

Démo sur React Router



Développer une application React de **"Gestion des produits pharmaceutiques"** :

- La racine affiche une page Login ("/login") avec un bouton **"Connexion"** qui renvoi directement sur la page d'accueil **"/home"** au click;
- La page **"/home"** affiche tout simplement un message de bienvenu avec un système de navigation fonctionnel;
- La page **"/produits"** liste différents produits avec un lien **"voir plus"** qui ramène à la description de ce produit;
- Et la page **"/about"** qui donne une description l'application.

- Comprendre les concepts fondamentaux de React Router;
- Comprendre la différence entre les routes statiques et dynamiques;
- Créer un menu de navigation avec des liens vers différentes pages;
- Utiliser les paramètres de route pour personnaliser les liens du menu de navigation.

1. Création du projet gestion-pharma avec vite (npm ou yarn);
2. Ajout de la dépendance "react-router-dom" (npm ou yarn)
3. Structuration du projet
4. Page "Login"
5. Page "Home"
6. Page "About"
7. Composant "ProductPreview"
8. Page "ProductsList"
9. Page "SingleProduct"
10. Page "Products"
11. Configuration des routes
12. Class "ProductService"
13. Résumé;
14. Bonus;
15. Aller plus loin

Création du projet gestion-pharma avec vite (npm ou yarn)

```
npm create vite@latest
```

```
cd gestion-pharma
```

```
npm install
```

```
yarn create vite
```

```
cd gestion-pharma
```

```
yarn
```

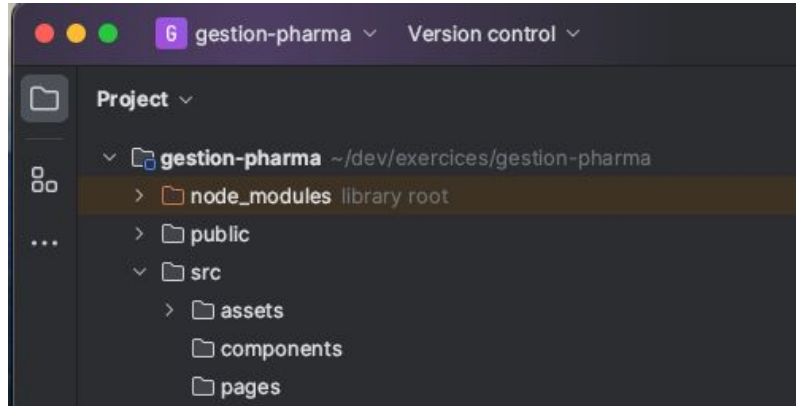
Ajout de la dépendance "react-router-dom" (npm ou yarn)

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

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

Structuration du projet

1. Créons un dossier "**components**" dans "**src**" qui contiendra tous les composants;
2. Ensuite, rajoutons un dossier "**pages**" dans "**src**" qui contiendra à son tour les composants qui serviront des pages principales dans l'application.



Page "Login"

Dans le dossier "**src/pages/**", créons la page "**login**" avec le snippet ci-dessous en utilisant la fonction "**useNavigate()**" fournie par **React Router** qui permet de naviguer vers une autre page depuis le code js.

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

```
function Login() {
```

```
  const navigate = useNavigate();  
  const handleClick = () => {  
    navigate('/home');  
  }
```

```
  return (  
    <>  
      <h1>Login Page</h1>
```

```
      <button onClick={handleClick}>  
        Connexion  
      </button>  
    </>  
  );
```

```
export default Login;
```


Page "Home"

Toujours dans "**src/pages/**", ajoutons la page "home" avec un message de bienvenu

```
function Home() {  
  return (  
    <>  
    <h1>Welcome to Gestion Pharma</h1>  
    </>  
  );  
}  
  
export default Home;
```

Présentons une description de l'application sur la page "**about**"

```
function About() {  
  return (  
    <>  
    <h1>À Propos de nous</h1>  
    <p>Ceci est une application de gestion des produits pharmaceutiques</p>  
    </>  
  );  
}  
  
export default About;
```

Composant "ProductPreview"

- Le composant "**ProductPreview**" va nous servir à afficher l'aperçu d'un produit avec sa description ou un lien "*voir plus*";
- Nous faisons appel au composant "<Link>" fourni par "**React Router**" qui permet de créer des liens de navigation. Dans ce cas, nous passons l'id du produit d'une façon dynamique à la route.

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

```
function ProductPreview({product, seeMore}) {  
  return (  
    <div>  
      <img src={product.image} alt="product image"/>  
      <h2>{product.name}</h2>  
      {  
        seeMore ?  
        <p>{product.description}</p>  
        :  
        <Link to={` /products/${product.id}`}>Voir plus</Link>  
      }  
    </div>  
  );  
}
```

Page "ProductsList"

Cette page récupère la liste des produits afin d'afficher son contenu. Elle utilise des données mock à partir de "*ProductService.getProducts()*" à retrouver aux slides suivants.

```
import ProductPreview from "../components/productPreview.jsx";
import ProductService from "../models/services/productService.js";

function ProductsList() {

  const data = ProductService.getProducts();

  return (
    <>
      <ul>
        {
          data.map((e) =>
            <li>
              <ProductPreview product={e} seeMore={false}></ProductPreview>
            </li>
          )
        }
      </ul>
    </>
  );
}
```

```
export default ProductsList;
```

Page "SingleProduct"

