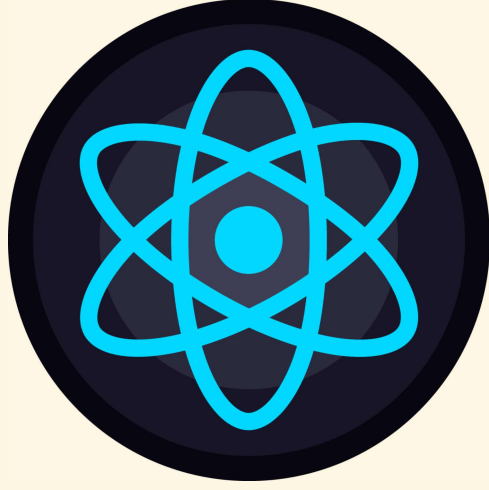


REACT : ROUTER

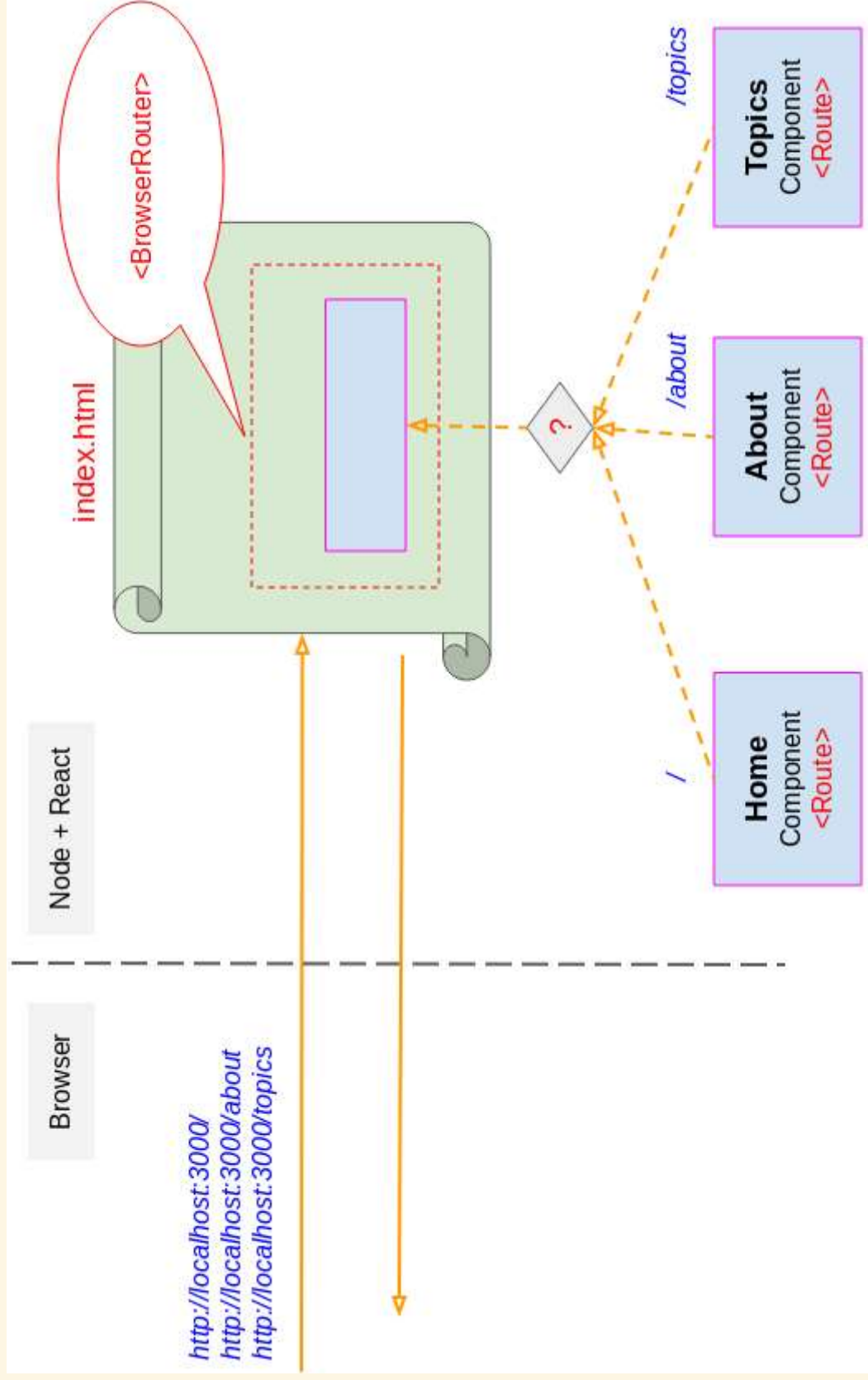


QU'EST-CE QUE LE ROUTER ?

