

Primeros pasos React Router V6.4+

Por: Angel Mouzaber



Instalación

```
yarn add react-router-dom
```

```
npm install react-router-dom
```

```
pnpm install react-router-dom
```

Creando el archivo de Rutas (Estructura)

```
// Router.jsx

const router = createBrowserRouter([
  { path: "/home", element: <HomePage /> }
  ...
]);

export const Router = () => {
  return <RouterProvider router={ router } />
}
```

Añadiendo el Browser Router

```
// main.jsx

import { Router } from './router/Router';
ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <Router />
  </React.StrictMode>
)
```



Componentes

Link:

Un **<Link>** es un elemento que permite al usuario navegar a otra página haciendo clic o tocándola.

```
import { Link } from "react-router-dom";
export const ComponenteX = () => {
  return (
    <Link
      to="auth/login" // Ruta a la que se desea ir
    >
      Login // Texto a mostrar en el link
    </Link>
  )
}
```

NavLink:

Un **<NavLink>** es un tipo especial de **<Link>** que sabe si está o no **"active"**.

```
import { NavLink } from "react-router-dom";
<NavLink
  className={({isActive}) => `nav-item nav-link${isActive ? ' active' : ''}`}
  to="/home"
>
  Home
</NavLink>
```

useNavigate:

El Hook **useNavigate** retorna una función que te permite navegar entre rutas de manera dinámica. Por ejemplo, en un **useEffect**.

```
import { useNavigate } from "react-router-dom";

const navigate = useNavigate();
useEffect(() => {
  if (userIsInactive) {
    fake.logout();
    navigate("/session-timed-out");
    // Acá hace la navegación hacia la ruta establecida
  }
}, [userIsInactive]);
```



Outlet:

El `<Outlet />` se debe de usar en el `element` de la ruta padre para renderizar los elementos de sus rutas hijas. Esto se utiliza para hacer Layouts, por ejemplo.

```
export const PrivateRoute = ({ children }) => {
  const { state } = useContext(AuthContext);
  return (state.logged)
    ?
    <HeroesPage>
      <Outlet />
    </HeroesPage>
    :
    <Navigate to="/login" />
}
```

useLocation:

El Hook `useLocation()` retorna un objeto que corresponde a la ubicación actual, al URL actual. Se puede acceder a las propiedades como `pathname` y `search`.

```
{
  pathname: '/rutaprotegida/acceso',
  search: '?q=haciendo+una+busqueda'
}
```

useParams:

El Hook `useParams` retorna un objeto cuya key corresponde a la variable dinámica definida en la ruta y su value corresponde al valor exacto de la ruta dinámica.

```
// DEFINICIÓN DE LA RUTA
{ path: 'hero/:heroId', element: <Hero /> }

//URL ACTUAL
"hero/marvel-spider"

//OBTENIENDO PARAMS
const params = useParams();
console.log(params); // { heroId: marvel-spider }
```



Gist simplificado



Gist detallado



Documentación



Más recursos

