Le génie pour l'industrie

Laboratoire 2

Application web Éco Vélo

PAR,

Maxime Chaussé, CHAM18109501

Edwin S Lopez Andino, AQ43820

Juan Esteban Ladino, LADJ16039708

Vincent Chang, CHAV09089808

Soumis le 8 juillet 2022 à

Responsable de laboratoire : Marcos Dias de Assuncao

et l'enseignant : Marcos Dias de Assuncao

Cours : GTI525 - Technologies de développement Internet

Session Été 2022

Introduction

De nos jours, il devient de plus en plus rare de devoir développer des applications pour le web qui sont statiques et qui ne communiquent pas avec aucun service de données. Dans notre laboratoire 1, nous avons développé une application qui ne comportait seulement qu'une partie frontale. Dans ce rapport de laboratoire, nous allons expliquer comment nous avons intégrer une partie dorsale qui va nous permettre de gérer efficacement les requêtes aux données nécessaires à l'application ainsi que les choix que nous avons pris pour ajouter des fonctionnalités dans la partie frontale afin effectuer un traitement et pour formater c'est donné de façon à créer une belle vue pour l'utilisateur.

Questions

R1: Décrivez l'architecture logicielle utilisée pour la partie front-end de ce livrable et comment elle a évolué par rapport au livrable 1. Décrivez brièvement l'organisation et le rôle des différents éléments (classes, fonctions) de votre code JavaScript. Si vous avez utilisé des cadriciels, mentionnez-les et indiquez de quelle façon ils sont utilisés.

Nous avons fait l'implémentation de la partie frontale dans ce laboratoire en utilisant le concept de composants. Nous avons donc créé un composant pour toutes les différentes sections de l'application. Par exemple, nous avons un composant nommé content qui sert de conteneur pour injecter d'autres composants. La route utilisée détermine le composant qui sera injecté. Le fichier « app-routing » est celui qui gère toutes les routes accessibles et qui les paires avec un composant.

La raison pour laquelle nous avons utilisé ce type de structure est parce que nous avons décidé d'utiliser le cadriciel Angular pour simplifier la structure frontale de l'application. C'est aussi le cadriciel que nous avons utilisé lors du laboratoire un alors nous avons pu itéré sur la structure déjà présente. Nous avons ajouté deux nouveaux composants sous le dossier content puisque ce sont des vues pour la nouvelle carte et le diagramme d'utilisation. Le dossier services est celui qui a le plus grandi depuis le laboratoire précédent. Il contient maintenant quatre classes , dont trois nouvelles qui ont été ajoutées. Celles-ci ont pour but de faire les requêtes à la partie dorsale du projet.

R2: Décrivez de quelle façon vous vous y êtes pris pour afficher la carte et les marqueurs correspondants aux compteurs vélos. Si vous avez utilisé des librairies, mentionnez-les et décrivez de quelle façon elles sont utilisées.

Tout d'abord, avant de se lancer dans l'affichage de la carte, on a fait plusieurs recherches sur quelques services externes (API) et on a constaté qu'il y en avait plusieurs, mais que la plupart étaient payants. Heureusement, notre chargé de laboratoire nous a proposé d'utiliser Leaflet. C'est une librairie de code source ouvert qui nous permet d'avoir une carte interactive. Le meilleur de cette librairie, c'est qu'elle est totalement gratuite et compatible

avec notre application. Étant donné qu'on se sert du cadriciel Angular (qui utilise une structure de composantes), on peut traiter cette librairie comme un composant. Pour se servir de cette librairie, il a fallu tout d'abord importer un service de modèle qui est fourni par NG Bootstrap qui nous permettait de mettre en arrière-plan une fenêtre sans nécessairement changer d'onglet. Dans cette fenêtre, on a mis en place la librairie Leaflet qui nous permet d'afficher une carte et aussi de laisser des marqueurs pour situer des emplacements spécifiques.

R3: Décrivez l'architecture logicielle utilisée pour la partie back-end de ce livrable. Décrivez brièvement l'organisation et le rôle des différents éléments (classes, fonctions) de votre code JavaScript. Décrivez également votre choix de cadriciels pour l'implémentation de la première version de l'application dorsale et de l'API pour fournir des informations sur les nombres de passages détectés par les compteurs.

