

Analyse des langage et Framework

Ce document a pour objectif de présenter et d'aider à choisir parmis les différents langages et frameworks retenus pour le projet de gestion d'inscription. La présentation inclura les avantages et inconvénients pour le projet en prenant en compte les connaissances de l'équipe.

Les langages présentés seront Node.js, Java EE et PHP. Les frameworks aborderont la communication avec la base de données et la partie web (CSS et JavaScript).

Langage

Description

Commençons par comparer par plusieurs points les langages :

Critère	Java EE	PHP	Node.js
Type	Statique, fort	Dynamique, faible	Dynamique, faible
Multiplateforme	Oui	Oui	Oui
Compatibilité ascendante	Oui	Non	Oui
Pré-requis	JVM et serveur d'application (Tomcat ou Glashfish)	Interpréteur PHP et serveur web (Apache,...)	Node.js, serveur web (Apache,...)
Lourdeur de mise en place	Moyen	Faible	Faible
Environnement	IDE (Netbeans, Eclipse,...)	Editeur de texte, IDE	Editeur de texte, IDE
Connaissance de l'équipe	Connu par la plupart des membres. Architecture connu et maitrisé par ces membres	Connu par deux membres, les autres ont les bases	Langage connu par la moitié de l'équipe mais aucune architecture connu
Facilité à apprendre	Moyen	Facile	Moyen

Il en ressort que PHP et Node.js serait plus simple à mettre en place que Java (pas besoin de serveur d'application, une modification de code est vérifiable par simple rafraichissement). Par contre Java étant très structuré il est plus simple de détecter les erreurs :

- Un bug en PHP n'est pas bloquant (peut bloquer un peu l'affichage) mais il est plus difficile de cerner un bug et de le corriger. De plus la qualité des logs est bien plus faible que celle des autres langages.
- Un bug en Java est bloquant, la page ne s'affiche pas mais affiche l'erreur permettant une correction plus facile. Le développement est toutefois un peu plus long nécessitant de redémarrer le serveur d'application là où le PHP et le node.js nécessite un simple rafraichissement.
- Un bug en Node.js est bloquant et permet une correction rapide via un affichage de l'erreur dans la console et une correction à vérifier via un simple rafraichissement de la page

Pour le moment on peut retenir que Java est le langage le plus connu dans l'équipe, le plus facile à organiser (bonne connaissance des architectures) mais le plus lourd à développer. PHP est peu connu

mais facile à apprendre néanmoins il est difficile de détecter et corriger les erreurs. Enfin Node.js est plutôt connu mais sans connaissance de l'architecture il deviendra difficile d'organiser, le développement sera par contre assez rapide.

Impact machine

Au niveau de l'impact machine, PHP consommera moins que le JEE (peut s'inverser si l'application demande beaucoup de calculs ce qui n'est pas le cas ici). Node.js a moins d'impact que le PHP sur la machine.

Performance

En terme de performance Java 8 a permis d'alléger énormément JEE et les besoins de serveur d'application ont relativement fondu dans le temps, les problèmes de forte consommation de mémoire seraient résolus. Dans le cas de PHP c'est plutôt l'inverse. Enfin dans le cas de Node.js il est plus rapide que PHP pour faire des calculs. Au vu du projet qui ne nécessitera qu'un accès base de données et un affichage de page sans réel calcul les performances doivent avoir peu d'impact sur la décision du langage.

Sécurité

Même si on ne peut parler de failles de sécurité dans les langages, les utilisateurs de PHP et Node.js savent bien qu'il est plus difficile de se prémunir des attaques car le langage ne force pas à se sécuriser.<

De son côté JEE, valide à chaque étape les données de telle sorte qu'il est difficile de faire des attaques de types SQL injection par exemple. La relative lourdeur de JEE apporte un gain en sécurité très important, le contrôle et la validation des données n'étant plus laissé à la simple discrétion du développeur mais est directement incluse au cœur du langage. Cet atout sécurité est indéniablement un plus pour le monde JEE.

Conclusion

Node.js et Java sont les deux langages les plus intéressants, ils sont connus par plus de membres de l'équipe même si Java est bien mieux maîtrisé. Node.js permettra de développer plus rapidement mais Java permettra un développement plus stricte et donc plus facile en équipe. Dernier point pour Node, sa rapidité est assurée par des fonctions asynchrones qui nécessitent un temps d'adaptation.

Frameworks base de données

PHP

Le seul framework connu par ceux qui maîtrisent le langage est celui de mysql. Il est techniquement simple à apprendre et fonctionne comme de simple requête SQL.

Le framework nécessite de relier la partie DAO à la partie Entité mais reste globalement maîtrisé par tous les membres (inutile de connaître un langage pour connaître la procédure de cast d'un tableau vers objet)

Node.js

Même fonctionnement que PHP, à la différence que la partie Object est beaucoup plus difficile à mettre en place et probablement pas maîtrisée par la plupart des membres.

Le problème de Node est que les connaissances de l'équipe sont en autodidacte et pas suite à un cours structuré, néanmoins il est toujours possible de s'approcher d'une architecture type Java EE avec un peu d'apprentissage.

Java

Au niveau de Java deux frameworks sont possible JDBC et JPA. JDBC est très similaire au frameworks de PHP et Node.js et globalement maîtrisé.

