1. **Co to jest? Do czego służy?**  
     
   Skrypt napisany w Pythonie, służy do pobierania **raportu generalnego i raportu PKD** z APIGUSu dla wprowadzonej listy identyfikatorów firm. Dozwolone identyfikatory to: NIP, REGON, KRS. Jako wartość domyślną skrypt wykorzystuje NIP. Po pobraniu dane są procesowane, łączone w tabelę, przeprowadzane są testy do narzędzia FAIT. Po wszystkim następuje eksport do wskazanego katalogu. Skrypt wykorzystuje gotowe rozwiązanie do połączenia się z API i pobierania raportów dostępne pod adresem: <https://github.com/bogdal/gusregon>

Na potrzeby zespołu rozwiązanie zostało zmodyfikowane w celu otrzymania spójnej struktury dostosowanej do potrzeb projektowych.

1. **Jak działa?**  
     
   Skrypt pobiera dane dla wprowadzonej przez użytkownika listy podmiotów za pomocą ustanowionego połączenia z API. Pobrane dane są przetwarzane, zapisywane jak dataframe, wykonywana jest kalkulacja testów i następnie wyniki eksportowane do formatu xlsx w oczyszczonej postaci. Następnie dane są wykorzystywane przez narzędzie FAIT.

Wybór użytkownika polega na:

1. Zapisania jako zmienną *INPUT\_FILE* ścieżki do pliku w formacie .xlsx (łącznie z nazwą pliku), w którym znajduje się lista identyfikatorów podmiotów, dla których mają zostać pobrane dane. Dozwolone typy identyfikatorów to NIP, REGON, KRS. Domyślnie wykorzystywany jest NIP. W celu zmiany na REGON/KRS konieczna jest ingerencja w kod.
2. Zapisania jako zmienną *XLSX\_OUTPUT\_PATH* ścieżki do pliku (łącznie z nazwą pliku), do którego zwrócone mają zostać sparsowane pliki w formacie .xlsx,
3. Zapisania jako zmienną *CLIENT\_PKD* głównego PKD klienta, dla którego realizowany jest projekt. Jest to konieczne w celu wyliczenia testów na ryzykowne działalności kontrahentów w kontekście działalności prowadzonej przez klienta.

W pliku keys\_coded:

1. Zapisania jako zmienną *PROJECT\_NAME* nazwy projektu, pod którą tabele (opisane później) mają zostać wyeksportowane do bazy danych.

Sposób działania skryptu:

W pierwszej kolejności wprowadzoną przez użytkownika listę NIP sprawdzanych podmiotów i zapisuje je jako listę.

Następnie za pomocą funkcji **get\_data\_gus** pobierane są raporty ogólne i raporty nt PKD weryfikowanych podmiotów. Możliwe jest także pobranie raportu nt adresu podmiotu, ale nie jest to konieczne do potrzeb projektowych. Pobrane raporty są parsowane w celu otrzymania spójnej struktury danych. Zduplikowane dane i pola, gdzie występuje bardzo duży odsetek pustych wartości są pomijane. Następuje też transformacja wartość pól z PKD kontrahentów. Działalności dodatkowe są skupiane w jednej kolumnie w celu uniknięcia rozbudowania tabeli.

Następnie przy wykorzystaniu funkcji **calculate\_gus\_tests** wyliczane są wartości testów wykorzystywanych do projektów FAIT. Są one wyliczane na złączonych raportach ogólnych i PKD, które zawsze mają taką samą strukturę. Tworzona jest też oddzielna tabela z adresami podmiotów, która jest gotowa do wykonania geocodingu w celu uzyskania widoków street view z Google API.

Następnie za pomocą funkcji **create\_output**, tworzone są tabele końcowe.Prawidłowo sparsowane rekordy są łączone w jedną tabelę i następnie zapisywane w formacie DataFrame. Tabele są eksportowane do oddzielnych zakładek pliku xlsx do katalogu wskazanego przez użytkownika. Dodatkowo, dane są eksportowane do bazy danych (domyślnie GHC) jako dwie tabele. Są to adresy do geokodowania i tabela z danymi z GUS oraz wyliczonymi testami. Tabele są eksportowane z odpowiednim guidem wskazanym w zadeklarowanej zmiennej *PROJECT\_NAME*.

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

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

- walidacja NIP, REGON, KRS i dodanie automatycznego sprawdzania typu identyfikatora

- error handling – skrypt jest bardzo prosty, ponieważ osoby, które go wykorzystywały były w stanie obsłużyć je na bieżąco,

- testy eksportu do SQL