

# Manuel d'installation de l'application Brique Copernicus

Application Brique Copernicus



# Sommaire

<b>Pré-requis</b>	<b>2</b>
<b>Télécharger le projet</b>	<b>3</b>
<b>Lancer l'application</b>	<b>4</b>
<b>Configuration</b>	<b>5</b>
Identifiant pour requête Copernicus	5
Changer le port du serveur local	6

# Pré-requis

Tout d'abord pour faire fonctionner le projet il est nécessaire que l'ordinateur dispose de :

- Python ( minimum version 3.7.6 ) ; <https://www.python.org/downloads/>. Attention, il ne faut pas oublier de cocher la case "Add Python to PATH" dans la fenêtre de l'application de téléchargement de Python.
- NodeJs ( minimum version 12.13.0 ) ; <https://nodejs.org/en/download/>

Le projet fonctionne sur Windows 10 Famille (version 1909), Windows 7 professionnel (version 64 bits) et Ubuntu 16.04 LTS. De plus, les navigateurs Mozilla Firefox (version 78.0.2) et Microsoft Edge (version 44.18362.449.0 et 83.0.478.64) et Google Chrome (version 84.0.4147.89) sont compatibles avec l'application.

# Télécharger le projet

Pour récupérer les fichiers sources (ainsi que le manuel d'utilisation), vous pouvez télécharger une archive .zip depuis le dossier Drive du projet.

Le téléchargement des fichiers sources peut aussi se faire via le dépôt Github (<https://github.com/Projet-Brique-Copernicus-URISA/Brique>) en clonant le projet ou en téléchargeant une archive .zip.

Après le téléchargement des sources, il est essentiel, avant d'utiliser l'application, de télécharger les modules et extensions nécessaires. Pour cela, exécutez (en cliquant deux fois) sur le fichier "pre-install" (sur le fichier de commande pour Windows et le script .sh pour Linux). Vous pouvez retrouver les messages renvoyés par le script dans le fichier "log\_preinstall.txt" dans le dossier "log". Si le script lève des erreurs, essayez d'exécuter les lignes du fichier une par une dans un terminal (l'invite de commande windows ou le terminal linux).

Un problème fréquent lors de l'exécution de "pre-install.sh" est l'installation du module basemap (ce qui correspond à la ligne 43 du script). Si cette installation échoue, il est possible de télécharger ce module grâce à quelques manipulations. Attention, il est important de noter que les conseils suivants ne prennent en compte que Windows. Tout d'abord il faut récupérer le fichier source de téléchargement de basemap. Vous devez télécharger le fichier qui sera correct pour vous à l'adresse suivante <https://www.lfd.uci.edu/~gohlke/pythonlibs/#basemap>. Il est important de noter que, pour savoir quel fichier fonctionnera dans votre cas, il faut suivre deux règles par rapport au nom de ce fichier dans la liste que vous voyez sur la page :

- le numéro après le "cp" du nom du fichier correspond à la version de Python que vous utilisez. Par exemple, pour la version 3.7 de Python, le titre contiendra "cp37".
- le dernier numéro du titre correspond à la version de windows que vous utilisez, 32 bits ou 64 bits.

Dans notre cas, c'est le fichier "*basemap-1.2.1-cp37-cp37m-win\_amd64.whl*" qui correspond à notre configuration. Ensuite, il faut ouvrir le dossier des fichiers sources de Python, puis y copier le fichier source de téléchargement de basemap. Enfin il faut ouvrir un terminal (ou l'invite de commande) et exécuter la commande "pip install *nomDuFichierDeBasemap*".

# Lancer l'application

Pour lancer l'application, il faut suivre les étapes suivantes :

- 1) Ouvrir le dossier du projet
- 2) Cliquer sur le fichier exécutable "*launchserver*". Une nouvelle fenêtre va s'ouvrir et afficher le message "Le serveur fonctionne sur le port 8082". Si ce n'est pas le cas, réessayez de cliquer sur le fichier exécutable. Attention, il est possible que le pare-feu windows tente de bloquer l'ouverture du fichier, il est donc nécessaire d'accepter pour continuer à lancer l'application.
- 3) Ouvrir le navigateur et entrer l'adresse <http://localhost:8082/> dans la barre d'adresse

Il est important de noter que l'application utilise un serveur local qui, tel qu'il est fait, ne fonctionnera que sur l'ordinateur sur lequel il est lancé. Donc chaque ordinateur doit lancer un serveur en local pour utiliser l'application. Déployer le serveur de notre application sur un ordinateur distant ou sur un réseau local avec plusieurs clients sur le même serveur ne sont pas des solutions qui ont été testées.

# Configuration

## Identifiant pour requête Copernicus

Les requêtes de téléchargement envers Copernicus ont besoin de clés de l'API, obtenables à partir d'un compte Copernicus du service correspondant (dans notre cas CAMS et C3S), pour fonctionner correctement.

Dans le cas où vous souhaitez utiliser vos propres comptes Copernicus, il est possible de configurer l'application pour pouvoir le faire. Si vous souhaitez tout de même utiliser vos comptes mais que vous n'en avez pas, vous pouvez vous créer des comptes grâce aux liens suivants:

- <https://ads.atmosphere.copernicus.eu/user/register?destination=/cdsapp> pour Atmosphere
- <https://cds.climate.copernicus.eu/user/register?destination=%2F%23!%2Fhome> pour Climate

Ensuite pour récupérer les clés nécessaires aux requêtes, il faut aller sur la page :

- <https://ads.atmosphere.copernicus.eu/api-how-to> pour Atmosphere
- <https://cds.climate.copernicus.eu/api-how-to> pour Climate

Sur chacune des pages on peut retrouver les clés correspondantes aux paragraphes "Install the CDS API key". Les clés ont la forme suivante :

```
url: https://cds.climate.copernicus.eu/api/v2
key: 49162:5ee6baf1-bb6c-4b49-bceb-829ec042ff47
```

Remarque :  
seul la partie **key**: nous intéresse ici.

Une fois que vous avez trouvé les clés de l'API, vous pouvez les ajouter dans le fichier "assets/js/copernicus\_key.js", dans la partie qui correspond (ligne 22 pour atmosphere et ligne 26 pour climate).

```
1 //Key for the atmosphere API (CAMS)
2 //clé de l'api Copernicus Atmosphere
3 api_key_atmosphere = "1825:3bfdb1b4-8a37-4009-b6e5-e5c69d96ea68";
4
5 //Key for the climate change API (C3S)
6 //clé de l'api Copernicus Climat
7 api_key_climate = "49162:5ee6baf1-bb6c-4b49-bceb-829ec042ff47";
```

Exemple du fichier copernicus\_key.js

Pour que la configuration soit effective, il suffit de recharger la page de l'application.

## Changer le port du serveur local

Par défaut, nous avons choisi de lancer le serveur local sur le port 8082. Pour le modifier il faut changer la valeur de la variable *port* du fichier *launchServer.js*, dans le dossier *./Blockly/*. Ensuite il faut reformer l'exécutable si vous souhaitez l'utiliser, en exécutant la commande `"nexe -i launchServer.js"` pour reformer l'exécutable du script.