JPA est un frameworks qui est maîtrisé par peu de membres de l'équipe (environ 4) mais qui accélérerait grandement le développement dû à une gestion des requêtes en Objet qui permet une détection d'erreur directement via l'IDE, cette détection permet aussi de plus facilement modifier l'architecture de la base de donnée puisqu'il n'est pas nécessaire de touché à toutes les requêtes.

En terme d'apprentissage il y a deux solutions :

- On apprend à l'équipe à ne faire que des requêtes et on confie la partie annotations aux membres déjà connaisseurs.
- On apprend la totalité de l'interface à l'équipe ce qui prendrait plus de temps mais les rendrait plus efficace en cas de problème

Frameworks web

Les connaissances purement web c'est-à-dire HTML, JavaScript et CSS de l'équipe sont globalement faible. La meilleure solution serait de se tourner vers des frameworks et des modules qui exécuteront une grande partie du travail sans forcément en connaître le fonctionnement.

Bootstrap

Bootstrap est une frameworks qui fournit un CSS, plusieurs script JQuery (une bibliothèque JavaScript) et une police d'écriture. Son avantage est une compatibilité accrue avec la plupart des navigateur (tout navigateur récent et Internet explorer 8 maximum) tout en proposant une un CSS entièrement responsive design.

Il est assez simple d'utilisation mais nécessite d'avoir accès à la documentation.

Module JQuery

JQuery est une bibliothèque Javascript conçu pour faciliter l'écriture de script coté client.

Concrètement JQuery permet une utilisation simplifier de Javascript. Dans le cadre de ce projet l'intérêt ne serait pas d'écrire nos propres scripts mais plutôt de s'appuyer sur des modules pour faciliter la mise en page, exemple avec le module dataTables :

Show

10

▼
entries

Search:

Name	↓↑	Position	↑↓	Office	↑↓	Salary	↑↓
Airi Satou		Accountant		Tokyo		\$162,700	
Angelica Ramos		Chief Executive Officer (CEO)		London		\$1,200,000	
Ashton Cox		Junior Technical Author		San Francisco		\$86,000	
Bradley Greer		Software Engineer		London		\$132,000	
Brenden Wagner		Software Engineer		San Francisco		\$206,850	
Brielle Williamson		Integration Specialist		New York		\$372,000	
Bruno Nash		Software Engineer		London		\$163,500	
Caesar Vance		Pre-Sales Support		New York		\$106,450	
Cara Stevens		Sales Assistant		New York		\$145,600	
Cedric Kelly		Senior Javascript Developer		Edinburgh		\$433,060	

Showing 1 to 10 of 57 entries

Previous

1

2

3

4

5

6

Next

Le module fournit un tableau qui peut être en responsive avec un champ de texte qui filtre à la volée, une pagination, ...

Il peut être intéressant de s'appuyer sur les modules pour certaines tâches (tableau, choix d'une date dans un calendrier...).

Impact de l'apprentissage sur le projet

Etant donné que chaque membre n'a pas le même niveau voir ne connaît presque pas les langages il est important de notifier combien de temps prendrait la maitrise de chaque langage et frameworks. Commençons par lister combien de membres connaissent chaque partie :

Critère	Java EE - JPA	Java EE - JDBC	PHP	Node.js
Langage	8	8	2	Environ 5-6
Frameworks BD	4	8	2	Environ 5-6
Bootstrap				
JQuery				

A présent le temps qu'il serait probablement nécessaire pour expliquer chaque partie :

Critère	Java EE - JPA	Java EE - JDBC	PHP	Node.js
Langage	2h	2h	1h	1h30
Frameworks BD	1h si annotation, 30mn si seulement partie requête	30mn	30mn	30mn
Bootstrap (présentation partie CSS)	20mn	20mn	20mn	20mn
JQuery (si simple description)	15mn	15mn	15mn	15mn
Total	3h35 ou 3h05	3h05	2h05	2h35

Parmi ces horaires il faut prendre en compte le temps d'adaptation et de blocage. PHP est rapide à apprendre mais peu de personne le connait, il sera par conséquent long et difficile de débloquer les utilisateurs.

Dans le cas de Node.js le langage est facile à apprendre mais nécessitera un temps d'adaptation de par sa structure interne (fonctions asynchrones, gestion d'objet, ajout de fonctionnalité par module, ...).

Conclusion

PHP correspond plutôt bien à ce type de projet (petit projet web) mais il n'est pas assez connu ce qui risque de rapidement poser problème en terme de mauvaise odeurs, de plus le fait qu'il soit peu connu risque de causer beaucoup de blocage assez long de la part des membres de l'équipe (du fait également que seulement deux membres connaissent ce langage pour débloquer les autres).

Node.js est lui moyennement connu, il risque également de poser des blocages au début en particulier au niveau des fonctions qui sont asynchrone et il sera plus difficile de relire le projet pour l'harmoniser mais s'il est bien architecturé et suivi (en terme d'intégration) il a de très bonne chance d'aboutir.

Java EE est plutôt bien connu, bien architecturé et en choisissant JPA la vitesse de développement devrait rattraper les retards infliger par le serveur d'application. Néanmoins il nécessitera une grande vigilance en terme de suivi et particulièrement au niveau de JPA qui risque de poser des blocages au début.

Comme Node.js il sera nécessaire lors des premières phases de code de bien vérifier le respect de l'architecture et de la bonne compréhension des frameworks.