- Mécanisme permettant de naviguer d'une page à une autre sur un site web
- React dispose de la librairie `react-router-dom`



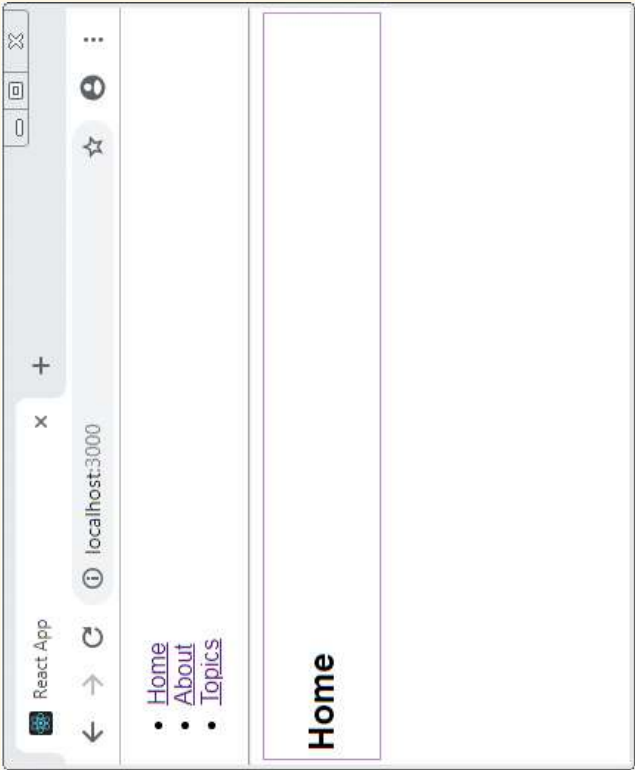
QU'EST-CE QUE LE ROUTER ?



QU'EST-CE QUE LE ROUTER ?

React Router permet de définir des **URL** dynamique et de sélectionner un **Composant** approprié pour **render** (afficher) sur le navigateur d'utilisateur en correspondance à chaque **URL**





METTRE EN PLACE LE ROUTER

- Installer react-router-dom : `npm install react-router-dom`
- `<BrowserRouter>` stocke l'emplacement actuel dans la barre d'adresse du navigateur à l'aide d'URL



METTRE EN PLACE LE ROUTER

```
// Dans le fichier index.tsx
import React from "react";
import ReactDOM from "react-dom/client";
import "./reset.css";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
// Importer BrowserRouter
import { BrowserRouter } from "react-router-dom";

const root = ReactDOM.createRoot(
  document.getElementById("root") as HTMLElement
);
root.render(
  <React.StrictMode>
    <BrowserRouter>
```



CREER DES ROUTES

- Dans `/src` créer un dossier `routes`
- Se placer dans le dossier `routes`
- Créer un fichier `/routes.tsx`



CREER DES ROUTES

- Dans le fichier `/routes.tsx` nous allons utiliser le type `RouteObject`
 - Il représente un objet qui décrit une route en utilisant `react-router-dom`



CREER DES ROUTES

Il contient les propriétés suivantes :

- **path**: Chaîne de caractères qui définit le chemin de la route (par exemple, "/home")
- **element**: Élément React qui sera affiché lorsque l'application se trouve sur cette route (par exemple, un composant)



CREER DES ROUTES

Exemple:

```
// Dans le fichier routes.tsx

import { RouteObject } from "react-router-dom";
import Accueil from "../pages/accueil";
import Apropos from "../pages/apropos";

const routes: RouteObject[] = [
  {
    path: '/',
    element: <Accueil />
  },
  {
    path: '/apropos',
    element: <Apropos />
  },
];
```



RECUPERER DES DONNEES SELON UN PARAMETRE DANS L'URL

```
import Profil from "../pages/profil";

const routes : RouteObject = {
  path: 'profil/:id',
  element : <Profil />
};
```



RECUPERER DES DONNEES SELON UN PARAMETRE DANS L'URL

- Utilisation du hook `useParams` de `react-router-dom`

Renvoie un objet contenant tous les paramètres de l'URL pour accéder à un paramètre spécifique en utilisant la déstructuration



RECUPERER DES DONNEES SELON UN PARAMETRE DANS L'URL

- Utilisations de la variable `id` dans la dépendance de votre hook `useEffect`.
 - On parcourt la liste des `profils` et on vérifie si l'`ID` du `profil` correspond à la valeur de `id`



RECUPERER DES DONNEES SELON UN PARAMETRE DANS L'URL

- Si c'est le cas, on utilise le hook `setProfil` pour mettre à jour l'état du **profil** courant
- Utilisation de la valeur de **profil** pour afficher les détails du profil



RECUPERER DES DONNEES SELON UN PARAMETRE DANS L'URL

```
// Dans le composant Profil
import { useEffect, useState } from "react";
import { useParams } from "react-router-dom";
import Profil from "../models/Profil";
import profils from "../models/profils";

const Profil: React.FC = () => {
  const [profil, setProfil] = useState<Profil>();
  const { id } = useParams<{ id: string }>();
  useEffect(() => {
    if (id) {
      heros.forEach((profil) => {
        if (profil.id === +id) {
          setProfil(profil);
        }
      });
    }
  });
};
```



LES ROUTES IMBRIQUEES

- Nous devons faire de la route de profil un enfant de la route racine
- On déplace la route profil pour qu'elle soit enfant de la route racine

```
// Dans routes.tsx
{
  path: '/accueil',
  element: <Accueil />,
  children: [
    {
      path: '/accueil/profil/:id',
      element: <Profil />,
    },
  ],
},
],
},
```



LES ROUTES IMBRIQUEES

Rien ne s'est passé à part le changement d'URL ?
C'est normal nous devons indiquer à la route racine où nous voulons qu'elle rende ses routes enfants. On fait

ça avec `<Outlet>`



LES ROUTES IMBRIQUEES

```
// Dans le composant Profil
import { useEffect, useState } from "react";
import { useParams } from "react-router-dom";
import Profil from "../models/Profil";
import profils from "../models/profils";

const Profil: React.FC = () => {
  const [profil, setProfil] = useState<Profil>();
  const { id } = useParams<{ id: string }>();
  useEffect(() => {
    if (id) {
      heros.forEach((profil) => {
        if (profil.id === +id) {
          setProfil(profil);
        }
      });
    }
  });
};
```





