

Integracja OpenAI o3-pro Batch API, Google Cloud Run i LibreOffice Writer

Niniejszy raport opisuje system do masowego przetwarzania dokumentów prawnych z wykorzystaniem LibreOffice Writer, OpenAI o3-pro Batch API, Google Cloud Run, Firestore, Google Cloud Storage oraz Perplexity.ai.

ŚRODOWISKO

Python 3.11.14

Makra uruchamiane bezpośrednio w LibreOffice Writer.

ŹRÓDŁA PRAWA

2 rejestry

Dziennik Ustaw oraz Monitor Polski jako główne źródła danych prawnych.

WALIDACJA

Obowiązkowa

Każdy wynik wymaga weryfikacji przez prawnika.

1. Podsumowanie funkcjonalności

System łączy LibreOffice Writer (Python 3.11.14), OpenAI o3-pro Batch API, Google Cloud Run, Firestore i Perplexity.ai. Umożliwia masowe przetwarzanie dokumentów prawnych. Dane prawne są pobierane z Dziennika Ustaw i Monitora Polskiego. System zapewnia buforowanie i monitorowanie statusu zadań oraz automatyczne uzupełnianie danych o aktualne podstawy prawne.

Zakres integracji

LibreOffice Writer pełni rolę interfejsu użytkownika, z którego uruchamiane są makra Python odpowiedzialne za przekazanie treści dokumentów do infrastruktury w

Źródła danych prawnych

Dane prawne są pobierane z Dziennika Ustaw i Monitora Polskiego, co umożliwia automatyczne uzupełnianie dokumentów o

Google Cloud Run oraz dalej do OpenAI o3-pro Batch API.

aktualne podstawy prawne oraz ich późniejszą weryfikację.

Monitorowanie i buforowanie

Firestore przechowuje status zadań, a wyniki są buforowane w Google Cloud Storage. System zapewnia śledzenie postępu przetwarzania oraz ponowne wykorzystanie wygenerowanych wyników.

