

Projet de Programmation

Chat entre passagers de voitures

Choix des technologies

Thomas Blot
Corentin Drezen
Leo Favre
Mouhamadou Mansour Gueye
Mohamed Larinouna

8 février 2024

1 CA - Script

Pour la mise en place d'un serveur et d'une autorité de certification nous avons choisi l'utilisation de python avec comme outils (soit OpenSSL, soit certbot deux outils facilement exploitable car intégré à python pour la mise en place du CA). Grâce à ces outils nous allons pouvoir mettre en place une architecture légère et cohérente. Il existe d'autre outils permettant d'obtenir le même résultat mais voici les contraintes que nous avons :

- Documentation fournis / Outil utilisé par la communauté scientifique (pour pouvoir demander de l'aide si nous rencontrons des difficultés).
- Utilisable sur un large choix de machine (pas forcément une distribution précise).
- Léger
- Intégration facile sur un serveur
- Gratuit

Après réflexion seul deux outils répondent à ces besoins, OpenSSL et Certbot. Nous allons donc utilisé OpenSSL qui est le plus connue et possède une bibliothèque python PyOpenSSL ce qui sera plus facile à utiliser avec notre serveur en python.

En ce qui concerne le choix entre Django et Flask, ainsi que celui entre NGINX et Apache pour la gestion des requêtes HTTP et HTTPS, nous avons pris la décision stratégique d'opter pour Django en tant que framework web et NGINX en tant que serveur web.

Django a été choisi en raison de sa robustesse et de sa structure plus complète. En effet, Django offre une architecture MVC robuste, des fonctionnalités intégrées pour la gestion de bases de données ce qui dans notre cas est crucial. Ces caractéristiques permettent une développement rapide et une maintenance simplifiée.

Concernant le choix du serveur web, nous avons opté pour NGINX en raison de ses avantages significatifs. NGINX excelle dans la gestion multithread des demandes de connexion, ce qui se révèle particulièrement avantageux lorsqu'il s'agit de gérer de multiples connexions simultanées. Sa conception légère et sa capacité à gérer efficacement un grand nombre de connexions en font le choix idéal pour notre application, assurant ainsi une performance optimale même dans des conditions de charge élevée.

2 BDD

SQLite

- Légèreté et simplicité, SQLite est une solution légère, car elle ne nécessite pas de serveur de base de données séparé. Les bases de données sont stockées dans un fichier unique, ce qui simplifie la gestion des données et la configuration du système.

- Puisqu'une base de données SQLite est stockée dans un seul fichier, le déploiement est aussi simple que de copier ce fichier d'un endroit à un autre.

Cela est donc particulièrement utile pour les applications mobiles.

Firebase

- Firebase offre Cloud Firestore et Firebase Realtime Database, deux bases de données cloud qui permettent la synchronisation des messages en temps réel entre les utilisateurs. Cela garantit que les messages sont reçus instantanément, ce qui est crucial pour une expérience utilisateur fluide dans une application de chat.

-

3 Application

Dans le cadre du développement de l'application de ce projet, le choix entre Java et Kotlin s'est avéré être une décision cruciale. Bien que ces deux langages présentent des similitudes, certaines différences significatives ont été identifiées, comme exposé dans l'article [2].

En ce qui concerne Kotlin, il se distingue par une efficacité supérieure en termes de nombre de lignes de code nécessaires. Cette caractéristique permet d'écrire des codes plus concis par rapport à Java, ce qui se traduit par un processus de développement logiciel plus rapide. De plus, Kotlin tend à demander moins de ressources, de données, ou d'espace mémoire que Java pour accomplir les mêmes tâches, entraînant ainsi une utilisation plus efficace de la mémoire et des ressources. Ces avantages se traduisent directement par une amélioration des performances de l'application et une réduction des erreurs liées à la gestion des données.

En revanche, du point de vue de la compilation, Java présente un avantage lors de la première construction d'une application par rapport à Kotlin. Toutefois, lors de constructions incrémentielles, où des modifications sont apportées à un code existant, Kotlin équivaut à Java en termes de temps de compilation.

En synthèse, le choix entre Java et Kotlin dépend des priorités du projet. Java peut être privilégié pour le développement sur Android si l'accent est mis sur la taille de l'application et le temps de compilation/construction. En revanche, Kotlin pourrait être préféré si l'objectif est d'avoir un code plus concis, moins sujet aux erreurs et plus rapide à développer.

L'article [1] apporte un éclairage supplémentaire sur le choix entre Java et Kotlin. En tant que langage moderne suscitant un intérêt croissant parmi les développeurs et étant étroitement lié à Java, Kotlin gagne aujourd'hui en préférence, que ce soit pour les mises à jour de logiciels existants ou le développement de nouvelles applications. Cette tendance est étudiée de manière approfondie dans l'article, qui recueille les opinions de 78 développeurs ayant une expérience dans l'utilisation des deux langages.

Les conclusions les plus pertinentes de leur enquête sont les suivantes : a) Éviter les limitations des versions de Java, soulignant la nécessité de solutions plus flexibles. b) Produire un code plus sûr, mettant en avant la réputation de Kotlin pour améliorer la sécurité du code par rapport à Java. c) Utiliser un langage de programmation moderne avec de nouvelles fonctionnalités, reflétant

l'attrait de Kotlin pour ses caractéristiques modernes. d) Suivre la direction de Google, soulignant l'impact majeur de la prise en charge officielle de Kotlin par Google, renforçant sa préférence dans le développement Android. e) À des fins d'apprentissage, pour se former à un nouveau langage gagnant en popularité aujourd'hui.

Pour ce projet, l'accent est mis sur le développement, et le but est d'avoir un produit qui fonctionne à la fin. Le poids de notre application n'est pas un facteur que nous prenons en compte dans ce projet. Ainsi le choix de Kotlin pour le développement de ce projet semble plus pertinent.