LOG4420 – Conception de sites web dynam. et transact.

Travail pratique 2

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 le langage JavaScript, la manipulation du DOM (à l'aide de la libraire intégrée de Javascript ou jQuery) et les requêtes Ajax.

Ce travail pratique contient deux volets :

— L'apprentissage de Javascript à travers l'implémentation de fonctions

— Introduction à la manipulation du DOM de notre site web de groupe de recherche

À la fin de ce travail, le site web ne sera pas encore 100% utilisable, puisqu'il manquera l'implémentation du côté serveur en Node.

2 Introduction

Lors du premier travail pratique, vous avez mis en place la structure et la mise en forme du site web à réaliser. Cependant, comme vous l'avez sans doute remarqué, celui-ci était complètement statique. En effet, il n'était pas possible de modifier le HTML pour supprimer une publication, par exemple.

Afin d'être en mesure d'ajouter des interactions du côté client, le langage JavaScript doit être utilisé. En ce sens, ce langage permet entre autres de manipuler les éléments du DOM (*Document Object Model*) et de réaliser des requêtes asynchrones (AJAX) pour récupérer des ressources ou pour communiquer avec des services web.

3 Travail à réaliser

Vous allez débuter à partir du corrigé du premier travail pratique disponible sur Moodle afin de partir du bon pied pour ce deuxième travail.

Avertissement

Afin de promouvoir la compréhension du langage JavaScript, seule la bibliothèque jQuery (mais elle n'est pas obligatoire) est permise pour ce travail pratique.

Avant de débuter, assurez-vous d'avoir récupéré l'archive associée à ce travail pratique sur Moodle. Cette archive contient des fichiers supplémentaires que vous devez inclure dans votre projet. Pour ce faire, copiez l'ensemble des fichiers de l'archive dans votre dossier de projet.

3.1 Prise en charge de Node.js

Au cours de ce travail pratique ainsi que des prochains, vous aurez à utiliser Node.js. Il s'agit d'un environnement multiplateforme qui permet l'exécution d'applications JavaScript. Node.js est extrêmement populaire en raison de sa légèreté et de son efficacité. En effet, l'environnement est basé sur un modèle d'évènements asynchrones permettant d'éviter les blocages et les attentes (p. ex. entrées/sorties) tout en étant conçu pour créer des applications extensibles (*scalables*) [1].

L'environnement Node.js est déjà installé sur les postes du laboratoire à Polytechnique. Vous pouvez consulter ce lien si vous souhaitez l'installer sur votre machine.

Node.js est également livré avec le gestionnaire de paquets npm (*Node Package Manager*). Ce gestionnaire de paquet est extrêmement utile puisqu'il répertorie plusieurs centaines de milliers de paquets pouvant être utilisés pour le développement d'applications de toutes sortes. Par ailleurs, ce gestionnaire de paquets peut facilement être utilisé via le terminal. En effet, les commandes sont invoquées en utilisant le mot `npm` suivi du nom de la commande. Vous utiliserez principalement les commandes `install`, `uninstall` et `start`.

Toutes les applications Node.js qui utilisent le gestionnaire de paquets npm doivent posséder le fichier `package.json` à leurs racines. Ce fichier indique plusieurs informations telles que le nom, les auteurs ainsi que les dépendances qui sont utilisés par l'application.

En ce qui concerne ce travail pratique, le fichier `package.json` vous est fourni avec toutes les dépendances nécessaires pour le bon fonctionnement de l'application web. En d'autres mots, vous n'avez pas besoin d'installer de nouvelles dépendances pour ce travail. Cependant, vous devez installer les dépendances nécessaires en tapant la commande suivante à la racine de votre projet :

```
npm install
```

Cette commande installera toutes les dépendances dans un nouveau dossier appelé « `node_modules` » à la racine de votre projet.

1. Pour en savoir plus sur les particularités de Node.js, vous pouvez consulter ce lien.

3.2 Apprentissage du langage Javascript

Le premier volet de ce travail pratique est de vous familiariser avec le language Javascript sur la version ES6. Pour se faire, vous allez implémenter l'ensemble des fonctions qui se trouve dans le fichier `./javascripts/utils.js`. Dans cette partie, vous n'êtes pas obligé de lancer le site web à travers la commande `npm start`.