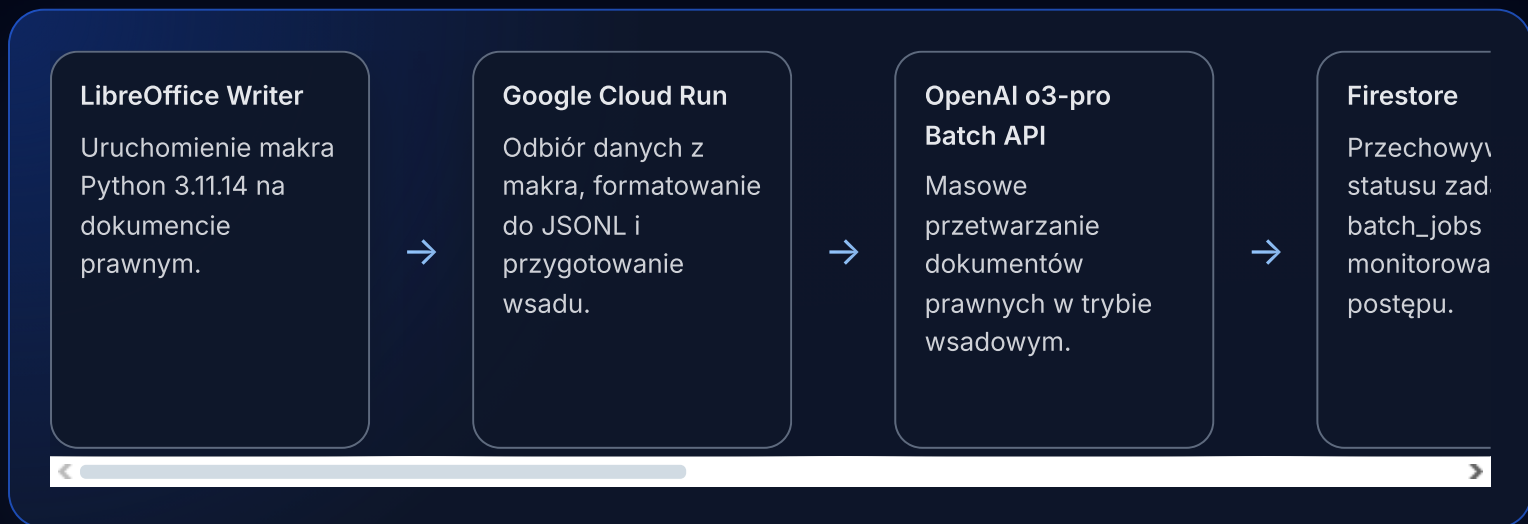
2. Metodologia działania

Poniżej przedstawiono kolejne etapy działania systemu od uruchomienia makra w LibreOffice Writer do pobrania wyników z Google Cloud Storage oraz uzupełnienia ich o aktualne akty prawne przy pomocy Perplexity.ai.

1. Użytkownik uruchamia makro Python w LibreOffice Writer.
2. Makro wysyła dane do usługi Google Cloud Run.
3. Cloud Run formatuje dane do JSONL i przekazuje do OpenAI Batch API.
4. Firestore przechowuje status zadań.
5. Wyniki są odbierane i zapisywane w Google Cloud Storage.
6. Użytkownik pobiera wyniki do dokumentu Writer.
7. Perplexity.ai uzupełnia dane o aktualne akty prawne.
8. System wymaga weryfikacji przez prawnika.

Schemat przepływu danych

Powiększ diagram



3. Fragmenty kodu i ich opisy

Wysłanie pliku do Google Cloud Run (Bash)

Ten kod wysyła plik z danymi do usługi Google Cloud Run przez `curl`.

```
curl -X POST \  
  -H "Content-Type: application/json" \  
  -d @plik.json \  
  https://twoj-cloud-run-url.run.app/endpoint
```

Kopiuuj kod

Sprawdzenie statusu zadania (Python 3.11.14)

Ten kod sprawdza status zadania w Firestore.

```
import firebase_admin  
from firebase_admin import credentials, firestore  
  
cred = credentials.ApplicationDefault()  
firebase_admin.initialize_app(cred)  
  
db = firestore.client()  
  
doc_ref = db.collection("batch_jobs").document("job_id")  
doc = doc_ref.get()  
  
if doc.exists:  
    print("Status:", doc.to_dict().get("status"))  
else:  
    print("Brak danych o zadaniu.")
```

Kopiuuj kod

Pobranie wyniku z Google Cloud Storage (PowerShell)

Ten kod pobiera plik wynikowy z Google Cloud Storage.

```
gsutil cp gs://twoj-bucket/nazwa_pliku.json .
```

4. Sugestie nowoczesnego projektu responsywnego

Poniższe wytyczne opisują kluczowe elementy nowoczesnego, responsywnego interfejsu dla systemu przetwarzania dokumentów prawnych, z naciskiem na czytelność, dostępność i prostotę obsługi.

Typografia i kontrast

- Użyj dużych, czytelnych czcionek i kontrastowych kolorów.
- Zapewnij możliwość powiększania tekstu.

Nawigacja i elementy interfejsu

- Stosuj proste przyciski i czytelne ikony.
- Strona powinna działać na komputerach i telefonach.

Dostępność treści

- Dodaj opisy alternatywne do grafik.
- Zapewnij łatwy dostęp do najważniejszych funkcji.

5. Wizualizacje

Wizualizacje procesu powinny wykorzystywać proste wykresy, takie jak diagram przepływu danych, pokazujące kolejne etapy przetwarzania. Opisy muszą być formułowane prostym językiem, a zastosowane kolory – stonowane i nierzuciające się w oczy.

Wytyczne dla diagramów

- Użyj prostych wykresów (np. diagram przepływu danych) pokazujących kolejne etapy procesu.
- Wykresy powinny być opisane prostym językiem.
- Kolory powinny być stonowane i nie rozpraszać.

1. Podsumowanie funkcjonalności

2. Metodologia działania

3. Fragmenty kodu

4. Sugestie projektu

5. Wizualizacje