LOG4420 – Conception de sites web dynam. et transact.

Travail pratique 4

Chargés de laboratoire:

Konstantinos Lambrou-Latreille

Automne 2019

Département de génie informatique et génie logiciel

1 Objectifs

Le but de ce travail pratique est de vous familiariser avec la libraire React qui permet de faire la gestion d'interfaces interactives côté client.

Plus particulièrement, vous aurez à concevoir une application web avec React à partir du site web du groupe de recherche que vous avez réalisé lors des autres travaux pratiques. En ce sens, vous devrez migrer toute la logique du site web vers des composants React, et vous aurez à communiquer avec les services web que vous avez mis en place lors du travail pratique 3 pour récupérer ou mettre à jour les données sur le serveur.

2 Introduction

Lors des autres travaux pratiques, vous aviez à réaliser un site web qui disposait de plusieurs routes pour accéder aux différentes pages. Ainsi, il était nécessaire d'effectuer des requêtes vers le serveur pour récupérer une page en particulier ainsi que pour charger les ressources contenues dans celle-ci. Tout cela demandait un certain temps de chargement et cela peut parfois affecter négativement l'expérience utilisateur.

Afin d'éviter les longs chargements entre les pages d'un site web, les applications web monopage (*single-page application*) sont devenues très populaires depuis quelques années. En effet, comme leurs noms l'indiquent, une seule page compose le site. En ce sens, il est uniquement nécessaire de récupérer les différentes ressources lors du chargement initial de la page. Par la suite, les mises à jour des vues et des données se font grâce à des requêtes AJAX, ce qui est transparent pour l'utilisateur. Lors de ce travail pratique, vous aurez justement à mettre en place une application web monopage à l'aide de React.

3 Travail à réaliser

À partir du code que vous avez produit lors des travaux pratiques précédents, vous aurez à réaliser une application web pour le groupe de recherche avec React. Puisqu'un gabarit assez complet vous est fourni et que la structure du code React est complètement différente, vous n'avez pas à réutiliser du code des travaux pratiques précédents.

Avant de débuter, assurez-vous d'avoir récupéré l'archive associée à ce travail pratique sur Moodle. Cette archive contient le gabarit à utiliser pour organiser correctement votre code React.

3.1 Prise en charge du gabarit fourni

La première étape à réaliser pour ce travail pratique est de se familiariser avec le gabarit qui vous est fourni. En ce sens, cette section présente les éléments importants pour faciliter cette prise en charge.

3.1.1 Structure du gabarit

Le gabarit fourni comporte trois dossiers principaux : « `client` », « `server` » et « `tests` ». Le dossier « `client` » contient tout le code qui sera exécuté du côté client, le dossier « `server` » inclut l'ensemble du code nécessaire pour l'exécution du serveur et le dossier « `tests` » contient les tests d'acceptation à exécuter pour valider votre travail. Par souci de clarté, la structure du gabarit est illustrée à la figure 1.