Des tests unitaires ont été mis en place pour vous permettre de vérifier que les fonctions ont été correctement implémentées. Pour les exécuter, simplement lancer la commande `npm test`.

L'idée de cette partie est de vous familiariser avec les concepts de la programmation fonctionnelle qui sont possibles de mettre en pratique en Javascript pour la manipulation de données.

ⓘ Contraintes pour la réalisation de cette partie

1. Aucune boucle `for` ou `while` doit être utilisée.
2. Vous ne pouvez pas simuler une boucle avec la recursivité (sauf pour les fonctions `map`, `find`, et `fold`).
3. Au lieu des boucles, utilisez les méthodes `map, filter, find, reduce` pour la manipulation de tableaux.
4. Sauf pour la fonction `genererCompteur`, vous devez uniquement déclarer des variables avec `const`.
5. Vous ne pouvez pas utiliser les méthodes `map, find, reduce` dans les fonctions `map, find, fold` que vous devez implémenter.
6. Respectez une convention de codage et produisez un code propre et lisible (p. ex. alignement consistant, noms de variables claires, etc.).
7. L'implémentation de la fonction `fold` est optionnelle. Si vous réussissez à l'implémenter, j'ajoute un 0.5/20 point boni.

3.3 Manipulation du DOM

Après avoir implémenté les fonctions Javascript, vous pouvez commencer la section qui porte sur la manipulation du DOM. Vous avez le choix d'utiliser la libraire intégrée de Javascript ou jQuery pour la manipulation du DOM et pour les requêtes AJAX.

Les sous-sections qui suivent décrivent les éléments à réaliser pour les différentes parties

du site web. Il est à noter que le ou les fichiers qui contiendront votre code JavaScript doivent se trouver dans le dossier `./javascripts`.

N'oubliez pas de lancer le site web avec la commande `npm start`.

Notez bien

Assurez-vous de respecter les identifiants (`id="..."`) et les classes (`class="..."`) qui vous sont imposés. En effet, les éléments qui doivent posséder des identificateurs particuliers seront utilisés par les tests automatisés qui devront être exécutés sur le site web (voir la section §Exécution de tests d'acceptation automatisés).

3.3.1 Entête

Lorsqu'on change de page, vous devez afficher la bonne page courante dans le menu de navigation avec la classe CSS `.active`. P. ex. dans la page d'accueil, on met un arrière-plan bleu pour l'item `Accueil`, alors qu'on mettrait un arrière-plan bleu pour l'item `Équipe` dans la page de membres.

3.3.2 Page d'accueil

Notez que la section qui décrit le site web n'est plus disponible, donc l'élément HTML `div.jumbotron` est vide. Par contre, la variable qui décrit le site se trouve sur le serveur à travers l'adresse `http://localhost:3000/api/description`. Si vous essayez d'accéder à cette URL, le serveur va vous envoyer la description sous le format de partage de données JSON. Ce format de représentation de données est très similaire à un objet Javascript, donc il se manipule de façon similaire.

Votre travail consiste à faire un appel AJAX vers cette URL pour récupérer la description et de l'insérer dans la balise `div.jumbotron`. N'oubliez pas d'appeler la méthode `.split(...)` pour séparer la description en paragraphe et afficher plusieurs balises `p` à l'intérieur de `div.jumbotron`.

3.3.3 Page des publications

Dans la page des publications, la première fonctionnalité à implémenter est la suppression d'une ligne du tableau de publication lorsqu'on clique sur l'icône de la poubelle.

La deuxième fonctionnalité consiste à gérer les options d'affichage des publications, notamment les options de trie et de pagination. L'utilisateur a la possibilité de modifier les options

d'affichage qui modifieront l'URL de la page en ajoutant des paramètres *query string*. Le numéro de la page en cours est modifié avec l'option `page`, le nombre d'éléments par page avec l'option `limit`, la méthode de trie avec l'option `sort_by` et l'ordre de trie avec l'option `order_by`.

