

Étude comparative des différentes technologies côté serveur

CODEV

BENALI Myriam – NAAJI Dorian
POLYTECH LYON
5A INFO GROUPE 2, ÉQUIPAGE 2

1.INTRODUCTION

Ce document a pour but de préciser ce qui a motivé nos choix technologiques pour la partie serveur de l'application M' Ton Air, qui sera une API permettant d'interroger aqicn.org et de fournir des données à la partie cliente.

2.LES TECHNOLOGIES

2.1. Python

Python est un langage puissant de calcul scientifique, mais il permet également la programmation orientée objet et le développement d'APIs, notamment grâce aux framework Flask et Django. C'est d'ailleurs le 3^{ème} framework le plus utilisé pour des APIs [3]. Malgré cela, nous n'avons pas d'affinités particulières avec ce langage et souhaitons le mettre de côté dans notre étude et donc pour notre projet.

2.2. Node.js

Node.js pour du JavaScript côté serveur est une option intéressante, qui nous aurait permis d'apprendre les rouages de Node pour ce projet. Néanmoins, le JavaScript reste un langage utilisé beaucoup utilisé pour les navigateurs.

Outre cela, nous ne disposons pas du temps nécessaire afin de nous plonger dans ce langage pour de la programmation côté serveur. Malgré tout, il a retenu notre attention ; Javascript avec Node offre énormément de possibilités, et le langage est doté de particularités qui le rendent unique par rapport aux autres langages côté serveur. Notre choix se prédestinait initialement vers cette technologie.

2.3. PHP

PHP est un langage côté serveur permettant de réaliser des applications web ou de traitements côté serveur. C'est selon rapidapi le langage le plus utilisé pour développer des APIs [3].

En revanche, nous trouvons que le code produit à PHP sans framework ressemble très vite à du bricolage si les bases ne sont pas bien posées dès le début. Avec framework, les choses sont différentes, mais nous concevons dans ce cas PHP avec Laravel par exemple plus comme un outil pour la création d'applications web métier, avec des interfaces utilisateur. Nous mettrons de côté ce langage.

2.4. C#

Le C# grâce au framework ASP.NET est un très bon compris pour développer des applications web en tout genre, dont des APIs grâce à un système de route très complet semblable à PHP et Spring Boot.

2.5. Java

De même que pour le C#, à la seule différence que le Java nous a semblé être une valeur plus sûre et est multiplateformes. L'ensemble de l'équipe est également plus compétente dans ce langage que dans tous les autres. C'est pourquoi nous retiendrons ce choix plutôt que C#.

3.CONCLUSION

Notre choix pour le développement côté serveur se tourne donc vers le Java. Nous avons une connaissance approfondie du langage grâce à tous les cours réalisés à Polytech Lyon, et nous avons pu utiliser quelques frameworks facilitant le développement, comme Spring Boot, Hibernate, etc.

4.SOURCES

[1] : [Overview of RESTful APIs Description Languages](#)

[2] : [IBM, REST APIs.](#)

[3] : [Most Used Programming Languages for APIs on RapidAPI](#)

[4] : [PHP APIs Frameworks](#)

[5] : [Best language for RESTful APIs](#)