



Connect - Download - Unzip

1 - Import des bibliothèques

```
Entrée [ ]: from urllib.parse import urlencode
from urllib.request import urlopen
import json
import requests
import time
from pyunpack import Archive
```

2 - Récupération des données de connections à l'api BIHR

```
Entrée [ ]: authen_settings = json.load(open("private/authen_prod.json", "r"))

data = {
    'UserName': authen_settings["CustomerId"],
    'PassWord': authen_settings["securityKey"]
}

post_data = urlencode(data).encode('utf8')
```

3 - Sélection du fichier à télécharger

```
Entrée [ ]: choice ={'1':'fullref',
                    '2':'fullprices',
                    '3':'fullimg',
                    '4':'fullattr'
                }
for x, y in choice.items():
    print(x, "-", y)

yourChoice = input("Saisissez votre choix : ")
catalog = choice[yourChoice]
print(f'Téléchargement du fichier {catalog}')
```

4 - Génération du token d'authentification

```
Entrée [ ]: with urlopen("https://api.bihr.net/api/v2/Authentication/token", post_data) as response:
    resp = json.load(response)

    access_token = resp['access_token']

    access_token
```

5 - Demande du Ticket_ID avec le token

```
Entrée [ ]: headers = {'Authorization' : f'bearer {access_token}',}

Ticket = requests.get(f'https://api.bihr.net/api/v2/Catalog/{catalog}', headers=headers)
resp = Ticket.json()

Ticket_ID = resp['TicketId']

Ticket_ID
```

6 - Demande de génération du fichier à télécharger

6a - Demande initiale

```
Entrée [ ]: Download = requests.get(f'https://api.bihr.net/api/v2/Catalog/status?ticketId={Ticket_ID}')
resp = Download.json()

Download_ID = resp['DownloadId']

status = resp['RequestStatus']
PositionInQueue = resp['PositionInQueue']

print(status)
print(f'PositionInQueue = {PositionInQueue}')
```

6b - Demandes suivantes jusqu'à génération du fichier

```
Entrée [ ]: while status != "DONE": # On gère l'attente pour préparation du DownloadId
    time.sleep(10)
    response = requests.get(f'https://api.bihr.net/api/v2/Catalog/status?ticketId={Ticket_ID}')
    resp = response.json()
    status = resp['RequestStatus']
    print(resp['RequestStatus'])
    print(resp['PositionInQueue'])
    if status == "ERROR":
        print("Traitement en erreur !")
        break

Download_ID = resp['DownloadId']
print(f'Download_Id = {Download_ID}')
```

7 - Téléchargement du fichier généré précédemment

7a - Construction et lancement de la requête.

```
Entrée [ ]: url = f'https://api.bihr.net/api/v2/Catalog/download?downloadId={Download_ID}'

r = requests.get(url, headers=headers)
```

7b - Affichage des "headers" de la réponse

```
Entrée [ ]: r.headers
```

7c - Renommage du fichier

```
Entrée [ ]: dfile = r.headers['Content-Disposition']
dfile = dfile.split("; ")
dfile = dfile[1]
dfile = dfile.replace("filename=", "")
dfile = dfile.replace("\\", "")

dfile
```

7d - Ecriture du fichier dans le dossier dédié

```
Entrée [ ]: with open(f"download_Files/{dfile}", "wb") as file:
            file.write(r.content)
```

8 - Décompression du fichier .7z téléchargé

```
Entrée [ ]: # On dézippe Le fichier reçu et on le range dans dossier unzipped
new = Archive(f"download_Files/{dfile}").extractall('unzipped_files/.')
```

```
Entrée [ ]:
```

```
Entrée [ ]: # Avec Requests
url = "https://api.bihr.net/api/v2/Authentication/token"
resp = requests.post(url, post_data).json()

resp
```

```
Entrée [ ]: # Avec urllib.request
with urlopen("https://api.bihr.net/api/v2/Authentication/token", post_data) as response:
    resp = json.load(response)

access_token = resp['access_token']

access_token
```

```
Entrée [ ]:
```