

Traitement parallèle de données

Objectif

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

Sujet du test

Vous devez réaliser un utilitaire réalisant les taches suivantes:

Serveur web

- Lancer un serveur web qui répond aux requêtes suivantes:
- `GET /` → Affiche une page avec un champ de texte et un bouton de recherche
- `GET /search` → Page accédée depuis le formulaire précédent affichant une liste de résultats définie dans le paragraphe suivant.

Récupération des données

- Récupérer 100 derniers dépôts GitHub publiques en utilisant leur API
- 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
- Aggréger les données et afficher les statistiques des dépôts GIT avec le langage choisi.

Objectif Bonus

- Limiter à 10 le nombre de traitements parallèles
- Les résultats finaux doivent s'afficher de manière triée, du dépôt avec le plus de lignes de code à projet avec le moins de ligne de code.
- En plus d'une liste d'entrées, afficher un graphe dans la page web (lib Javascript au choix), exemple:
- Nombre de projets créé par jour (histogramme)
- Volumétrie en quantité de ligne de code des projets (camembert)

Exemple d'affichage final (sans mise en forme, les noms des dépôts doivent être des liens):

```
JavaScript: 2206279 lines
- FreeCodeCamp/FreeCodeCamp: 24331 lines
- dthree/cash: 900 lines
- facebook/draft-js: 8459 lines
...
```

Modalité de rendu

- Le langage employé doit être 'Go'
- Envoie d'un lien à jobs@scalingo.com vers le dépôt GitHub public contenant le code de l'exercice

- Il doit y avoir un fichier README.md qui explique comment fonctionne le projet
- Le programme doit être installable à partir de la commande 'go get'

Évaluation

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

Ressources

- <https://golang.org>
- <https://godoc.org>
- <https://developers.github.com>

PS. Merci de me confirmer de la bonne réception de cet email.