

React.js with PHP API REST

2. Reprise du projet PHP MVC en React et useState Hook

Nous allons reprendre l'application front-office du blog faite en PHP MVC pour la migrer en React.js

Nous commençons par ajouter bootstrap dans notre fichier index.html

```
index.html > ...
1  <!doctype html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <link rel="icon" type="image/svg+xml" href="/vite.svg" />
6      <meta name="viewport" content="width=device-width, initial-scale=1" />
7      <title>Vite + React</title>
8      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1R1" />
9    </head>
10   <body>
11     <div id="root"></div>
12     <script type="module" src="/src/main.jsx"></script>
13     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcxpyD9F7jL+jjL+kouU" />
14   </body>
15 </html>
```

Nous reprenons le code de la navbar écrite en php pour l'inclure dans un nouveau composant React que nous nommons Navbar.jsx (Pensez à renommer les attributs class en className)

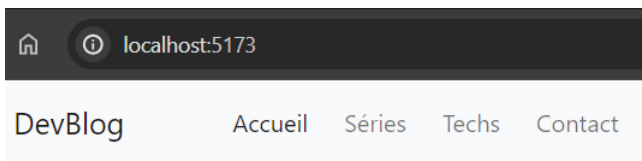
```
src > components > navbar > Navbar.jsx > Navbar
1  function Navbar() {
2    return (
3      <nav className="navbar navbar-expand-lg fixed-top bg-light">
4        <div className="container-lg">
5          <a className="navbar-brand" href="/">DevBlog</a>
6          <button className="navbar-toggler" type="button"
7            data-bs-toggle="collapse" data-bs-target="#navbarContent"
8            aria-controls="navbarContent" aria-expanded="false"
9            aria-label="Toggle navigation">
10             <span className="navbar-toggler-icon"></span>
11          </button>
12          <div className="collapse navbar-collapse ms-5" id="navbarContent">
13            <ul className="navbar-nav me-auto mb-2 mb-lg-0">
14              <li className="nav-item me-2">
15                <a className="nav-link active" href="/">Accueil</a>
16              </li>
17              <li className="nav-item me-2">
18                <a className="nav-link" href="/series">Séries</a>
19              </li>
20              <li className="nav-item me-2">
21                <a className="nav-link" href="/techs">Techs</a>
22              </li>
23              <li className="nav-item me-2">
24                <a className="nav-link" href="#" data-bs-toggle="modal"
25                  data-bs-target="#myModal">Contact</a>
26              </li>
27            </ul>
28          </div>
29        </div>
30      </nav>
31    );
32  }
33  export default Navbar;
```

Le composant Navbar est stocké dans un répertoire src/components/navbar/

Nous l'ajoutons ensuite dans le composant App (à la place des HomeScreen du précédent TP)

```
src > App.jsx > App
1  import './App.css';
2  import Navbar from './components/navbar/Navbar';
3
4  function App() {
5    return (
6      <>
7        <Navbar/>
8      </>
9    );
10 }
11 export default App;
```

Résultat



Dans notre composant Navbar, nous devons inclure le formulaire de contact sous forme de modal (fichier contact.php)

Nous ne pouvons reprendre que le code HTML comme nous l'avons fait dans la Navbar

Nous ajoutons donc un sous-répertoire contact (dans src/components) et y créons le fichier Contact.jsx

En plus de l'attribut class nous avons dû renommer les attributs tabIndex en tabIndex et for en htmlFor

```
src > components > contact > Contact.jsx > Contact
1  function Contact() {
2    return (
3      <div className="modal fade" id="myModal" tabIndex="-1" aria-
4        <div className="modal-dialog">
5          <div className="modal-content">
```

```
<form id="contactForm" method="POST" action="">
  <div className="mb-3">
    <label className="form-label" htmlFor="fullname">Name</label>
    <input className="form-control" name="fullname" type="text" place
  </div>
  <div className="mb-3">
    <label className="form-label" htmlFor="email">Email Address</la
    <input className="form-control" name="email" type="email" place
  </div>
  <div className="mb-3">
    <label className="form-label" htmlFor="message">Message</label>
    <textarea className="form-control" name="message" type="text" p
```

Nous pouvons importer puis ajouter notre composant Contact dans le composant Navbar.

```
src > components > navbar > Navbar.jsx > Navbar
3  function Navbar() {
29      </ul>
30      </div>
31      </div>
32      </nav>
33      <Contact /> ←
34  }
35  }
36  export default Navbar;
```

Pensez à encadrer la balise nav et le composant Contact par un fragment jsx

A ce stade, il y a un problème avec l'attribut style du textarea dans le formulaire de contact.
(voir l'erreur que vous récupérez en console de Chrome)