Pour le cadriciel de la partie dorsale, nous avons utilisé *node.js* car deux membres de l'équipe étaient familiers avec celui-ci donc la courbe d'apprentissage était minime. Nous avons pu servir les pages statiquement du côté serveur ce qui nous a permis d'avoir une application *MVC*. Donc, nous avons divisé l'application en trois grandes couches: modèle, vue et contrôleur. Nous avons au total sept paquets de classes: les modèles, les vues, les routes, les contrôleurs, les utilitaires, les avoirs (*assets*) et le service de base de données (MongoDB). En ce qui concerne les modèles, ce paquet contient la définition des données utilisée par l'application comme les compteurs, les fontaines et les statistiques. Pour le paquet des routes, celui-ci contient les requêtes de l'API pour rediriger vers le bon contrôleur (fontaine, compteur ou statistiques). Le paquet des contrôleurs regroupe les contrôleurs des différentes classes de données (fontaine, compteur ou statistiques) qui exécutent la demande des routes. Pour le paquet des vues, celui-ci contient toutes les vues qui parviennent de l'application frontale du cadriciel Angular. Pour le paquet utilitaire, celui-ci contient la gestion des dates ainsi que des scripts pour enregistrer les données statistiques vers la base de données. Le paquet avoir (*assets*) contient les différents fichiers Excel des données brutes fournies par le professeur. Finalement, le paquet service contient la gestion de connexion vers la base de données MongoDB.

Pour la première version de l'application dorsale, nous avons utilisé une base de données MongoDB. Les données brutes des fichiers Excel sont difficilement malléables donc nous avons créé un *script* pour les rendre plus utilisables dans notre application. Ce *script* s'exécute seulement lors de la première exécution de l'application. Il transforme les données en objet MongoDB pour que nous soyons capables d'en faire la recherche correctement. Nous avons exposé une route (localhost:3000/gti525/stats/id?debut=...&fin=...) qui fait appel au contrôleur des statistiques. Ce contrôleur se charge de faire la recherche des passages de la borne entre les dates voulues et retourne tous les passages vers la route. Finalement, la route se charge de renvoyer les résultats vers la vue.

R4: De quelle façon avez-vous subdivisé les tâches en équipe pour ce livrable? Décrivez le rôle et les tâches assignées à chacun des membres.

Pour ce nouveau livrable, on a changé un peu la façon de se distribuer les tâches. Heureusement, il y a deux membres de l'équipe qui sont très familiers avec le cadriciel Angular (Maxime Chaussé et Edwin S Lopez Andino), c'est pour cela qu'ils se sont occupés

des tâches les plus difficiles. Pour être plus précis, ces deux membres se sont occupés de la création de l'API et de la partie dorsale ainsi que d'ajouter l'affichage de la vue des statistique dans la partie frontale. Pour les deux autres membres de l'équipe, Juan Esteban Ladino et Vincent Chang, leurs tâches étaient de se servir de la librairie Leaflet pour faire les affichages de la carte. Malgré le fait qu'il y a une seule tâche pour deux personnes, on constate que pour chacun d'entre nous, c'était un vrai défi.

Conclusion

En conclusion, nous avons implémenté les nouvelles fonctionnalités demandées pour ce livrable. La partie dorsale de l'application est la nouveauté la plus importante de ce livrable. Grâce à celle-ci, l'application frontale peut maintenant obtenir les données provenant d'une base de données afin de les afficher. Le développement de ces nouvelles fonctionnalités nous a permis de nous pratiquer avec les interactions entre notre application frontale, notre application dorsale et notre base de données, en plus d'approfondir nos connaissances sur les technologies que nous utilisons pour ce projet. Nous pourrons ainsi travailler avec une plus grande facilité lors du prochain livrable.

JavaScript est un langage très populaire et puissant, c'est pourquoi il existe plusieurs autres cadriciels pour faciliter son développement autre que *Angular*. Dans nos futur projet, il serait intéressant de faire une application frontale à l'aide de *React Native* ou *Vue.js* pour voir comment ils gèrent la structure et le code en comparaison avec *Angular*.