



# Choix technologiques

Projet de fin d'études

Degrève Olivier, Denuit Maxime, Dubois Corenthin, Tavernier Cedric

Chef de projet : Cedric Tavernier

Membres : Degrève Olivier, Denuit Maxime, Dubois Corenthin

## Table des matières

1.	Introduction :	1
2.	Choix technologiques :	2
2.1.	Back-end :	2
2.2.	Base de données :	3
2.3.	Front-end :	4
3.	Diagramme d'architecture :	5
4.	Outils utilisés :	6
4.1.	Git-GitHub :	6
4.2.	Trello :	6
4.3.	Visual Studio Code :	6
5.	Bibliographie :	7

Chef de projet : Cedric Tavernier

Membres : Degrève Olivier, Denuit Maxime, Dubois Corenthin

## 1. Introduction :

Nous allons vous présenter les différents choix technologiques et les outils choisis pour l'outil de gestion demandé par l'Institut Paul Lambin.

Nous vous expliquerons ensuite pourquoi nous avons pris le parti de faire un Back-end en Node.js, un Front-end en JavaScript, HTML 5, CSS3 et JQuery et notre choix d'utiliser une base de données en PostgreSQL.

Nous avons décidé d'utiliser les outils suivants pour notre projet : Git-GitHub pour le partage de données, Trello pour l'organisation du projet et Visual studio Code pour l'implémentation.

## 2. Choix technologiques :

### 2.1. Back-end :

Nous utiliserons un back-end en Node.js. L'avantage de cette technologie est que nous avons les ressources nécessaires dans notre groupe pour l'utiliser et que la documentation est complète et libre d'accès sur internet. Une autre chose intéressante à propos de Node.js est sa grande modularité, ce qui nous permettra d'adapter le produit tout en maîtrisant les coûts. Il existe beaucoup de bibliothèques externes permettant d'effectuer des tâches pour un back-end - que ce soit des bibliothèques serveurs, de requêtes HTTP/HTTPS ou autres. Ce même genre de bibliothèque existe aussi pour le Front-end afin d'avoir une interface simple à mettre en place (Exemple : AngularJS) ou d'avoir des fonctions en temps réel grâce aux WebSockets de Node.js.

Si l'on compare Node.js au Java pour le Back-end, Node.js présente clairement plusieurs avantages. D'abord, l'utilisation de Node.js 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'éventuelles données transmises par le Front-end. Node.js permet aussi d'éviter les problèmes de conversion et de complexité du programme. En effet, si nous avions utilisé le Java cela aurait pu causer des problèmes de conversion et il nous aurait également fallu convertir le JSON avec Genson. Ensuite, Node.js a l'avantage d'être plus rapide que Java pour faire de l'Input/Output et ne bloque pas d'éventuelles requêtes supplémentaires. En effet, il y a une bonne gestion de la concurrence en Node.js, ce qui permet de pouvoir effectuer des requêtes Input/Output en parallèle. Bien sûr, il est possible d'implémenter de la concurrence en Java mais cela requiert plus de temps et engendre potentiellement plus de problèmes s'ils sont mal implémentés.

Node.js offre aussi l'avantage d'avoir une très bonne scalabilité. En effet, on peut déployer rapidement une application Node.js pour une entreprise - que ce soit une application web ou mobile. De plus, Node.js présente aussi l'avantage de présenter de très bonnes performances notamment en ce qui concerne la rapidité d'exécution des requêtes.

Il est à noter que Node.js sert principalement à effectuer des tâches peu complexes. Donc, si l'application requiert des calculs complexes, il serait dès lors préférable de donner la charge à un autre processus Node.js. Or, ce problème n'est pas rencontré en Java car les tâches complexes peuvent y être exécutées en un seul processus.

Niveau structure, Node.js 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.

Chef de projet : Cedric Tavernier

Membres : Degrève Olivier, Denuit Maxime, Dubois Corenthin

## 2.2. Base de données :

Nous avons choisi pour l'instant de faire une base de données en SQL et non en NoSQL. Durant la rencontre avec le client, il nous a été demandé d'implémenter un outil de gestion. Nous supposons que l'outil de gestion implique de nombreuses relations entre les différentes données, ceci explique le choix en SQL. Nous n'envisageons pas NoSQL car l'implémentation d'une base de données relationnelle dans ce langage est possible mais toutefois complexe.

