1. **Co to jest? Do czego służy?**

Skrypt napisany w Pythonie, służy do pobrania i procesowania danych rejestrowych z *API Polskiej Wywiadowni Gospodarczej (PWG)*. Dane są zapisane w formacie JSON, parsowane oraz eksportowane do pliku w formacie xlsx. Dane są następnie wykorzystywane do celów śledczej analizy danych, w szczególności realizacji projektów FAIT. Skrypt wykorzystuje metodę getFull, która została opisana w dokumentacji udostępnionej przez PWG:

<https://api.pwginfo.pl/doc/Api_Pwc.html>

1. **Jak działa?**

Skrypt pobiera JSONy zawierające informacje nt. danych rejestrowanych dla wprowadzonych przez użytkownika identyfikatorów podmiotów (mogą to być NIP, KRS, REGON, lub PWG\_ID). Wykorzystywana metoda to getFull.

Działanie użytkownika polega na:

1. Podaniu danych wymaganych do uruchomienia skryptu:
   1. Zapisania jako zmienną *INPUT\_FILE* ścieżki do pliku (łącznie z nazwą i fomate) w formacie .xlsx, w którym znajduje się lista NIPów sprawdzanych podmiotów oraz podania w pliku NIPów pierwszej kolumnie w pierwszej zakładce,
   2. Podania jako zmienną *JSONS\_OUTPUT\_PATH* ścieżki do katalogu, do którego mają zostać zwrócone pobrane JSONy.
   3. Podania jako zmienną *OUTPUT\_PATH* ścieżki do katalogu, do którego mają zostać wyeksportowane sparsowane dane rejestrowe w formacie .xslx.
2. Uruchomieniu skryptu:

W skrypcie-bibliotece **pwg** znajdują się obiekty oraz funkcje wykorzystywane do parsowania i eksportowania danych. Skrypt został udokumentowany w pliku *pwg\_doc*. Proces pobierania danych opiera się na skrypcie *get\_data.py*, który został udokumentowany w pliku *get\_data\_doc*.

Skrypt działa w następujący sposób:

1. Przy wykorzystaniu funkcji **get\_data** dla wprowadzonej przez użytkownika listy identyfikatorów podmiotów pobierane są dane rejestrowe w formacie JSON. Dane są eksportowane do wskazanego katalogu (zmienna *JSONS\_OUTPUT\_PATH*). JSONy dla udanych i nieudanych pobrań są zapisywane odpowiednio w słownikach *company\_ids* oraz *company\_ids\_errors*.
2. Dane są wykorzystywane jako input do funkcji **parse\_get\_full**. W pętli, dla pojedynczych NIPów i JSONów parsowane są dane przechowywane w JSONach. Funkcja odczytuje pobrane dane i parsuje z nich wyselekcjonowane dane rejestrowe. Zbędne dane są usuwane. Prawidłowo pobrane i rozparsowane dane rejestrowe są oczyszczane ze zbędnych kolumn, następuje zmiana nazw kolumn na bardziej czytelne. Zagnieżdżone pola są przekształcane w pojedyncze kolumny.

Rekordy, których nie udało się rozparsować są odkładane i zapisywane z odpowiednim opisem błędu. Prawidłowo sparsowane rekordy są łączone w jedną tabelę i następnie zapisywane w formacie DataFrame. Dane są następnie łączone w jedną tabelę. Nieprawidłowo pobrane/sparsowane dane także są łączone w tabelę DataFrame, która jest łączona z poprawnie pobranymi danymi, przez co tworzony jest ujednolicony plik outputowy.

Jako osobne zmienne zwracane są:

*nip\_finance\_year* – słownik przechowujący informacje na temat identyfikatorów podmiotów (key) oraz najnowszego roku finansowego, dla którego w bazie PWG dostępne są dane (value), jeśli takowe występują.

*shareholder\_press* – słownik zawierający identyfikatory (key) podmiotów oraz sparsowane listy (value) NIPów udziałowców, którzy są przedsiębiorstwami (jeśli takowi występują).

1. Za pomocą funkcji **export\_output** sparsowane dane są eksportowane w formacie .xslx do wskazanego folderu outputowego. Tworzone są zakładki zawierające odpowiednio:

* FullOutput – całkowity output, który udało się poprawnie pobrać i sparsować oraz listę nipów (i EMIS\_ID jeśli są dla niego dostępne informacje w EMIS),
* ValidOutput – tylko pobranie pobrane i sparsowane rekordy
* InvalidOutput - tylko niepoprawnie pobrane i sparsowane rekordy, w celu łatwiejszej identyfikacji braku / niepoprawnych danych.

Zakładki ValidOutput oraz InvalidOutput tworzone są jedynie w sytuacji, gdy zawierają odpowiednie dane. Jeśli taka sytuacja nie występuje nie zostają utworzone.

1. **Uwagi. Możliwości poprawy/rozwoju.**

- Pobieranie listy nipów z bazy SQL (po dostosowaniu się do migracji GUS)

- Parsowanie zagnieżdżonych pól przy wykorzystaniu bardziej ogólnych metod, nie pojedynczych funkcji