por defecto para user

```
const user = {
    username: 'admin',
    password: 'admin',
};
```

Para usar la navegación en este componente, es necesario importar ***import { useNavigate } from 'react-router-dom';*** desde la librería ya instalada y crear una constante ***const navigate = useNavigate();***

Para el manejo del error del formulario, vamos a crear un estado, para eso imposta useState desde react en tu componente ***import React, { useState } from 'react';*** el estado lo debemos inicializar en falso

```
// Estado para el error de login
const [loginError, setLoginError] = useState(false);
```

Para validar el formulario con los valores estáticos vamos crear dos funciones onFinish y onFinishFailed, para la función onFinish primero válida que los datos ingresados por el usuario coincidan con los valores asignados por defecto, en caso de que esta condición se cumpla usamos la propiedad ***navigate*** pasándole por parámetro el path de la ruta a la que se va a navegar. Sí esta condición no se cumple, llamar a la función **onFinishFailed** que por ahora solo manda el mensaje de error en la consola y actualiza el valor del estado.

```
17      //Función para mostrar errores en el formulario
18      const onFinishFailed = (errorInfo) => {
19          console.log('Failed:', errorInfo);
20          setLoginError(true);
21
22      };
23      //Función para validar el usuario y contraseña
24      const onFinish = (values) => {
25          const { username, password } = values;
26          if (username === user.username && password === user.password) {
27              navigate('/');
28          } else {
29              onFinishFailed();
30          }
31      };
```

Por último, antes del botón que envía el formulario, coloca una condicional en caso de error en el login, con esto evalúa el valor del estado y devuelve un texto con el mensaje especificado cuando algo falla.

{loginError && <p style={{ color: 'red' }}>Credenciales incorrectas. Inténtalo de nuevo.</p>}

Configura todas las rutas necesarias para el proyecto (hasta ahora).