- A present, écrivons le code pour la page "**SingleProduct**" qui à partir du lien dynamique ".../:id" va extraire l'identifiant passé en paramètre (dans l'url) pour ensuite récupérer le produit correspondant et le passer comme "**Props**" au composant "<**ProductPreview**>";
- Nous utilisons la fonction "**useParams()**" pour récupérer ce paramètre dynamique.

```
import {useParams} from "react-router-dom";
import ProductService from "../models/services/productService.js";
import ProductPreview from "../components/productPreview.jsx";

function SingleProduct() {

  const {id} = useParams();

  const data = ProductService.getProducts();

  const product = data.find((e) => e.id === id);

  return (
    <>
      <ProductPreview product={product} seeMore={true}></ProductPreview>
    </>
  );
}

export default SingleProduct;
```

Page "Products"

Et la page "Products" qui va imbriquer soit "la liste des produits" soit "la description" d'un produit grâce au composant "<Outlet>" de "React Router"

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

```
function Products() {
```

```
  return (  
    <>  
      <h1>Products</h1>  
      <Outlet/>  
    </>  
  );  
}
```

```
export default Products;
```

Configuration des routes

Dans le fichier "**main.jsx**", configurons nos routes pour indiquer à "**React Router**" de faire correspondre l'url de navigation à nos différentes pages :

- "/" -> Page "Login";
- "/home" -> Page "Home";
- "/about" -> Page "About";
- "/products/" -> Page "ProductsList"
- "/products/1" -> Page "SingleProduct" avec description où produit "Id" == "1";
- Etc.

Nous faisons appel à la fonction "**createBrowserRouter()**" et au composant "**<RouterProvider>**" de "React Router" pour cette configuration :

Configuration des routes

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import './index.css'
import Login from './pages/login.jsx';
import {createBrowserRouter, RouterProvider} from "react-router-dom";
import Home from './pages/home.jsx';
import SingleProduct from './pages/singleProduct.jsx';
import About from './pages/about.jsx';
import Products from './pages/products.jsx';
import ProductsList from './pages/productsList.jsx';
```

```
const router = createBrowserRouter(
  [
    {
      path: "/",
      element: <Login/>
    },
    {
      path: "/home",
      element: <Home/>
    },
    {
      path: "/about",
      element: <About/>
    },
    {
      path: "/products",
      element: <Products/>,
      children: [
```


Class "ProductService"

```
class ProductService {
    static getProducts() {
        const products = [
            {
                id: "1",
                name: "Doliprane",
                image: "https://dummyimage.com/200x200/000/fff&text=Doliprane",
                description: "Le Doliprane est un médicament contre la douleur et la fièvre couramment utilisé pour soulager les maux de tête, les douleurs musculaires, les courbatures, les douleurs dentaires, les maux de gorge et les fièvres."
            },
            {
                id: "2",
                name: "Aspirine",
                image: "https://dummyimage.com/200x200/000/fff&text=Aspirine",
                description: "L'aspirine est un médicament anti-inflammatoire non stéroïdien (AINS) qui est utilisé pour soulager la douleur, la fièvre et l'inflammation."
            },
            {
                id: "3",
                name: "Ibuprofène",
                image: "https://dummyimage.com/200x200/000/fff&text=Ibuprofene",
                description: "Il est disponible en comprimés, en gélules, en suppositoires et en sirop. La dose recommandée pour les adultes est de 200 à 400 milligrammes par prise, jusqu'à 4 prises par jour"
            }
        ];

        return products;
    }
}
```

- Nous avons installé la librairie "**react-router-dom**" pour pouvoir gérer le routage dans notre applications React;
- "**createBrowserRouter()**" et "<**RouterProvider**>" pour configurer nos routes;
- "**useNavigate()**" pour naviguer depuis une fonction JS;
- "<**Link**>" pour créer des liens de navigations dans nos composants;
- "**useParams()**" pour récupérer les paramètres d'une URL avec un retour "clé/valeur".

- Ajouter une barre de navigation en écartant la page "Login";
- Personnaliser le style;
- Et plus....

Ajouter une barre de navigation

Créons un composant **"NavBar"** avec **"NavLink"**

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

function NavBar() {
  return (
    <ul>
      <li>
        <NavLink to="/home">Home</NavLink>
      </li>
      <li>
        <NavLink to="/products">Product</NavLink>
      </li>
      <li>
        <NavLink to="/about">A Propos</NavLink>
      </li>
      <li>
        <NavLink to="/">Deconnexion</NavLink>
      </li>
    </ul>
  );
}

export default NavBar;
```

Ajoutons un composant "Layout" parent

L'objectif ici est de placer un composant parent qui va contenir notre "**NavBar**" et les pages une fois l'utilisateur connecté

```
import NavBar from "../components/navBar.jsx";  
import {Outlet} from "react-router-dom";
```

```
function Layout() {  
  return (  
    <>  
      <nav>  
        <NavBar/>  
      </nav>  
      <main>  
        <Outlet/>  
      </main>  
    </>  
  )  
}
```

```
export default Layout;
```

Adaptations des routes

```
const router = createBrowserRouter([
  {
    path: "/",
    element: <Login/>,
  },
```

```
  {
    path: "/",
    element: <Layout/>,
    children: [
      {
        path: "/home",
        element: <Home/>
      },
      {
        path: "/about",
        element: <About/>
      },
      {
        path: "/products",
        element: <Products/>,
        children: [
          {
            path: "",
            element: <ProductsList/>
          },
          {
            path: ":id",
            element: <SingleProduct/>
          }
        ]
      }
    ]
  }
])
```

- Documentation "React Router" : <https://reactrouter.com/en/main>;
- D'autres concepts à voir : "errorElement";
- Repo de l'exercice : <https://github.com/elgsylvain85/gestion-pharma.git>.

Thank You

