

TME4 – Construction de l'architecture du site en React

Dans ce TME, nous nous concentrerons sur la page de connexion du site web. Elle inclut un panneau de navigation où l'on pourra se connecter/déconnecter et un formulaire d'inscription.

Ajout des extensions développeur

Sur le navigateur que vous allez utiliser pour tester le fonctionnement de votre site, installez l'extension React Developer Tools :

- sur [Firefox](#)
- sur [Chrome/Chromium](#)
- sur [Edge](#)

Exercice 1 – Page principale

On s'intéresse en premier lieu à la structure du site web.

Q 1.1 Créer un composant `MainPage`, pour le moment vide qui initialise l'état du composant qui permet de spécifier la page courante (page de connexion par défaut) et si l'utilisateur est connecté (`isConnected`, au début `false`).

```
import { useState } from 'react';

function MainPage (props) {
  const [isConnected, setConnect] = useState(false);
  const [page, setPage] = useState("signin_page");

  return (
    <div>

    </div>
  );
}

export default MainPage;
```

Q 1.2 Créer la méthode `getConnected()` qui permet de modifier l'état du composant (page courante sera désormais le ou les forums (page "message_page") et l'utilisateur sera connecté).

```
const getConnected = () =>{
  setConnect(true);
  setPage("message_page");
}
```

Q 1.3 Créer la méthode `setLogout()` qui permet de modifier l'état du composant (page courante : "signin_page" et utilisateur non connecté).

```
const setLogout = () => {
  setConnect(false);
  setPage("signin_page");
}
```

Q 1.4 Afficher ce composant en l'appelant dans le fichier `main.jsx`. Par la suite, on le modifiera pour afficher le panneau de navigation, le contenu de la page (connexion, forum ou page profil...)

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <MainPage />
  </React.StrictMode>
);
```

Exercice 2 – Panneau de navigation

Ce panneau de navigation contient, suivant l'état de connexion, le formulaire pour se connecter ou le bouton pour se déconnecter.

Q 2.1 Créer un composant `NavigationPanel` qui retourne pour l'instant un simple paragraphe.

```
function NavigationPanel (props) {
  return (
    <nav id="navigation_pan">
      <p>Texte temporaire</p>
    </nav>
  );
}

export default NavigationPanel;
```

Q 2.2 Le composant `NavigationPanel` est appelé dans le composant `MainPage`. Lors de son appel, on fournira les attributs suivants : `login=getConnected` `logout=setLogout` `isConnected=isConnected` :

```
<NavigationPanel login={getConnected} logout={setLogout} isConnected={isConnected} />
```

Modifier le composant `NavigationPanel` pour qu'il retourne une balise `<nav>` qui contient soit le composant `Login` soit le composant `Logout`, selon l'état de la prop `isConnected`. Le composant `Login` contient comme attribut qui sera relié à la méthode `getConnected()` définie dans le composant `MainPage`, lors de son appel. De même pour le composant `Logout` qui contient l'attribut qui sera relié à la méthode `setLogout()`.

```
1 import Login from './login';
2 import Logout from './logout';
3
4 function NavigationPanel (props) {
5   return (
6     <nav id="navigation_pan">
7       {(props.isConnected) ? <Logout logout={props.logout}/> : <Login login={props.login}/>}
8     </nav>
9   );
10 }
11
12 export default NavigationPanel;
```

Q 2.3 Créer le composant `Logout` qui affiche le bouton de déconnexion.

```

1 function Logout (props) {
2   return (
3     <div>
4       <button className="button">logout</button>
5     </div>
6   );
7 }
8
9 export default Logout;

```

Q 2.4 Créer le composant `Login` qui affiche le formulaire de connexion (login, mot de passe et bouton). Utiliser un `state` pour stocker login et mot de passe.

