

PRÉSENTATION

Ce test a pour but d'évaluer votre niveau de connaissances, d'adaptation, de bonnes pratiques et de logique en développement logiciel. Le test propose d'implémenter plusieurs fonctionnalités, libre à vous de proposer votre approche. Le but n'est pas forcément de finir d'implémenter complètement toutes les fonctionnalités décrites, mais d'essayer de bien faire ceux dont vous pouvez. Vous serez évalué sur la qualité de votre code et votre compréhension du sujet.

Vous trouverez ci-dessous quelques liens utiles pour vous orienter.

BackEnd

Pour le BackEnd vous devez créer une API REST en Golang, permettant d'enregistrer des données dans une base de données **MongoDB**. Vous devrez utiliser le routeur **Gin Gonic**. L'utilisation de **Goroutines** est obligatoire, les **channels** sont conseillés mais vous pouvez utiliser les **WaitGroup**, notamment pour les parties sérialisation/désérialisation et encryption/decryption. Aidez vous du fichier **dataset** fournir.

Cette API devra supporter les 4 fonctions **CRUD**.

1. Create

- Requête: POST /add/users

Description: Le fichier d'input JSON doit être sérialisé puis enregistré dans une base de données MongoDB. Le fichier JSON contiendra un tableau d'utilisateur. Vous devrez ensuite désérialiser les données de façon concurrente puis les insérer en base sans traiter à nouveau les entrées déjà insérées. Le mot de passe doit être crypté avec **bcrypt** et le hash de celui-ci inséré en base.

En plus de l'insertion en base de ces données, vous devrez générer un fichier par entrée avec comme nom de fichier l'**id** de l'entrée, ce fichier devra contenir uniquement le champ **Data**.

2. Login

- Requête: Post /login

Description: L'utilisateur doit pouvoir se connecter sur un son profil afin de pouvoir accéder à toutes les données qui lui sont attribuées. On considère que l'utilisateur à un id et un mot de passe.

3. Delete

- Requête: DELETE /delete/user/:id

Description: Dois supprimer un user avec son Id, ainsi que le fichier généré.

4. Read

- Requête: GET /users/list
- Requête: GET /user/:id

Description: Dois récupérer un utilisateur avec son id ou une liste d'utilisateurs.

5. Update

- Requête: UPDATE /user/:id

Description: Dois modifier un utilisateur avec son id, si le champ data change le fichier doit être modifié.

Liens utiles:

<https://godoc.org/golang.org/x/crypto/bcrypt>

<https://github.com/mongodb/mongo-go-driver>

<https://github.com/gin-gonic/gin>