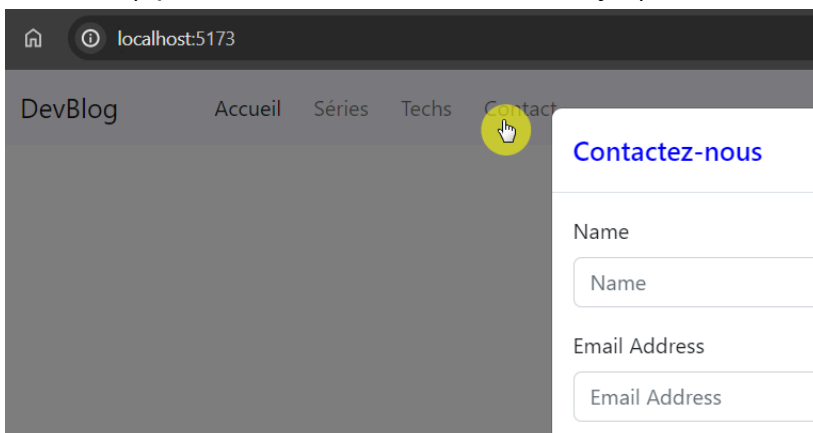
Nous devons modifier la valeur de cet attribut qui doit être un objet littéral (spécificités de React.js)

```
<div className="mb-3">
  <label className="form-label" htmlFor="message">Message</label>
  <textarea className="form-control" name="message" type="text" placeholder="Message"
    style={{height: '10rem'}}></textarea>
</div>
<div className="mb-1 text-center">
  <button className="btn btn-success" name="send" type="submit">Envoyer</button>
</div>
```

Ressources :

<https://www.pluralsight.com/resources/blog/guides/inline-styling-with-react>

Résultat (après modification de l'attribut style)



Nous verrons la partie traitement du formulaire ultérieurement. Nous allons maintenant voir une notion importante en React : le state (useState hook)

Le useState hook

Nous revenons sur notre composant Navbar et allons y ajouter la gestion de la page active dans le menu lors d'un clic

Nous importons le hook useState de React et l'initialisons sur la valeur "/home"

```
src > components > navbar > Navbar.jsx > Navbar
1 | import { useState } from "react";
2 | import Contact from "../contact/Contact";
3 |
4 | function Navbar() {
5 |   const [activeLink] = useState("/home");
6 |
7 |   return (<
8 |     <nav className="navbar navbar-expand-lg fixed-top bg-light">
9 |       <div className="container-lg">
```

Nous modifions la valeur du className des liens de la Navbar pour que la classe active soit présente (ou pas) en fonction de la valeur du state

```
<ul className="navbar-nav me-auto mb-2 mb-lg-0">
  <li className="nav-item me-2">
    <a className={"nav-link" + (activeLink == "/home" ? " active" : "")}
      href="/home">Accueil</a>
  </li>
  <li className="nav-item me-2">
    <a className={"nav-link" + (activeLink == "/series" ? " active" : "")}
      href="/series">Séries</a>
  </li>
  <li className="nav-item me-2">
    <a className={"nav-link" + (activeLink == "/techs" ? " active" : "")}
      href="/techs">Techs</a>
  </li>
  <li className="nav-item me-2">
    <a className="nav-link" href="#" data-bs-toggle="modal"
      data-bs-target="#myModal">Contact</a>
  </li>
</ul>
```

Puis nous ajoutons le setter du state (ligne 5) et créons une fonction qui va gérer le clic sur les liens de la Navbar

```
src > components > navbar > Navbar.jsx > Navbar > handleNavLinkClick
4 | function Navbar() {
5 |   const [activeLink, setActiveLink] = useState("/home");
6 |
7 |   const handleNavLinkClick = (event) => {
8 |     event.preventDefault();
9 |     console.log("href=" + event.target.href);
10 |    const url = new URL(event.target.href);
11 |    console.log("path=" + url.pathname);
12 |    setActiveLink(url.pathname);
13 |   }
14 | }
```

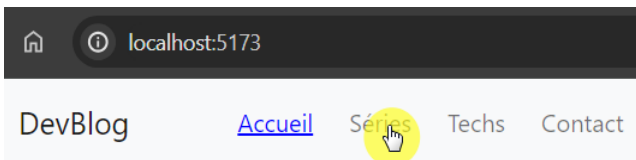
Il nous reste à ajouter l'événement onClick sur nos liens pour appeler la fonction handleNavLinkClick() lors d'un clic

```
<ul className="navbar-nav me-auto mb-2 mb-lg-0">
  <li className="nav-item me-2">
    <a className={"nav-link" + (activeLink == "/home" ? " active" : "")}
      href="/home" onClick={handleNavLinkClick}>Accueil</a>
  </li>
  <li className="nav-item me-2">
    <a className={"nav-link" + (activeLink == "/series" ? " active" : "")}
      href="/series" onClick={handleNavLinkClick}>Séries</a>
  </li>
  <li className="nav-item me-2">
    <a className={"nav-link" + (activeLink == "/techs" ? " active" : "")}
      href="/techs" onClick={handleNavLinkClick}>Techs</a>
  </li>
  <li className="nav-item me-2">
    <a className="nav-link" href="#" data-bs-toggle="modal"
      data-bs-target="#myModal">Contact</a>
  </li>
</ul>
```

