Projet : Module 7

Google Earth Engine (GEE) & Big Data

M2 GER
Université de Nantes -- IGARUN

Objectifs du Module 7

Nous avons automatisé le traitement local (GeoPandas). Maintenant, automatisons l'acquisition des données satellites.

PROGRAMME

- **Administration :** Créer un compte GEE et un projet Cloud (étape obligatoire).
- **Technique :** Authentifier Python auprès de Google.
- **Pratique :** Lancer une requête pour télécharger une image Sentinel-2.

1. Accès & Configuration GEE

Étape 0 : Créer un Compte

GEE n'est pas ouvert à tout le monde. Il faut s'inscrire.

> **ACTION (À FAIRE CHEZ VOUS SI PAS FAIT)**
>
> 1. Allez sur earthengine.google.com/signup
> 2. Utilisez votre compte Google.
> 3. Remplissez le formulaire (But : "Academic Research / Education").
> 4. Attendez l'email de validation (souvent immédiat).

Sans compte validé, impossible d'utiliser l'API.

Étape 0-Bis : Le Projet Cloud

Crucial

Depuis 2024, Google oblige à lier GEE à un **Projet Google Cloud** (même gratuit).

1. Une fois connecté à GEE, allez sur l'éditeur de code (code.earthengine.google.com).
2. En haut à droite, cliquez sur l'icône profil/projet.
3. Cliquez sur "**Register a new Cloud Project**".
4. Nommez-le (ex: `m2-ger-projet-2026`).
5. Sélectionnez "Unpaid usage" (Usage gratuit pour l'éducation).

> **Notez bien l'ID du projet** (ex: `m2-ger-projet-2026`), on va vous le demander dans Python.

2. Authentification Python

L'API Python

Nous utilisons la librairie officielle `earthengine-api` (nom du module : `ee`).

```python
import ee

# Vérification de l'installation
try:
    print(ee.__version__)
except ImportError:
    print("Librairie non installée")
```

Authentification (La commande qui fait peur)

La première fois, Python doit obtenir un "jeton" (token) de votre part.

```
# Lancez cette cellule UNE SEULE FOIS
ee.Authenticate()
```

CE QUI VA SE PASSER

1. Une nouvelle fenêtre de navigateur va s'ouvrir.
2. Google va vous demander de choisir votre compte.
3. Il va vous demander de sélectionner votre **Projet Cloud** (celui créé juste avant).
4. Il va vous donner un code (token) à copier, ou le valider automatiquement.

Initialisation

Une fois authentifié, il faut initialiser la session avec votre projet.

```
# Remplacez par VOTRE ID de projet
MY_PROJECT = "m2-ger-projet-2026"

try:
    ee.Initialize(project=MY_PROJECT)
    print("Connexion GEE réussie !")
except Exception as e:
    print(f"Erreur : {e}")
```

3. Requête et Export

La Logique GEE

Dans GEE, on ne télécharge pas l'image tout de suite. On envoie une "recette" au serveur.

- **ImageCollection :** Une pile d'images (ex: Sentinel-2).
- **Filter :** Sélectionner (Dates, Lieu, Nuages).
- **Reduce/Mosaic :** Compresser la pile en une seule image.

Serveurs Google

Exercice 1 : Définir la Zone

Utilisons le géocodage du Module 6 pour définir notre zone d'intérêt (ROI).

```
# Supposons qu'on ait récupéré [Lon, Lat] pour Nantes
coords = [-1.55, 47.21]

# Créer un point GEE
point_geo = ee.Geometry.Point(coords)

# Créer un buffer (cercle) de 10km autour
roi = point_geo.buffer(10000)
```

Exercice 2 : La Requête Sentinel

Demandons l'image la moins nuageuse de l'été 2023.

```
# Charger la collection
s2 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED") \
    .filterBounds(roi) \
    .filterDate("2023-06-01", "2023-09-30") \
    .filter(ee.Filter.lt("CLOUDY_PIXEL_PERCENTAGE", 10)) \
    .sort("CLOUDY_PIXEL_PERCENTAGE") \
    .first()  # Prendre la meilleure

# Sélectionner les bandes utiles (RGB + NIR)
image_finale = s2.select(["B2", "B3", "B4", "B8"])
```

L'Export vers Drive

Code

GEE ne permet pas de télécharger directement en Python facilement. Le standard est d'envoyer vers Google Drive, puis de télécharger.

```
# Créer une tâche d'export
task = ee.batch.Export.image.toDrive(
    image=image_finale,
    description="Nantes_S2_Export",
    folder="GEE_Data",
    scale=10,       # Résolution 10m
    region=roi,     # Notre zone tampon
    crs="EPSG:4326"
)

# Lancer la tâche (C'est asynchrone, ça tourne sur le serveur)
task.start()
print("Export lancé ! Vérifiez votre Google Drive dans quelques minutes.")
```

4. Conclusion & Suite

Le Workflow Complet

1. **Python (Local) :** Envoie le nom de la ville à Geopy $\rightarrow$ Coordonnées.
2. **Python (Local) :** Envoie les coordonnées à GEE.
3. **GEE (Cloud) :** Cherche l'image, découpe, prépare.
4. **Google Drive :** Reçoit le fichier GeoTIFF.
5. **Python (Local) :** Télécharge et analyse avec Rasterio (Module précédent).

Dernière étape

Nous avons automatisé l'acquisition et le traitement.
Il nous reste à assembler tous les résultats (les indicateurs de chaque ville) dans un tableau final pour produire le rapport.

> **MODULE 8 (FINAL)**
>
> Agrégation des données, Statistiques comparatives et Visualisation finale.