```

1 import { useState } from 'react';
2
3 function Login (props) {
4   const [login, setLogin] = useState("");
5   const [password, setPassword] = useState("");
6
7   const getLogin = (evt) => {setLogin(evt.target.value)}
8   const getPassword = (evt) => {setPassword(evt.target.value)}
9
10  return (
11
12    <form method="POST" action="">
13      <label htmlFor="login">Login</label><input id="login" onChange={getLogin}/>
14      <label htmlFor="mdp">Mot de passe</label><input type="password" id="mdp"
15        ↪ onChange={getPassword}/>
16      <button type="submit">Log In</button><button type="reset">Annuler</button>
17    </form>
18  );
19 }
20
21 export default Login;

```

Q 2.5 Vérifiez l’affichage de votre site web ! Changez l’état dans le composant `MainPage` (`isConnected` à `true` ou `false`) pour vérifier que les composants `Login` et `Logout` s’affichent bien.

Exercice 3 – Formulaire d’inscription

Q 3.1 Reprendre les étapes précédentes pour afficher le formulaire d’inscription en dessous du panneau de navigation :

- Création d’un composant `Signin` "vide".
- Modification du composant `MainPage` pour appeler le composant `SignIn`. Attention, cet affichage dépend de la page courante spécifiée dans l’état.
- Modification du composant `Signin` pour reprendre la structure vue lors du TD2. Si les champs de saisie des deux mots de passe sont différents, afficher un message d’avertissement.

```

1 import { useState } from 'react'
2
3 function Signin (props) {
4

```

```

5      const [login, setLogin] = useState("");
6      const [password, setPassword] = useState("");
7      const [firstName, setFirstName] = useState("");
8      const [lastName, setLastName] = useState("");
9      const [passOK, setPassOK] = useState(false);
10
11     const [pass1, setPass1] = useState("");
12     const [pass2, setPass2] = useState("");
13
14     const getLogin = (evt) => {setLogin(evt.target.value)};
15     const getFirstName = (evt) => {setFirstName(evt.target.value)};
16     const getLastName = (evt) => {setLastName(evt.target.value)};
17     const getPass1 = (evt) => {setPass1(evt.target.value)};
18     const getPass2 = (evt) => {setPass2(evt.target.value)};
19
20     const submissionHandler = (evt) => {
21         if (pass1 === pass2) setPassOK(true);
22     }
23
24     return (
25         <div>
26             <label htmlFor="firstname">First name</label><label htmlFor="lastname">Last
27             ↪ name</label>
28             <input id="firstname" onChange={getFirstName}/><input id="lastname"
29             ↪ onChange={getLastName}/>
30             <label htmlFor="signin_login">Login</label><input id="signin_login"
31             ↪ onChange={getLogin}/>
32             <label htmlFor="signin_mdp1">Password</label><input type="password"
33             ↪ id="signin_mdp1" onChange={getPass1}/>
34             <label htmlFor="signin_mdp2">Password (2)</label><input type="password"
35             ↪ id="signin_mdp2" onChange={getPass2}/>
36             <button onClick={submissionHandler}>Sign In</button><button
37             ↪ type="reset">Reset</button>
38             {passOK ? <p></p> : <p style={{color:"red"}}>Erreur: mots de passe différents</p>}
39         </div>
40     );
41 }
42
43 export default Signin;
44
45 ... et pour MainPage :
46
47 import { useState } from 'react';
48 import NavigationPanel from "../navigationpanel";
49 import Signin from "../signin"
50
51 function MainPage (props) {
52     const [isConnected, setConnect] = useState(false);
53     const [page, setPage] = useState("signin_page");
54
55     const getConnected = () => {
56         setConnect(true);
57         setPage("message_page");
58     }
59
60     const setLogout = () => {
61         setConnect(false);
62         setPage("signin_page");
63     }
64
65     return (
66         <div>
67             {page==="signin_page"? <Signin /> : <NavigationPanel login={getConnected}
68             ↪ logout={setLogout} isConnected={isConnected}/>}
69         </div>
70     )
71 }

```

```
22         </div>
23     );
24 }
25
26 export default MainPage;
```

Exercice 4 – La suite...

Q 4.1 Continuer le développement du client jusqu'à avoir le site complet : page forums (un ou deux affichés selon le statut de membre simple ou d'administrateur), page profil... Entre autres, il vous faudra faire un composant pour :

- Un message (si vous choisissez d'associer un message à un/des commentaires, il faudra implémenter les composants du commentaire, de la liste des commentaires, du formulaire de saisie de commentaire...)
- Une liste des messages
- Un formulaire de saisie de message
- Un utilisateur (ses informations, *etc.* pour l'afficher sur le bandeau de la page profil)
- Une page de profil
- Un forum
- ...