Dokumentacja do zadania

Nel Kułakowska

> Co powinno być objęte przez system monitoringu, żebyśmy wiedzieli, że w naszym przepływie coś nie działa poprawnie?

Program działa tak że wypisuje logi w konsoli (chyba że się wyłączy to opcją `-p 0`) oraz zapisuje do wskazanych przez użytkownika plików. Domyślnie zapisuje wszystkie logi do *"all_logs.log"*, a logi, które wskazują na błędy i nieprzewidziane zdarzenia do pliku do *"warning_logs.log"*. System monitoringu powinien obserwować czy w pliku z błędami nie pojawiają się jakieś wpisy.

> W jaki sposób powinno być uruchamiane rozwiązanie na serwerze, gdzie zostanie umieszczone?

Rozwiązanie powinno być uruchomione na serwerze, gdzie, oprócz odpowiedniej wersji Pythona, dostępne będą biblioteki zapisane w *"requirements.txt"*. Rozwiązanie można uruchomić:

```
python3 converse.py json_folder
```

gdzie json_folder - ścieżka do folderu z plikami json.
Podczas uruchamiania można podać więcej parametrów : - `-output` / `--output_folder_path` - ścieżka do folderu wyjściowego, domyślnie *"xml_folder"* - `-tww` / `--time_wait_write` - czas, przez który program sprawdza czy plik nie jest jeszcze zapisywany (czyli czy podczas sprawdzania zawartości pliku, nie zmienił się jego atrybut dotyczący ostatniej modyfikacji) - `-log_a` / `--log_a` - ścieżka do pliku ze wszystkimi logami, domyślnie *"all_logs.log"* - `-log_w` / `--log_w` - ścieżka do pliku z logami na poziomie warning lub wyższym, domyślnie *"warning_logs.log"* - `-np` / `--no_print` - jeśli ustawimy ten parametr (bez wartości) to nic nie będzie wypisywane w konsoli

Opis ten można uzyskać też wpisując w konsoli: `python3 converse.py -h`.

Funkcje programu:

1. Monitorowanie zmian w folderze wejściowym:
   Program monitoruje zmiany w folderze wejściowym, reagując na modyfikacje i tworzenie nowych plików. Wykorzystuje do tego mechanizm wątków oraz bibliotekę watchdog.
2. Konwersja plików JSON na XML:
   Po wykryciu zmiany w pliku JSON, program przekształca jego zawartość do formatu XML, zachowując istotne dane zgodnie z wymaganiami.
3. Obsługa błędów:
   Program wyposażony jest w mechanizmy obsługi błędów, które zapewniają ciągłość działania nawet w przypadku wystąpienia nieprzewidzianych sytuacji, takich jak brakujące klucze w pliku JSON lub problemy z odczytem danych.

4. Logowanie zdarzeń:
   Działania programu są rejestrowane za pomocą logów, co umożliwia monitorowanie przebiegu procesu konwersji oraz identyfikację ewentualnych problemów.
5. Dostosowanie parametrów uruchomieniowych:
   Program umożliwia dostosowanie różnych parametrów, takich jak ścieżki do folderów wejściowego i wyjściowego, czas oczekiwania na zapis pliku oraz aktywowanie lub dezaktywowanie wyświetlania komunikatów.