

Test technique - Traitement de données en parallèle

Objectif

Le but de ce test est de montrer votre capacité à réaliser un utilitaire utilisant des ressources externes de type API HTTP et de traiter les résultats de ces ressources en parallèle. Ces résultats devront être servis de manière simple via une API JSON. Il vous sera également nécessaire de vous familiariser avec le langage de programmation Go, fortement utilisé dans l'infrastructure de Scalingo.

Sujet du test

À partir du canvas d'application suivant :

<https://github.com/Scalingo/scInng-backend-test-v1>

Vous devez réaliser un logiciel proposant une API JSON permettant de réaliser les tâches suivantes :

- Renvoi des données à propos des 100 dernier dépôts GitHub publics
- Permet de limiter les résultats en fonction de différents paramètres (langage, licence, etc.)
- Une attention particulière doit être apportée sur les performance et la mise à l'échelle du projet. Le projet doit répondre le plus rapidement possible et pouvoir être utilisé par le plus grand nombre d'utilisateurs simultanés possible.
- Toute disposition mise en place pour rendre le projet maintenable dans le temps est un plus.

Récupération des données

Pour récupérer les données sur les dépôts, vous utiliserez l'API REST publique de GitHub : <https://docs.github.com/en/rest?apiVersion=2022-11-28>

- Traiter en parallèle les résultats pour récupérer les informations concernant les langages de programmation qu'ils utilisent
- Filtrer les données par rapport à la recherche faite par l'utilisateur
- Agréger les données et afficher les statistiques des dépôts Git avec le langage choisi.

Exemple d'affichage final :

```
{
  "repositories": [
    {
      "full_name": "FreeCodeCamp/FreeCodeCamp",
      "owner": "FreeCodeCamp/FreeCodeCamp",
      "repository": "FreeCodeCamp",
      "languages": {
        "javascript": {
          "bytes": 123456
        }
      }
    },
    // ...
  ],
  [
    //...
  ]
}
```

Modalité de rendu

- Le langage employé doit être 'Go'
- Envoi d'un lien vers le dépôt GitHub public contenant le code de l'exercice

- Il doit y avoir un fichier README.md qui explique l'architecture du projet et toute prise de décision qui a été réalisée

Évaluation

L'évaluation se fera sur vos capacités à répondre au sujet, dans la qualité du code rendu, ainsi que sa documentation.

Ressources

- <https://go.dev>
- <https://docs.github.com/en>

PS : merci de me confirmer de la bonne réception de cet email.