1. **Co to jest? Do czego służy?**  
     
   Skrypt napisany w Pythonie, służy do pobrania danych z APIwywiadowni gospodarczej *EMIS*. Po pobraniu dane są zapisywane w formacie JSON. W zależności od użytej metody pobierania danych mogą przechowywać różne informacje. Na ten moment wykorzystujemy dwie metody:
2. getCompanyLatestStatement – pobiera informacje o najnowszym roku, dla którego dla firmy w EMIS dostępne jest sprawozdanie finansowe i dane finansowe
3. getStatementByCode – pobiera sprawozdanie finansowe dla zadanej firmy, roku finansowego, waluty i statusu konsolidacji

Pełna dokumentacja API i dostępnych metod jest dostępna tutaj:

<https://www.emis.com/EMIS-API-Documentation-Company-data>

<https://www.emis.com/EMIS-API-Documentation-News-and-reports>

1. **Jak działa?**  
     
   Skrypt pobiera JSONY dla wprowadzonej przez użytkownika listy podmiotów, metody pobierania danych, roku finansowego i pozostałych parametrów.

Wybór użytkownika polega na:

1. Wprowadzeniu nazwy użytkownika (zmienna *user\_name*),
2. Wprowadzeniu hasła użytkownika (*password*),
3. Wprowadzeniu metody (*method*), która ma zostać użyta do pobierania danych (więcej informacji w dokumentacji),
4. Wprowadzeniu roku finansowego, dla którego pobrane mają zostać dane finansowe (*financial\_year*). Krok konieczny w przypadku korzystania z **getStatementByCode**,
5. Zapisania jako zmienną *input\_file* ścieżki do pliku w formacie .xlsx, w którym znajduje się lista NIPów podmiotów, dla których pobrane mają zostać dane podmiotów,
6. Zapisania jako zmienną *JSON\_OUTPUT\_PATH* ścieżki do folderu outputowego, do którego zwrócone mają zostać dane wynikowe.

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\_list**) i zapisuje je jako listę. Definiowana jest funkcja (**request\_single\_isic**), która dla pojedynczego podmiotu pobiera kod ISIC (identyfikator podmiotu w bazie EMIS, który jest konieczny do wykonania zapytania dla firmy). Nie jest możliwe wykonanie zapytań bezpośrednio za pomocą NIP, REGON, itp. Następnie skrypt za pomocą funkcji **get\_isic\_list** pobiera dla każdego z podmiotów wprowadzonych w pliku inputowym kod ISIC. W przypadku niepowodzenia (firmy nie ma w bazie EMIS) NIP podmiotu, dla którego nie udało się pobrać kodu ISIC jest zapisywany w oddzielnej liście wraz z odpowiednią adnotacją i funkcja kontynuuje pobieranie kodów. Funkcja zwraca 2 dataframe – przechowujące odpowiednio informacje na temat udanych i nieudanych pobrań kodu ISIC.

W dalszej kolejności definiowana jest funkcja **request\_data**, która tworzy odpowiedni request dla zadanej przez użytkownika metody, roku finansowego. Funkcja odczytuje wygenerowane w poprzednim kroku klucze ISIC. Utworzone zapytania o dane są dalej wykorzystywane przez funkcję **get\_data**. Funkcja w pętli pobiera dane dla listy podmiotów dostępnych w bazie EMIS i zapisuje je w formacie JSON. Funkcja wykonuje metodę podobną do funkcji vlookup i mapuje numery ISIC, dla który pobierane są dane z numerami i NIP. Eksportowane pliki zawierają odpowiedni NIP firmy. W przypadku powodzenia pobrania dane są zapisywane we wskazanym folderze outputowym z odpowiednim guidem (nazwą pliku), który zawiera kod ISIC podmiotu, nazwę wykorzystywanej metody i rok finansowy (jeśli zapytanie dotyczy danych finansowych). Nie każde zapytanie musi zakończyć się sukcesem (np. z powodu braku danych za dany rok finansowy w bazie EMIS), 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, który wskazuje na to, że pobranie danych jest nieudane.

Funkcja zwraca 2 dataframe’y przechowujące odpowiednio informacje na temat udanych i nieudanych zapytań.

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

**Istnieją rozbudowana wersje skryptu, które oprócz pobierania danych parsują dane finansowe i dane rejestrowe a następnie eksportują je w formie Excel do tabel.**

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

- tworzenie zapytania dla wszystkich zadanych podmiotów jednocześnie w formie listy, nie w formie pętli dla pojedynczych podmiotów jak ma to miejsce obecnie (API udostępnia taką możliwość, ale takie rozwiązanie wiąże się z koniecznością parsowania zwracanych danych i odczytywaniu pojedynczych ISIC. Parsowanie jest wtedy bardziej problematyczne, ale metoda będzie bardziej wydajna),

- rozbudowanie skryptu o pobieranie danych za pomocą innych metod, np. getFullCompany, należy zastanowić się, które metody są godne uwagi (są opisane w dokumentacji),

- parsowanie danych dla innych metod, teraz za pomocą Pythona parsowane są tylko dane finansowe i dane rejestrowe