Si nous faisons le choix du SQL, nous avons pris une base de données PostgreSQL, ce qui présente l'énorme avantage de pouvoir gérer un grand nombre de données. En outre, un support important existe et PostgreSQL étant une technologie libre, les coûts pour la société seront plus faibles (car il n'est pas nécessaire d'avoir une licence commerciale). Nous avons également écarté le choix de PgSQL car cela ne convient que pour de petits sites.

Si finalement notre choix devait quand même se porter sur le NoSQL, nous ferions alors le choix d'aller vers du MangoDb qui a comme avantage de pouvoir stocker d'importantes données et d'avoir une performance élevée.

Chef de projet : Cedric Tavernier

Membres : Degrève Olivier, Denuit Maxime, Dubois Corenthin

### 2.3. Front-end :

Comme langage de programmation pour le front-end nous avons décidé d'utiliser le JavaScript pour plusieurs raisons.

Premièrement, nous avons décidé d'utiliser le JavaScript car il permet d'agir directement et ne doit 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'interprétation comme par exemple avec Flash Player. En outre, le JavaScript n'occupe pas une grande place sur les disques des sites web. Nous utiliserons aussi une bibliothèque JavaScript JQuery qui nous assurera une compatibilité multi-navigateurs, une simplification et une normalisation des scripts.

Deuxièmement, comme nous avons décidé d'utiliser le Node.js comme langage back-end car cela nous permet de travailler en « full-stack », c'est-à-dire de pouvoir programmer toute l'application dans un même langage et ne pas devoir utiliser un langage intermédiaire pour rendre compatibles nos données entre les différents langages.

Pour la mise en forme du site nous avons décidé d'utiliser l'HTML5 (HyperText Markup Language 5) et le CSS3 (Cascading Style Sheet 3).

Nous avons choisi la dernière version de ces deux langages car ceux-ci nous permettent d'en utiliser toutes les dernières fonctionnalités.

Toutefois, un inconvénient mineur dont il faudra tenir compte est que certains navigateurs ne prennent pas en charge la dernière version de l'HTML5. Pour régler ce problème, nous devons donc coder en fonction du navigateur le moins « up-to-date ».

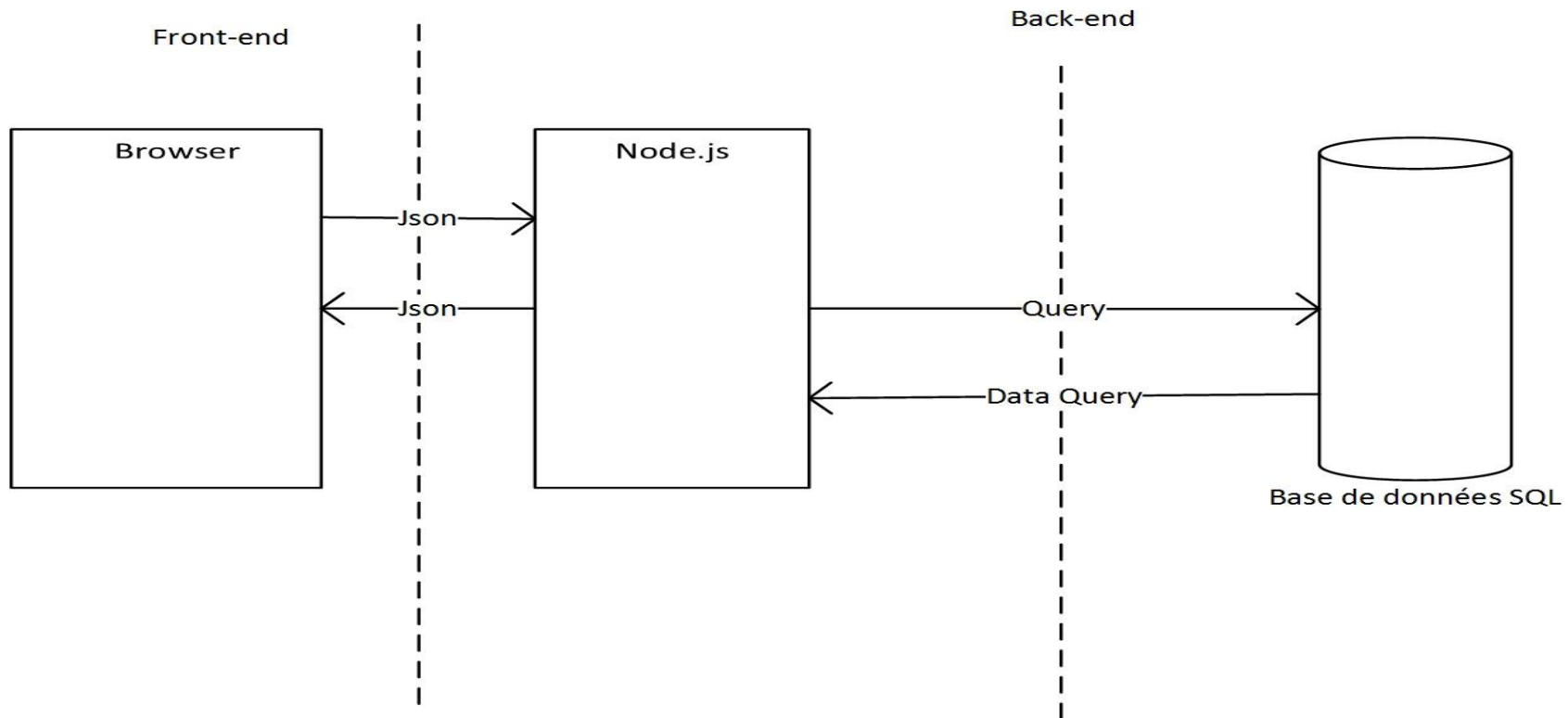
Pourquoi n'allons-nous pas utiliser Bootstrap ?

