1. **Co to jest? Do czego służy?**  
     
   Skrypt napisany w Pythonie, służy do pobrania **danych finansowych** z APIwywiadowni gospodarczej *EMIS* dla zadanej listy firm i wybranych lat finansowych. Po pobraniu dane są zapisywane w formacie JSON a następnie parsowane do postaci wymaganej do wykorzystania danych do narzędzia FAIT. Dane finansowe są eksportowane do pliku xlsx.
2. **Jak działa?**  
     
   Skrypt pobiera JSONy dla wprowadzonej przez użytkownika listy identyfikatorów podmiotów, oraz lat finansowych. Dozwolone identyfikatory to: NIP, REGON, KRS. Jako default skrypt wykorzystuje NIP. Pobrane dane są przetwarzane, zapisywane jako dataframe i następnie eksportowane do formatu .xlsx w zagregowanej formie i dla każdego roku finansowego w oddzielnej zakładce. Następnie dane są wykorzystywane przez narzędzie FAIT do testów finansowych.

Wybór użytkownika polega na:

1. Zapisania jako zmienną ID\_CLASS typu identyfikatorów wprowadzonych podmiotów, dla których mają zostać pobrane dane finansowe. Dozwolone wartości to: 'NIP', 'REGON', 'KRS'.
2. Zapisania jako zmienną *INPUT\_FILE* ścieżki do pliku w formacie .xlsx (łącznie z nazwą pliku), w którym znajduje się lista NIPów podmiotów, dla których pobrane mają zostać dane finansowe,
3. Zapisania jako zmienną *JSON\_OUTPUT\_PATH* ścieżki do katalogu, do którego zwrócone mają zostać pliki JSON z danymi finansowymi,
4. Zapisania jako zmienną *XLSX\_OUTPUT\_PATH* ścieżki do pliku (łącznie z nazwą pliku), do którego zwrócone mają zostać sparsowane pliki w formacie .xlsx.
5. Wprowadzeniu lat finansowych, dla których mają zostać pobrane dane finansowe (*fiscal\_years*). Krok konieczny w przypadku korzystania z **getStatementByCode**. Dane powinny zostać wprowadzone w formacie listy.

W pierwszej kolejności skrypt pobiera klucz sesji (za pomocą funkcji **get\_session\_id)**, który składa request o podanie unikalnego klucza sesji. Skrypt jest pobierany i następnie wykorzystywany do autentykacji użytkownika podczas pobierania danych z API.

W dalszej kolejności skrypt odczytuje wprowadzoną przez użytkownika listę NIP sprawdzanych podmiotów (za pomocą funkcji **get\_nip\_name**) i zapisuje je jako listę. Lista wprowadzonych NIP jest walidowana oraz deduplikowana. Dodatkowo tworzony jest log, przechowujący informacje na temat poprawnych i niepoprawnych NIP.

Następnie skrypt za pomocą funkcji **get\_isic\_list** pobiera dla każdego z poprawnych NIP zwróconych przez funkcję get\_nip\_nape do listy *nip\_list* kod ISIC. Kod ISIC (EMIS\_ID) to unikalny identyfikator podmiotu w bazie danych EMIS. W przypadku niepowodzenia (firmy nie ma w bazie EMIS) NIP podmiotu, dla którego nie udało się pobrać kodu ISIC jest zapisywany w oddzielnym słowniku wraz z odpowiednią adnotacją, po czym funkcja kontynuuje pobieranie kodów. Funkcja zwraca 2 słowniki – *nips\_isics* oraz *nips\_errors* – przechowujące odpowiednio informacje na temat udanych i nieudanych pobrań kodu ISIC.

Obydwa słowniki przekazywane są jako input do funkcji **get\_financial\_data**. Dla podmiotów dostępnych w EMIS w podwójnej pętli dla kodów ISIC oraz dla wprowadzonych przez użytkownika lat finansowych (zmienna *fiscal\_years)* wysyłane są zapytania o dane finansowe, które są następnie odczytywane, rozkodowywane i eksportowane do wskazanego katalogu *JSON\_OUTPUT\_PATH* z odpowiednim guidem wskazującym na to, czy pobranie się powiodło, czy nie. W przypadku udanego pobrania danych JSON jest przekazywany jako wartość do słownika *nips­\_jsons* z kluczem w postaci numeru NIP podmiotu. W przypadku nieudanego pobrania danych (np. spowodowanego brakiem danych za wybrany rok finansowy, lub błędy połączenia z API) zbierany jest kod błędu i przekazywany do słownika *nips\_errors* wraz z odpowiednim kodem błędu jako wartość wskazującym przyczynę nieudanego pobrania.

Eksportowane pliki zawierają odpowiedni NIP firmy. W przypadku powodzenia pobrania dane są zapisywane we wskazanym folderze outputowym z odpowiednim guidem (nazwą pliku), który zawiera NIP podmiotu, oraz guid *EMIS\_FINANCE* + rok finansowy. Nie każde zapytanie musi zakończyć się sukcesem, dlatego w skrypcie umieszczona została klauzula try/except. W przypadku niepowodzenia skrypt zapisuje dany kod ISIC i zwracany kod błędu. W przypadku niepowodzenia plik zostaje zapisany i wyeksportowany do folderu outputowego z odpowiednim guidem (dodana fraza *INVALID)*, który wskazuje na to, że pobranie danych jest nieudane.

Po zakończeniu pobierania danych zostają one przekazane jako input do funkcji **parse\_financial\_data**. Funkcja odczytuje prawidłowo pobrane dane, i odczytuje z nich **tylko dane finansowe wymagane do testów przeprowadzanych w FAIT**. Jedynie część kolumn jest wykorzystywana. Odczytywane są także rok finansowy, data ostatniej aktualizacji danych oraz typ sprawozdania finansowego (consolidated / individual). Kody ISIC mapowane są na pierwotnie wprowadzone NIPy.

W dalszej kolejności wykorzystywana jest funkcja **create\_financial\_output**, która jako input wykorzystuje sparsowane dane i przekształca je do postaci tabelarycznej (pd.DataFrame). Do tabeli przekazywane są wszystkie sparsowane informacje, które są sortowane po identyfikatorze podmiotu i roku finansowym.

Prawidłowo pobrane dane finansowe są eksportowane w formacie .xslx do wskazanego folderu outputowego za pomocą funkcji **export\_output**. Wszystkie pobrane dane są agregowane i zapisywane w zakładce „Total”. Oprócz tego, dane dla poszczególnych lat finansowych, dla których udało się pozyskać dane zapisywane są do oddzielnych zakładek z odpowiednią nazwą wskazującą na rok finansowy.

Z pobrania tworzony jest log, który w konsoli wyświetla informacje na temat pobierania danych.

**Uwagi. Możliwości poprawy/rozwoju.**

- rozbudowanie skryptu o dodatkową funkcjonalność, która automatycznie pobiera najnowszy rok finansowy, odczytuje go i dla niego pobiera dane finansowe i eksportuje w obecnej formie,

- rozbudowanie skryptu o parsowanie pozostałej części pobieranych danych finansowych,

- dodanie funkcjonalności eksportowania przetworzonych danych do bazy danych,