1. **Co to jest? Do czego służy?**

Skrypt napisany w Pythonie, służy do pobrania i procesowania danych finansowych z API Polskiej Wywiadowni Gospodarczej (PWG). Dane są pobierane dla wskazanych przez użytkownika listy lat finansowych. 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 getFinance, która została opisana w dokumentacji udostępnionej przez PWG:

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

1. **Jak działa?**

Skrypt pobiera JSONy zawierające informacje nt. danych finansowych dla wprowadzonych przez użytkownika identyfikatorów podmiotów (mogą to być NIP, KRS, REGON, lub PWG\_ID). Dane są pobierane dla wskazanych przez użytkownika lat finansowych. Wykorzystywana metoda to getFinance.

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 *XLSX\_OUTPUT\_PATH* ścieżki do katalogu, do którego mają zostać wyeksportowane sparsowane dane rejestrowe w formacie .xslx.
   4. Zapisania jako zmienną *fiscal\_years* (lista) listy lat finansowych, dla których mają zostać pobrane dane. Lata finansowe (każdy element) powinny zostać wprowadzone przez użytkownika jako string.

1. Uruchomieniu skryptu:

W skrypcie-bibliotece **pwg** znajdują się obiekty oraz funkcje wykorzystywane do parsowania i eksportowania danych. Skrypt został udokumentowany w pliku *pwg\_doc*.

Skrypt działa w następujący sposób:

1. Definiowana jest funkcja **get\_jsons**, która pobiera dane finansowe dla wprowadzonej listy identyfikatorów podmiotów i lat finansowych. Wykorzystywana jest zagnieżdżone pętla, w które dane są pobierane dla kolejnych podmiotów i lat finanowych. 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\_finance**. W pętli, dla pojedynczych NIPów i JSONów parsowane są dane przechowywane w JSONach. Funkcja odczytuje pobrane dane i parsuje z nich wyselekcjonowane finansowe. Na ten moment wykorzystywane są jedynie 9 pól finansowych niezbędnych do realizacji projektów FAIT. Zbędne dane są usuwane. Prawidłowo pobrane i rozparsowane dane są oczyszczane ze zbędnych kolumn, następuje zmiana nazw kolumn na bardziej czytelne. Dostawiane są kolumny wskazujące na rodzaj sprawozdania finansowego raz rok finansowy dla którego nastąpiło pobranie danych.

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. Funkcja zwraca DataFrame – całkowity output, który udało się poprawnie pobrać i sparsować. Tabela zawiera dane finansowe oraz podstawowe informacje na temat podmiotu, takie jak: NIP, EMIS\_ID oraz rodzaj sprawozdania finansowego,

1. Za pomocą funkcji **export\_output** sparsowane dane są eksportowane w formacie .xslx do wskazanego folderu outputowego. Tworzone są zakładki zawierające dane pogrupowane według lat finansowych, za które udało się pobrać dane finansowe, ich nazwy wskazują na dany rok finansowy. Zakładki tworzone są jedynie w sytuacji, gdy zawierają odpowiednie dane. Jeśli za dany rok finansowy nie zostały pobrane żadne dane, nie zakładka nie zostaje utworzona.
2. **Uwagi. Możliwości poprawy/rozwoju.**

- Parsowanie wszystkich pozostałych pól z danymi finansowymi,

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