Il nous semble intéressant de pouvoir créer de l'HTML / CSS « fait maison », ce qui nous permettra d'utiliser facilement notre HTML/CSS dans nos codes JAVASCRIPT, avec des balises/identifiants qui sont clairs et lisibles. De ce fait, il nous a semblé intéressant de créer nous même un site responsif et de ne pas utiliser un Template tout fait, comme proposé par Bootstrap.

Chef de projet : Cedric Tavernier

Membres : Degrève Olivier, Denuit Maxime, Dubois Corenthin

### 3. Diagramme d'architecture :



## 4. Outils utilisés :

Pour mener à bien ce projet, trois outils distincts vont être utilisés : Git-GitHub, Trello et Visual Studio Code.

### 4.1. Git-GitHub :

Git-GitHub en est le premier. D'une part, l'hébergement de projets et de dépôts, est pris en charge par le site web GitHub. Ce site web 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. Ensuite, il y aura moyen de « merger » les fichiers de tous les collaborateurs du projet en un travail unique.

Pourquoi s'être tourné vers Git-GitHub ?

Nous nous sommes tournés vers Git-GitHub car elle est considérée actuellement 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 ses services mais aussi pour y ajouter de nouveaux outils. De plus, GitHub va également nous permettre de communiquer directement avec Trello, notre second outil.

### 4.2. 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. Grâce à la fonctionnalité GitHub Power-Up de Trello, il nous est permis de lier notre compte GitHub à Trello. Nous pourrions ainsi joindre un « commit », « un pull-request » et en voir son état d'avancement.

Mais pourquoi le logiciel Trello et pas un autre ?

Nous avons choisi Trello car ce logiciel 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 aux mobiles, point important pour visualiser à n'importe quel moment les différentes tâches qu'il nous reste à réaliser.

### 4.3. Visual Studio Code :

Nous avons pris comme décision de développer sur Visual Studio Code car il est présent sur tous les systèmes d'exploitation, il est gratuit, il possède un système de terminal intégré et est disponible chez le client. C'est donc un outil extensible et personnalisable.



## 5. Bibliographie :

### 1) Documentation de la base de données :

- <http://blog.postgresql.fr/index.php?post/drupal/216>
- <https://www.mongodb.com/fr>

### 2) Documentation du Back-end :

- <http://www.infoworld.com/article/2975233/javascript/why-node-js-beats-java-net-for-web-mobile-iot-apps.html>
- <https://www.quora.com/What-are-the-advantages-of-using-Node-js>
- <https://blog.xervo.io/top-10-reasons-to-use-node>

### 3) Documentation Front-end :

- <http://www.punkchip.com/why-dont-you-use-bootstrap/>
- <http://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html>
- <https://www.nczonline.net/blog/2013/10/07/node-js-and-the-new-web-front-end/>

### 4) Documentation d'outils utilisés :

- <http://www.journaldunet.com/web-tech/developpeur/1185911-github-le-reseau-social-des-developpeurs/>
- <http://learnosm.org/fr/advanced/github-sharing/>