

[Skills Test] Backend Engineer

Introduction

Merci par avance pour le temps que vous allez consacrer à cet exercice. Via cet exercice nous voulons avant tout évaluer vos méthodes de travail et votre raisonnement.

Prenez tout le temps qu'il vous semblera nécessaire pour effectuer ce projet, quand vous aurez fini, vous pouvez soit nous envoyer votre travail dans **une archive compressée**, soit nous envoyer un lien vers un **dépôt de code public**. Dans les deux cas, envoyez les informations nécessaires à l'adresse email

engineering+backend_interview@join-jump.com.

Sujet

Chez Jump, nous essayons d'automatiser au maximum les tâches administratives pour faciliter la vie de nos clients.

Tous nos clients facturent leurs prestations et le suivi de ces factures est une tâche longue et fastidieuse, mais nécessaire car c'est le règlement de ces factures qui leur permet par la suite l'avoir l'argent suffisant pour se verser un salaire.

Afin d'automatiser cette tâche vous allez devoir implémenter un service web capable de recevoir des paiements, de les relier à une facture et de créditer la balance financière des utilisateurs du bon montant.

Instructions

Votre service devra être écrit avec le langage **Golang**.

Vous êtes libre d'utiliser les libraires de votre choix.

Vous pouvez également fournir un document texte avec votre exercice (README, docx, google docs ...) pour décrire vos choix d'implémentation, votre approche, les problèmes que vous avez rencontrés, ou les améliorations que vous auriez aimé faire avec plus de temps. N'hésitez pas non plus à fournir des instructions sur comment lancer votre projet.

La qualité du code doit être d'un niveau qui vous semble suffisant pour mettre votre

projet en production.

Votre service devrait exposer au minimum les 3 routes suivantes :

- Une route **GET** sur l'url **/users** qui renvoie la liste des utilisateurs dans un JSON au format suivant:

```
[
  {
    "user_id": 12,
    "first_name": "Kevin",
    "last_name": "Findus",
    "balance": 492.97
  },
  {
    "user_id": 22,
    "first_name": "Laura",
    "last_name": "Biding",
    "balance": 5200.60
  }
]
```

- Une route **POST** sur l'url **/invoice** pour créer une facture, qui prend en entrée un JSON au format suivant :

```
{
  "user_id": 21,
  "amount": 113.45,
  "label": "Work for April"
}
```

En cas de succès, la route doit retourner un code **204** , dans les autres cas un code d'erreur approprié.

- Une route **POST** sur l'url **/transaction** qui permet d'ajouter une transaction monétaire pour une facture donnée, qui prend en entrée un JSON au format suivant :

```
{
  "invoice_id": 42,
  "amount": 956.32,
  "reference": "JMPINV200220117"
}
```

Lors d'un appel à cette route, si une facture avec le bon identifiant et le bon montant est trouvée, la balance financière de l'utilisateur possédant cette facture doit être créditée du montant de la facture et le statut de la facture doit être modifié.

Une transaction pour une facture n'est valide que si le montant de la transaction correspond exactement au montant de la facture.

En cas de succès, la route doit retourner un code **204**

Si aucune facture n'est trouvée la route doit retourner un code **404**

Si le montant de la transaction ne correspond pas au montant de la facture la route doit retourner un code **400**

Si la facture est déjà payée la route doit retourner un code **422**

Un schema de base de données est disponible dans le dossier database de ce repository github : <https://github.com/Freelance-launchpad/backend-interviews>

Un Dockerfile est également disponible pour vous aider à lancer la base de données et tester votre projet, pour lancer la base de données :

```
$ git clone https://github.com/Freelance-launchpad/backend-interviews.git
$ cd backend-interviews/database
$ docker build . -t jump-database
$ docker run -p 5432:5432 jump-database
```

Vous pouvez également utiliser votre propre schéma de base de données si vous le souhaitez, si c'est le cas pensez à nous fournir un dockerfile ou les instructions nécessaires pour que nous puissions lancer la base de données de notre côté.

Good luck !