[Date]

Choix technologiques

Projet de fin d’étude

Denuit Maxime, Degrève Olivier, Dubois Corenthin, Tavernier cedric

Table des matières

# Choix Technologique :

## Back-End :

Nous utiliserons un back end en nodeJS. L’avantage de cette technologie est qu’elle est déjà très simple à apprendre. Dans notre groupe, 2 personnes connaissent déjà le NodeJS grâce au cours enseigné et les 2 autres possèdent des connaissances en JavaScript, ce qui rendra la tâche assez simple si l’un d’entre d’eux devait l’apprendre.

Et comme NodeJS est assez récent, la documentation est très facile de compréhension et facilement trouvable sur le site de NodeJS.

Une autre chose intéressante à propos de Node est sa grande modulabilité. Il existe beaucoup de librairies externes permettant d’effectuer des tâches pour un backend. Que ce soit des librairies serveurs, de requêtes http/https ou autre. Ce même genre de librairie existe aussi pour le front pour avoir une interface simple à mettre en place (Exemple : AngularJS) ou pour avoir des fonctions temps réel grâce aux WebSockets de Node.

Si l’on compare NodeJS au Java pour le backend, il y a clairement plusieurs avantages. D’abord, l’utilisation de Node permet le développement d’une application de type « full stack » (C’est-à-dire que l’entièreté du code est en JavaScript) ce qui permet de ne pas avoir de middleware pour convertir d’éventuel données transmises par le front. (Ce qui est le cas de java, il faut, par exemple, convertir le JSON avec Genson).

Ensuite, NodeJS a l’avantage d’être plus rapide que java pour faire de l’Input/Output et ne bloque pas d’éventuels requêtes supplémentaires. En effet, Il y a une bonne gestion de la concurrence en Node ce qui permet de pouvoir effectuer des requêtes Input/Output en parallèle. Bien entendu, il est possible d’implémenter de la concurrence en Java mais cela requiert plus de temps et engendre potentiellement plus de problèmes si mal implémenté.

NodeJS a aussi l’avantage d’avoir une très bonne scalabilité. En effet, on peut déployer rapidement une application Node pour une entreprise que ce soit une application web ou mobile. De plus, nodejs à l’avantage d’avoir de très bonnes performances et d’être assez rapide lorsqu’on effectue des requêtes.

Cependant, NodeJS est là principalement pour effectuer des tâches peu complexes. En effet, si l’application requiert des calculs complexes, il serait alors préférable de donner la charge à un autre processus node alors que Java n’aurait pas de problème pour le faire en un seul processus.

Niveau structure, Node peut utiliser des modules supplémentaires pour intégrer plus de fonctionnalités. Ces modules se situeront dans un dossier « node\_modules » à la racine du dossier contenant le projet. Sinon, il est préférable d’avoir quelques dossiers avec les différentes parties du code et d’éviter les sous-dossiers pour ne pas avoir de chemins trop longs.

## Front-End :

Pourquoi avons-nous décidé d’utiliser de l’HTML5/CSS3 ?

Nous avons décidé d’utiliser la dernière version d’HTML5/CSS3 car celle-ci nous permet d’utiliser toutes les dernières fonctionnalités.

Un inconvénient mineur sera qu’on devra tenir en compte que certains navigateurs ne prendront pas en compte la dernière version de l’HTML5. Pour régler ce problème nous devrons donc coder en fonction du navigateur le moins « up to date ».

Pourquoi nous n’avons pas utilisé bootstrap ?

Il nous semble intéressent de pouvoir créer de l’html / css « fait maison », ce qui nous permet d’utiliser l’html/css dans nos codes JAVASCRIPT facilement, avec les balises que nous voulons. Il nous semble intéressant de créer nous même un site réponsif et non d’utiliser un Template tout fait, comme proposé par bootstrap.

Comme language de programmation front-end nous avons décidé d’utiliser JAVASCRIPT 1.8 car :

Le javascript permet d’agir directement et ne dois pas attendre que les serveurs envoient une réponse. Ceci accélère l’ouverture des sites web sur les navigateurs des clients. Le javascript ne nécessite pas un programme d’interprétetation comme par exemple Flash player, en plus de ça, le javascript n’occupe pas une grande place sur disques des sites web.

# Diagramme d’architecture :

# Outils utilisés :

## Git – Github :

Pour réaliser à bien ce projet, trois outils distincts vont être utilisés.

Git-Github, en est le premier. D’une part, l’hébergement de projets et de dépôts, sont pris en charge par le site web Github. Cela facilite la collaboration sur des projets entre plusieurs intervenants. D’autre part, nous avons un logiciel qui permet aux utilisateurs de sauvegarder différentes versions de fichiers durant le cycle de vie d’un projet sur leur pc, à savoir : Git. Cette application permet également aux utilisateurs d’envoyer leurs différentes versions sur leur compte github ou sur un projet partagé. On pourra ensuite “merger“ les fichiers de tous les collaborateurs du projet en un travail unique.

Pourquoi s’être tourné vers Git-Github ?

Parce que github est considérée comme étant la plateforme d’hébergement de référence. Elle réunit plus de 5,8 millions de développeurs actifs et la société lève en permanence des fonds importants pour améliorer leur service mais aussi pour y ajouter de nouveaux outils.

Github va également nous permettre de communiquer directement avec Trello, notre second outil.

## Trello :

Trello est un logiciel permettant d’organiser nos différentes tâches, de les assigner à chaque développeur et de visualiser l’évolution de notre projet. Grace à la fonctionnalité Github Power-Up de Trello, il nous est permis de lier notre compte Github à trello. On pourra ainsi joindre un commit, une pull-request et en voir son état d’avancement.

Mais pourquoi Trello et pas un autre ?

Parce que Trello comptabilise plus de 10 millions d’utilisateurs dans le monde et est utilisé par des entreprises renommées, telles que Paypal, Google, Adobe, … Le logiciel est adapté pour les mobiles, point important pour visualiser à n’importe quel moment les différentes tâches qu’il nous reste à réaliser.

# Bibliographie :