

Tutoriel 2

March 31, 2021

1 GEE Python API et geemap

1.1 Python API

GEE fournit un aperçu et une documentation sur l'API Python.

1.1.1 Google Colab Python API

Google fournit aussi un guide d'utilisation de l'API Python dans un notebook Google Colab.

Par rapport à l'API JavaScript et à l'éditeur de code, la documentation sur l'API Python est plutôt limitée. La bonne nouvelle est que grâce à l'utilisation de deux modules Python clés, `earthengine-api` et `geemap`, vous pouvez obtenir une grande partie de la fonctionnalité comme dans l'éditeur de code, même si vous devez utiliser la documentation JavaScript comme référence principale pour savoir comment la plupart des structures de données et des algorithmes fonctionnent.

1.2 geemap

`geemap` est un package Python pour la cartographie interactive avec GEE.

`geemap` permet aux utilisateurs d'analyser et de visualiser les ensembles de données GEE de manière interactive, y compris dans un environnement Jupyter.

La fonctionnalité clé de `geemap` est organisée en plusieurs modules: - `geemap`: the main module for interactive mapping with Google Earth Engine, `ipyleaflet`, and `ipywidgets`. - `eefolium`: a module for interactive mapping with Earth Engine and `folium`. It is designed for users to run `geemap` with Google Colab. (Note: Google colab doesnt currently integrate with `ipyleaflet`) - `conversion`: utilities for automatically converting Earth Engine JavaScripts to Python scripts and Jupyter notebooks. - `basemaps`: a module for adding various XYZ and WMS tiled basemaps. - `legends`: a module for adding customized legends to interactive maps.

1.2.1 Getting started

`geemap` n'est pas inclus dans la bibliothèque standard de Python, ce qui signifie que vous devez le télécharger et l'installer.

Vous n'avez besoin d'installer ce package dans votre environnement qu'une seule fois. Après cela, pour chaque session (dans chaque cahier), il vous suffit de: `import geemap`

Nous allons utiliser les méthodes try et except de Python pour installer un package s'il n'est pas déjà importé. De cette façon, nous incluons ce code dans chaque notebook et le package sera importé s'il est déjà installé et sinon, il l'installera en premier.

NOTE 1 Nous utilisons pip pour installer tous les packages en Python.

```
[ ]: try:
    import geemap
except ModuleNotFoundError:
    if 'google.colab' in str(get_ipython()):
        print("package not found, installing w/ pip in Google Colab...")
        !pip install geemap
    else:
        print("package not found, installing w/ conda...")
        !conda install mamba -c conda-forge -y
        !mamba install geemap -c conda-forge -y
import geemap
```

NOTE 2 si vous installez ces bibliothèques dans une session Google Colab, vous serez invité à redémarrer votre Kernel afin d'importer les bibliothèques.

Avec le geemap importé, votre session Python ou votre notebook Jupyter est maintenant prêt à faire appel à toute la bibliothèque d'images / collections d'images GEE, etc., et aura les mêmes fonctionnalités que l'éditeur de code de la console GEE.

Rappelez-vous que pour utiliser GEE, vous avez besoin d'un compte. La première fois que vous exécutez cette cellule, vous serez invité à entrer votre code d'authentification. Une autre fenêtre de navigateur s'ouvrira dans laquelle vous pourrez sélectionner le compte Google que vous avez associé à votre compte GEE. Copiez le code d'authentification fourni et entrez-le dans la zone de saisie fournie dans ce cahier et appuyez sur Entrée et vous devriez être prêt à partir. Vous ne devriez pas avoir à recommencer sur cet ordinateur.

```
[ ]: try:
    ee.Initialize()
except Exception as e:
    ee.Authenticate()
    ee.Initialize()
```