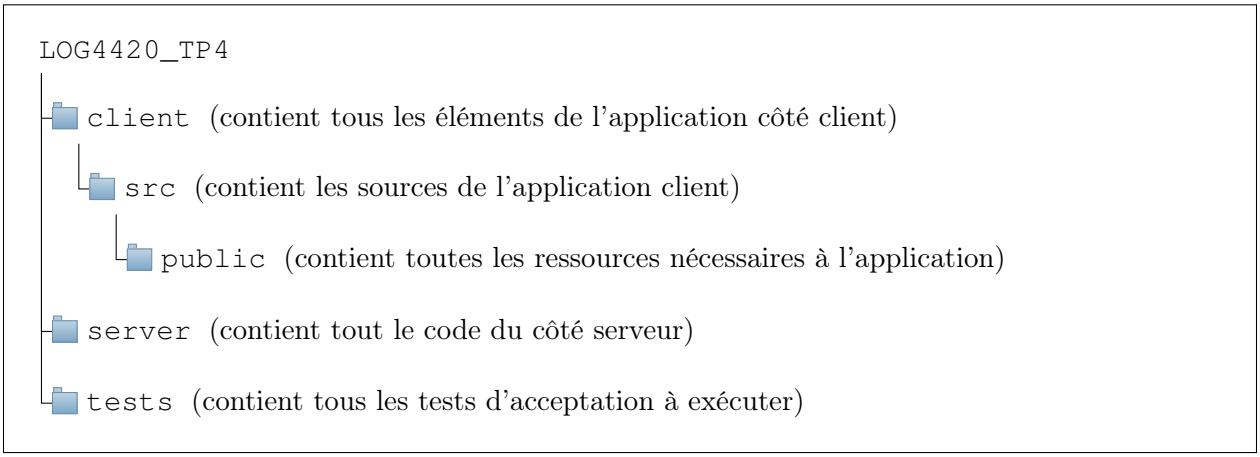

FIGURE 1 – Hiérarchie des dossiers du gabarit fourni

Avant de débuter, vous devez mettre à jour le *connect string* (`mongodb://...`) qui se trouve dans le fichier « `server/config.json` » afin que vous soyez en mesure de vous connecter à votre base de données Mongo.

Les seuls fichiers que vous aurez à ajouter et à modifier se trouvent dans le dossier « `client` ». En effet, le code de l'application React se trouve dans le dossier « `client/src` ». C'est dans ce répertoire que vous aurez à compléter l'application React qui vous est fournie.

En ce qui concerne le dossier « `server` », vous n'avez pas à réaliser de changements dans ce répertoire. Effectivement, le code du serveur vous est fourni pour ce travail. Ainsi, celui-ci définit les mêmes API qui étaient demandées au TP4 à une différence près. Contrairement au quatrième travail pratique, le serveur prend en charge les requêtes multi origines (*cross-site*). En ce sens, le serveur pourra répondre à une requête provenant du domaine « `http://localhost:8000` » alors que le serveur fonctionne sur le domaine « `http://-`

`localhost:3000` ». Ce mécanisme est nécessaire pour ce travail pratique puisque le serveur fonctionnera sur un port différent de l'application client. Par défaut, HTTP n'autorise pas ce type de requêtes pour des raisons de sécurité. Pour en savoir plus sur le partage de ressources de différentes origines (*Cross-Origin Resources Sharing*), vous pouvez consulter ce lien.

3.1.2 Installation et exécution de l'application

Pour être en mesure d'exécuter l'application, il faut d'abord installer les dépendances nécessaires au fonctionnement du serveur et de React en tapant la commande suivante dans le terminal à la racine du projet :

```
npm install
```

Une fois toutes les dépendances installées, il vous suffit de taper la commande suivante dans le terminal pour démarrer simultanément le serveur et l'application React :

```
npm start
```

Si tout se passe comme prévu, le serveur sera fonctionnel à l'adresse suivante : `http://-localhost:3000`. L'application client, quant à elle, sera accessible à l'adresse suivante : `http://localhost:8000`.

3.2 Réalisation de l'application React

La deuxième étape de ce travail pratique est de compléter l'application React qui vous est fournie. Vous aurez à compléter les différents composants correspondants aux pages du site d'achats en ligne que vous aviez à réaliser tout au long de la session. Ces composants se trouvent dans le dossier « `client/src` ». Il est à noter que le gabarit du projet est basé sur Create React App.

Afin de vous simplifier la tâche, un bon nombre d'éléments ont été réalisés pour vous. En ce sens, les routes vers les différents pages ont déjà été configurées à l'aide du module de routage de React et le service permettant de communiquer avec l'API des produits vous est fourni. De plus, la déclaration des composants nécessaires pour le site web de groupe de recherche ainsi que la définition du code Pug des vues utilisées par les composants ont été faites pour vous. Le tableau 1 décrit d'ailleurs les différents composants qui vous sont fournis.

Tableau 1 – Description des composants fournis

Nom	Description
`App`	Composant principal de l'application.
`Home`	Composant responsable d'afficher la page d'accueil.
`Header`	Composant responsable d'afficher l'entête de chaque page.
`Footer`	Composant responsable d'afficher le pied de page de chaque page.
`Project`	Composant responsable d'afficher un projet.
`Projects`	Composant responsable d'afficher les projets.
`Publication`	Composant responsable d'afficher les publications.

