

# Agrisoft

## Développement Backend en TypeScript avec Cloud Function pour Google Cloud Platform (GCP)

### But

Cet examen vise à évaluer l'agilité d'un développeur backend spécialisé en TypeScript et en développement de Cloud Function pour Google Cloud Platform (GCP). L'évaluation se fera à travers la manipulation de fichiers GeoJSON et la réalisation de recherches géographiques.

### Prérequis

Avant de commencer l'examen, assurez-vous d'avoir les éléments suivants :

- Un fichier texte nommé **coordonnees.txt** joint au courriel.
- Un fichier GeoJSON nommé **plan-culture.geojson** contenant des polygones représentant des zones géographiques et leurs propriétés associées.

Préparer l'environnement d'exécution de l'exercice :

- Un compte Google Cloud Platform avec un projet configuré.
- Le service Cloud Functions activé dans votre projet GCP.
- Le service Cloud Storage configuré dans votre projet GCP.
- Un environnement de développement Node.js et TypeScript configuré.
- Avoir un compte GitHub pour le dépôt du code du projet

### Exercice

#### Configuration du projet Google Cloud Platform

##### 1. Création du projet Google Cloud Platform

- Connectez-vous à la console Google Cloud Platform.
- Créez un nouveau projet dans GCP et nommez-le : **examen-cloud-function** + « **-votre prenom** » afin de bien distinguer votre projet des autres candidats.

##### 2. Configuration des permissions IAM (Identity and Access Management)

- Créez un bucket nommé **exercice** dans Cloud Storage pour stocker les fichiers suivants : **coordonnees.txt** et **plan-culture.geojson**.
- Attribuez les droits d'accès au bucket de Cloud Storage. Plus précisément, accordez uniquement les permissions de lecture et d'écriture sur ce bucket à l'utilisateur IAM **softconceptcanada@gmail.com**.

## Programmation et déploiement d'une Cloud fonction

Vous devez développer et déployer une Cloud Function en Node.js/TypeScript qui produit le résultat suivant :

Pour chacune des lignes du fichier **coordonnees.txt** dont les coordonnées intersectent avec les géométries du fichier **plan-culture.geojson** , ajouter les propriétés des features collections suivantes : *clecomposite*, *nochamp*, *culture*, *variete*, *nosemi*, *date\_semi*.

Le fichier résultant doit se nommer **resultat.csv**.

## Test et Validation

Appelez l'URL de la Cloud Function pour déclencher son exécution et le traitement des données.

Vérifiez que le fichier de sortie au format CSV nommé **resultat.csv** a été correctement généré et sauvegardé dans le bucket **exercice** dans le Cloud Storage.

## Dépôt du code dans GitHub

Déposer le projet de développement dans GitHub et donner un accès privé à ce dépôt à **lerobindesbois@icloud.com**.

## Critères d'évaluation

Votre travail sera évalué selon les critères suivants :

- Droits d'accès
- La bonne exécution de la Cloud fonction
- Fichier résultant
- Qualité du code dans le dépôt GitHub (repository).

## Livrable

Votre courriel du livrable doit contenir :

1. Votre nom
2. Lien vers GitHub à [lerobindesbois@icloud.com](mailto:lerobindesbois@icloud.com)
3. Lien vers le projet GCP de l'exercice
4. Lien de déclenchement de la Cloud Fonction

## Délai

Si vous avez des questions sur l'exercice, veuillez les adresser par courriel à Robin Tremblay avant jeudi 27 mars à 18h.

La date limite pour la livraison de l'exercice est le vendredi 28 mars 2025 à 18h à l'adresse courriel [r.tremblay@agrisoft.ai](mailto:r.tremblay@agrisoft.ai).

Bon exercice !

Agrisoft