

**Fonctionnement du script d'insertion**



## Fonctionnement du script d'insertion

Voici un document sur le fonctionnement du script d'insertion dans la base de données.

Nous avons choisi le langage python afin d'insérer nos données dans notre base de données PostgreSQL.

Vous pouvez trouver l'ensemble des fichiers juste [ici](#).

Commençons avec le script Bash il va d'abord vérifier les packages installés, si les packages ne sont pas installés, il les installera automatiquement. Il vérifie également les modules python nécessaires au fonctionnement du script python, s'ils ne sont pas installés celui-ci les installe automatiquement.

Après l'installation des dépendances, le script va créer un second script qui se nomme "autoImport\_IMDB\_ro\_POSTGRES.sh" celui-ci va exécuter main.py.

"installation\_autoImport.sh" va ensuite configurer "cron" qui exécutera le script "autoImport\_IMDB\_ro\_POSTGRES.sh" quotidiennement à 4h30 chaque jour. Celui-ci créait également des logs accessibles avec la commande suivante :

```
cat /etc/DraCorporation/logs_autoImport_IMDB_to_POSTGRES.txt
```

Allons voir plus en détail notre script python, il est composé de 11 fichiers python et il utilise divers modules.

| Module    | Utilité   |
|-----------|---|
| psycopg2  | Permet d'établir une connexion entre python et postgres, il permet également d'utiliser diverse commande SQL. |
| traceback | Permet de récupérer d'éventuelles erreurs qui ont eu lieu lors de l'exécution du script                       |
| requests  | Permet de télécharger des fichiers via des urls   |
| gzip      | Permet de décompresser des fichiers   |
| shutil    | Permet de décompresser les fichiers avec une extension en .gz   |
| os        | Permet d'utiliser diverses commandes pour créer ou supprimer des répertoires par exemple                      |
| time      | Permet de calculer le temps d'exécution du script   |

### main.py

main.py fait appel à plusieurs autres scripts que l'on détaille plus bas, dans un premier temps main.py nous connecte à notre base de données grâce à la fonction *createLink()* présent dans **getLink.py**, il va ensuite télécharger les données d'IMDB avec la fonction *download()* qui se situe dans le script **downloadFiles.py**, main.py lancera ensuite l'injection des données sur PostgreSQL avec le fichier **injectData.py**.

#### getLink.py

getlink.py crée une fonction *createLink()* qui ouvre le fichier **credentials.txt** qui contient toutes les informations nécessaires pour établir la connexion à PostgreSQL. Il récupère les données et essaye de se connecter à la base si celui-ci réussit, il nous renvoie un message qui nous confirme la connexion, si la connexion échoue celui-ci nous le signale et nous montre l'erreur grâce au module traceback.

#### downloadFiles.py

downloadFiles.py crée 4 fonctions, *getUrl()* qui récupère les URL d'IMDB pour télécharger les fichiers, *downloadFiles()* télécharge les fichiers et crée un répertoire pour ranger les fichiers, *unzipFiles()* permet de compresser nos fichiers et de les transformer en .csv et enfin *download()* qui fait appel aux 3 fonctions précédemment citées.

#### injectData.py

injectData.py crée 3 fonctions le premier *createTables()* permet de créer les tables pour cela, il accède à notre répertoire SQL qui contient divers scripts SQL pour créer les tables, ensuite il y a la fonction *injectFiles()* qui injecte les données dans les tables précédemment créées. Enfin *injectData()* fait appel aux deux fonctions précédemment créées.

[namebasicsInsert.py](#) | [titleakasInsert.py](#) | [titlebasicsInsert.py](#) | [titlecrewInsert.py](#) |  
[titleepisodelInsert.py](#) | [titleprincipalsInsert.py](#) | [titleratingInsert.py](#)

Les scripts se terminant par Insert.py fonctionnent comme ceci, ils ouvrent le fichier csv et le lis ligne par ligne, ensuite, ils regroupent les informations à travers une requête SQL et ils importent toutes les données du fichier csv 1 par 1. De plus, dans certains fichiers csv il manquait les crochets dans certaines colonnes pour y remédier les scripts les ajoute automatiquement.