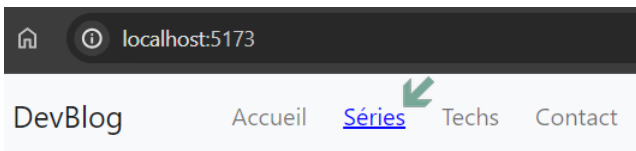
Nous ajoutons un peu de css pour rendre plus visible le lien actif.

```
src > App.css > .nav-link.active
5  .nav-link.active{
6    color: blue !important;
7    text-decoration: underline;
8  }
9
```

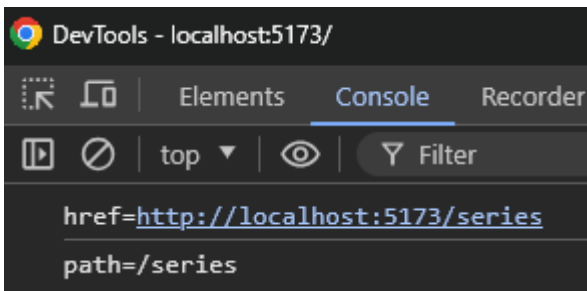
Nous testons en cliquant sur un lien de la Navbar



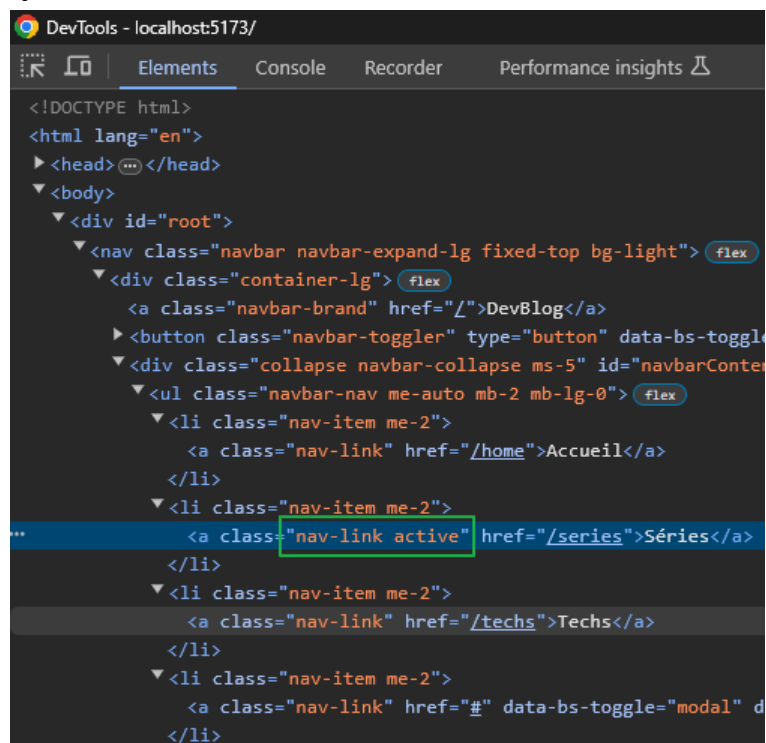
Résultat



En console



Dans l'inspecteur Chrome nous constatons que la classe active a été enlevée du lien Accueil et ajouté sur le lien Séries



```
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body>
    <div id="root">
      <nav class="navbar navbar-expand-lg fixed-top bg-light">
        <div class="container-lg">
          <a class="navbar-brand" href="/">DevBlog</a>
          <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse">
            <div class="collapse navbar-collapse ms-5" id="navbarCollapse">
              <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                <li class="nav-item me-2">
                  <a class="nav-link" href="/home">Accueil</a>
                </li>
                <li class="nav-item me-2">
                  <a class="nav-link active" href="/series">Séries</a>
                </li>
                <li class="nav-item me-2">
                  <a class="nav-link" href="/techs">Techs</a>
                </li>
                <li class="nav-item me-2">
                  <a class="nav-link" href="#" data-bs-toggle="modal" data-bs-target="#exampleModal">Modal</a>
                </li>
              </ul>
            </div>
          </button>
        </div>
      </nav>
    </div>
  </body>
</html>
```

Ressources :

<https://react.dev/learn/responding-to-events>

<https://react.dev/learn/state-a-components-memory>

Exercices sur les states et les events :

<https://www.clientside.dev/blog/react-use-state-practice-exercises>