À noter que nous vous demandons pas de faire la page d'équipe.

Pour ce travail, vous aurez à compléter les composants suivants : **`Home`**, **`Projects`**, **`Project`** et **`Publication`**. Les fichiers pour chaque composantes sont déjà été créés et chaque fichier contient des instructions sur les tâches à faire. Pour ce faire, vous devez vous assurer de respecter les spécifications des différentes pages du site web qui ont été décrites dans l'énoncé du travail pratique 3. De plus, vous devez communiquer avec les API des nouvelles, des projets et des publications pour mettre à jour les informations sur le client. Afin de vous assurer que vous respectez toutes les exigences demandées pour chacune des pages du site web, des tests d'acceptation automatisés vous ont été fournis afin que vous puissiez valider votre application (voir §Exécution des tests d'acceptation automatisés).

⚠ **Notez bien**

Vous ne devez pas utiliser jQuery ou manipuler le DOM directement pour mettre à jour les différentes vues. En effet, vous devez utiliser la syntaxe React pour créer et manipuler vos vues (liaison unidirectionnelle et bidirectionnelle, boucles, conditions, etc.).
De plus, **aucun rechargement de page est permis** mise à part pour le chargement initial du site web.

3.3 Exécution des tests d'acceptation automatisés

Tout comme pour les deux derniers travaux pratiques, des tests d'acceptation automatisés vous ont été fournis afin que vous puissiez vérifier votre application. En effet, ces tests permettront de valider que votre application web monopage respecte toutes les exigences fonctionnelles demandées. De plus, ces tests serviront à l'évaluation de votre travail. En ce

sens, il est important que vous vous assuriez que tous les tests fonctionnent **avant** de remettre votre travail pratique. Pour exécuter les tests d'acceptation, vous devez entrer la commande suivante dans un terminal :

```
npm run e2e
```

Avertissement

Il est important de n'apporter **aucune** modification aux fichiers dans le dossier `tests` (sauf « `nightwatch.json` »). En effet, ces fichiers contiennent tous les tests automatisés qui sont exécutés sur le site web et se doivent de ne pas être modifiés.

Conseils pour la réalisation du travail pratique

1. N'attendez pas à la dernière minute pour commencer le laboratoire ! L'apprentissage et la prise en charge de React prennent un certain temps.

4 Remise

Voici les consignes à suivre pour la remise de ce travail pratique :

1. Vous devez placer le code de votre projet dans un dossier compressé au format ZIP nommé « `TP4_matricule1_matricule2.zip` ». Assurez-vous d'exécuter la commande « `npm run clean` » dans un terminal à la racine du projet avant de remettre votre travail.
2. Vous devrez également créer un fichier nommé « `temps.txt` » à l'intérieur du dossier de votre projet. Vous indiquerez le temps passé au total pour ce travail.
3. Le travail pratique doit être remis avant **23h55**, le **8 décembre 2019** sur Moodle.

Aucun retard ne sera accepté pour la remise de ce travail. En cas de retard, le travail se verra attribuer la note de zéro. Également, si les consignes 1 et 2 concernant la remise ne sont pas respectées, une pénalité de -5% est applicable.

Le navigateur web **Firefox** sera utilisé pour tester votre site web.

5 Évaluation

Globalement, vous serez évalué sur votre structure de code React ainsi que sur le respect des fonctionnalités du site web. Plus précisément, le barème de correction est le suivant :

Exigences	Points
Respect des exigences du site web	
Résultats des tests d'acceptation automatisés	12
Application React	
Utilisation adéquate des différents éléments React	6
Qualité et clarté du code Javascript	2
Total	20

L'évaluation se fera en grande partie grâce aux tests d'acceptation automatisés qui vous sont fournis.

Ce travail pratique a une pondération de **6%** sur la note du cours.

6 Questions

Si vous avez des interrogations concernant ce travail pratique, vous pouvez poser vos questions sur le forum Moodle du cours. N'hésitez pas à poser vos questions sur ce canal afin qu'elles puissent également profiter aux autres étudiants.