Par défaut, si on accède à l'URL `/publications`, on affiche la 1re page des 10 premières publications triées en fonction de la date en ordre décroissant. Ensuite, chaque action de l'utilisation qui modifieront les `input` modifiera également l'URL de la page.

Voici une séquence de cas d'utilisation possible d'un utilisateur. S'il choisit de trié selon le titre, alors il est redirigé vers la page :

```
/publication?sort_by=title
```

S'il choisit de trié de nouveau par la date, alors il est redirigé vers la page :

```
/publication?sort_by=date
```

S'il choisit de trié en ordre croissant, alors il est redirigé vers la page

```
/publication?sort_by=date&order_by=asc
```

Notez que le programme se rappelle toujours des anciennes options choisies. S'il change de numéro de page, alors il est redirigé vers la page :

```
/publication?sort_by=date&order_by=asc&page=2
```

S'il change le nombre d'éléments à afficher par page, alors il est redirigé vers la page :

```
/publication?sort_by=date&order_by=asc&page=1&limit=20
```

Notez que lorsqu'on modifie le nombre d'éléments par page, la méthode de trie ou l'ordre, on revient toujours vers la première page.

3.4 Exécution de tests d'acceptation automatisés

Afin de valider l'ensemble des éléments à réaliser pour ce travail pratique, des tests d'acceptation automatisés (*End-to-End Tests*) vous ont été fournis afin que vous puissiez vérifier

votre travail. Tous les tests ont été réalisés avec Nightwatch.js. Il est à noter que ces tests serviront à l'évaluation de votre travail. En ce sens, il est important que l'ensemble des tests fonctionnent **avant** que vous remettiez votre travail pratique.

Avant d'exécuter les tests, assurez-vous que le serveur web est **en fonction**. Pour exécuter les tests, veuillez entrer la commande suivante dans un terminal à la racine du projet :

```
npm run e2e
```

L'environnement de test est actuellement configuré pour fonctionner sur n'importe quel système d'exploitation. Par défaut, Nightwatch utilise Firefox pour tester le site web. Si vous vouez utiliser Chrome, allez dans le fichier `package.json` et changez la valeur `firefox` de la commande `e2e` vers la valeur `chrome`. Également, par défaut, les tests d'acceptation automatisés sont lancés pour chaque page du site web. Si vous voulez, vous pouvez exécuter les tests d'une seule page en modifiant le script `npm run e2e`. Allez dans le fichier `package.json` et modifiez la commande `e2e` avec la ligne suivante :

```
"e2e": "nightwatch --config ./tests/nightwatch.conf.js --env firefox
↪  ./tests/e2e/accueil.js"
```

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. Assurez-vous donc de respecter les identifiants et les classes imposés afin que l'ensemble des tests puissent fonctionner.

Conseils pour la réalisation du travail pratique

1. Utiliser le mode strict dans vos scripts pour faciliter le débogage de votre code.
2. Utilisez le débogueur du navigateur pour vous aider à identifier les erreurs.
3. N'attendez pas à la dernière minute pour commencer le laboratoire ! Le débogage en JavaScript peut parfois s'avérer long et ardu.
4. Soyez consistant dans vos conventions de codage (voir guide de codage de Mark Otto).

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 « `TP2_matricule1_matricule2.zip` ». Assurez-vous d'exclure le dossier « `node_modules` » 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 **9 octobre 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 le respect des exigences des éléments à réaliser. Plus précisément, le barème de correction est le suivant :

Exigences	Points
Respect des exigences pour les éléments à réaliser	
Code Javascript (utils.js)	
Exécution des tests unitaires et respect des exigences	9
Implémentation de fold (bonus)	0.5
Manipulation du DOM	
Affichage de la page courante	1
Requête AJAX dans la page d'accueil	2
Supression d'une publication	2
Gestion des options d'affichage des publications	4
Qualité et clarté du code	2
Total	20

L'évaluation se fera en grande partie grâce aux tests unitaires et aux tests d'acceptation automatisés qui vous sont fournis. En effet, les tests automatisés permettront de valider les exigences pour les éléments à réaliser.

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 de laboratoire sur `Moodle`. N'hésitez pas à poser vos questions sur le forum afin qu'elles puissent également profiter aux autres étudiants.