

# Guide d'installation Copernicus dans mBlock

Voici les étapes OBLIGATOIRES pour pouvoir utiliser l'extension Copernicus dans mBlock :

- [Mise en place de l'environnement dans mBlock](#)
- [Mise en place du serveur Flask](#)

## Mise en place de l'environnement dans mBlock :

- Installer mBlock PC Version (Installer [ici](#)) [Version testée 5.4.0]
- Cliquer sur Objets
- Lorsque l'installation est terminée et prête à utiliser les blocs, glisser déposer le fichier .mext fourni dans l'application ([Téléchargeable ici](#))  
Une nouvelle catégorie nommée "Copernicus" devrait apparaître à la fin de la liste contenant des blocs spéciaux Copernicus.

## Mise en place du serveur Flask :

### Linux :

Prérequis : Python (Installé nativement sur Ubuntu) [Version testée 3.10.4]

- Ouvrir un terminal
- Créer l'environnement de travail : `mkdir -p Copernicus/serveur`
- Se placer dans le dossier du server : `cd Copernicus/serveur`
- Placer dans le dossier "Serveur" le fichier app.py fourni ([Téléchargeable ici](#))
- Créer un environnement virtuel : `python3 -m venv venv`  
Sous WSL natif, il sera nécessaire d'effectuer la commande au préalable :  
`apt install python3-venv`
- Taper "ls"  
Le dossier venv doit apparaître dans la liste
- Activer l'environnement virtuel : `source venv/bin/activate`  
Vérifier que "(venv)" s'affiche en tête d'invite  
(Pour sortir de l'environnement virtuel écrire simplement "deactivate" dans l'invite de commande)

Vous voici dans un environnement virtuel Python vierge.

Nous allons maintenant installer les dépendances au projet grâce à pip :

- Exécuter ces commandes et vérifier que chacune des commandes se termine par un "Successfully installed..." :  
 pip install Flask [Version testée 2.1.2]  
 pip install cdsapi [Version testée 0.5.1]  
 pip install xarray [Version testée 2022.3.0]  
 pip install rasterio==1.3a3 [Version forcée 1.3a3]  
 pip install rioaxarray [Version testée 0.11.1]  
 pip install netcdf4 [Version testée 1.5.8]
- Vous pouvez maintenant lancer le serveur : python app.py

## Windows :

Prérequis : Installer Python (Installer [ici](#)) [Version testée 3.10.4]

- Créer un dossier dans "Documents" nommé Copernicus via l'explorateur Windows
- Dans ce même dossier, créer un sous dossier nommé "serveur"
- Placer dans ce dossier le fichier app.py fourni ([Téléchargeable ici](#))
- Ouvrir une invite de commande (Touche Windows+R -> cmd.exe)
- Se placer dans ce dossier préalablement créé via  
 cd Documents\Copernicus\serveur
- Créer un environnement virtuel : python3 -m venv venv  
 (alternative si cela ne fonctionne pas : python -m venv venv)
- Taper "dir"  
 Le dossier venv doit apparaître dans la liste
- Activer l'environnement virtuel : .\venv\Scripts\activate  
 Vérifier que "(venv)" s'affiche en tête d'invite  
 (Pour sortir de l'environnement virtuel écrire simplement "deactivate" dans l'invite de commande)

Vous voici dans un environnement virtuel Python vierge.

Nous allons maintenant installer les dépendances au projet grâce à pip :

- Exécuter ces commandes et vérifier que chacune des commandes se termine par un "Successfully installed..." :  
 pip install Flask [Version testée 2.1.2]  
 pip install cdsapi [Version testée 0.5.1]  
 pip install xarray [Version testée 2022.3.0]  
 pip install rasterio==1.3a3 [Version forcée 1.3a3]  
 pip install rioaxarray [Version testée 0.11.1]  
 pip install netcdf4 [Version testée 1.5.8]
- Vous pouvez maintenant lancer le serveur : python app.py  
 Pour information, un serveur Flask sur 127.0.0.1 sur le port 5000 est lancé.

Votre environnement local d'utilisation des briques mBlock est maintenant prêt.

### Test du serveur Flask :

Pour vérifier le bon fonctionnement du serveur, cliquez [ici](#), l'heure courante doit apparaître dans un nouvel onglet de votre navigateur. Après vérification, vous pouvez refermer votre onglet.

Pour tester les appels, cliquez [ici](#), lorsqu'un message "Téléchargement terminé" s'affiche, un fichier "download.nc" devrait apparaître dans le dossier du serveur.

Vous pouvez le tester via un bloc Copernicus, cf. [<Guide de l'utilisateur>](#)