KARTA PRACY UCZESTNIKA MODUŁ 4 ROZDZIAŁ 1

# Pełny przepływ systemu RAG

Od indeksu z M3 do odpowiedzi i źródeł

| Czas orientacyjny | Pracujesz z | Rezultat |
|---|---|---|
| 25 min | notebookiem R1 i indeksem z M3 | mapa_przeplywu.json |

Prześledzisz drogę pytania przez gotowy kod i zobaczysz, jakie dane przechodzą między kolejnymi funkcjami.

## Cel i rezultat rozdziału

Wyjaśnisz rolę każdego etapu RAG oraz sprawdzisz, czy indeks, teksty i metadane tworzą spójne wejście.

### Po zakończeniu masz

- mapę kolejności funkcji i ich wejść oraz wyjść
- sprawdzony manifest i przykładowy fragment
- własny opis roli aplikacji i modelu

## Zanim zaczniesz

Pracuj na rozpakowanej kopii Student. Uruchom Jupyter z tego katalogu. Wybierz MODE: ollama albo checkpoint. Najpierw wykonaj notebook 00_sprawdzenie_srodowiska.ipynb.

| Materiał | Do czego służy |
|---|---|
| notebooki/M12_M4_R1_przeplyw_rag_start.ipynb | Notebook startowy tego rozdziału. |
| dane/semantic_index/ | Kompletny, niezmieniony indeks M3; teksty i metadane. |

### Gdzie wrócić w prezentacji

Prezentacja M12-M4-v4: rozdział 1, „Pełny przepływ systemu RAG”. Odwołania dotyczą nazw rozdziałów, a nie dawnych numerów slajdów.

# Praca krok po kroku

**Krok 1 Sprawdź indeks** — Uruchom Przygotowanie oraz Wykonaj próby. Odczytaj manifest, wymiary i fragment.

- R1 nie wywołuje modelu; wyniki służą analizie przepływu.

**Krok 2 Prześledź dane** — Otwórz kod/m4_tools.py i wskaż wejście oraz wyjście etapów z mapy.

- Oddziel embedding pytania od tekstu przekazywanego do LLM.

**Krok 3 Sprawdź ciągłość z M3** — Porównaj nazwy trzech plików indeksu i metadane fragmentu.

- Nie zmieniaj chunk_id, document_id ani vector_index.

**Krok 4 Rozdziel odpowiedzialności** — Wyjaśnij, co wyszukuje aplikacja, co generuje model i co ocenia człowiek.

- Kontekst nie jest trwałym trenowaniem modelu.

**Krok 5 Zapisz własną analizę** — Uzupełnij ocenę jednej pozycji przeplyw i zapisz ją.

- W polu dowod_w_kontekscie podaj element manifestu lub kodu.

**Ważne: Przygotowanie tworzy nowy katalog OUT. Po nowej generacji oceń nowe wyniki. Po zmianie komentarzy uruchom tylko komórkę zapisu oceny.**

## Punkty kontrolne pracy

| Kiedy | Co sprawdzasz | Co zrobić przy problemie |
|---|---|---|
| Po przygotowaniu | Właściwy tryb, indeks i katalog OUT. | Wróć do notebooka 00 i rozpakuj cały pakiet. |
| Po próbach | Pełne wyniki; status oddzielony od oceny treści. | Przeczytaj prompt, kontekst, answer i generation. |
| Po ocenie | Własne uzasadnienia powiązane z result_id. | Zastąp UZUPEŁNIJ i uruchom sam zapis oceny. |

## Nazewnictwo i zapis wyników

Notebook tworzy wyniki/R1/<proba_id>/. Zachowaj mapa_przeplywu.json, proby.jsonl, ocena_*.json oraz notebook. Nazwę próby nadaje program; odczytaj ją z komunikatu Wyniki.

Zapis <proba_id> oznacza nazwę folderu nadaną automatycznie. Nie wpisuj dosłownie nawiasów. Zapisz notebook przez Ctrl+S i wskaż próbę przekazywaną do oceny.

## Lista kontroli przed zakończeniem

- □ Zachowałem pełny log i nie zmieniłem danych indeksu.
- □ Wskazałem tryb i identyfikatory ocenianych wyników.
- □ Zastąpiłem UZUPEŁNIJ własnym uzasadnieniem.
- □ Sprawdziłem podstawy odpowiedzi lub przyczynę braku generacji.
- □ Zapisałem notebook i wybrany katalog próby.

## Gdy pojawi się problem

| Objaw | Co sprawdzić |
|---|---|
| Brak plików lub importu | Uruchom notebook z katalogu Student albo Student/notebooki. |
| Brak odpowiedzi modelu | Sprawdź Ollama i modele lub jawnie wybierz checkpoint. |
| Brak żądania w checkpoincie | Przywróć domyślne pytania i parametry albo użyj ollama. |
| Ocena nie zapisuje się | Wypełnij wszystkie cztery pola dla każdego result_id. |

**Zachowaj oryginalny wynik i własną analizę. Błędna odpowiedź modelu też jest wartościowym wynikiem ćwiczenia.**