

• POD MASKĄ

Jak odpowiedź z czatu **staje się notatką?**

Najciekawsza część SumizAI to krótki pipeline, który uruchamia się po zakończeniu tury rozmowy. **Cztery etapy, każdy z jednym zadaniem**, i kilka decyzji w środku, które przesądzą o tym, czy Twoja biblioteka będzie coś warta za pół roku.

Aktualizacja: 2026-08-14 8 pytań 4 min czytania

Q. Co dzieje się, kiedy odpowiedź jeszcze przychodzi?

Twoje pytanie jest zapisywane, zanim model zostanie wywołany. Ta kolejność jest celowa: jeśli dostawca nie działa, ogranicza Cię limitem albo zwraca błąd, pytanie nadal tam jest, a nieudana tura zostaje oznaczona, żebyś mógł ponowić ją w miejscu, zamiast wpisywać wszystko od nowa.

Odpowiedź płynie potem token po tokenie kanałem zdarzeń serwera, więc czytasz ją w trakcie powstawania, a nie po ścianie tekstu na końcu.

Jedna konsekwencja warta zapamiętania: jeśli zamkniesz kartę w połowie odpowiedzi, serwer nie przerywa pracy. Odpowiedź modelu jest już opłacona u dostawcy, więc zostaje dokończona i zapisana razem z notatką, a nie wyrzucona.

Q. Etap pierwszy — co produkuje generator szkicu?

Szkic o strukturze, a nie bryłę tekstu. Model ma zwrócić **tytuł**, **podsumowanie** i **rozwiniecie** — i to stają się trzy widoczne części notatki.

Podsumowanie jest pisane tak, żeby dało się je czytać samo, na liście, kiedy szukasz właściwej notatki. Rozwiniecie to ta część, która musi przetrwać czytanie za rok, bez otaczającej ją rozmowy.

Obok szkicu zapisywane jest pełne oryginalne pytanie i pełna oryginalna odpowiedź jako materiał źródłowy notatki. Ten zapis jest tylko dopisywalny przez całe życie notatki — żadna późniejsza operacja go nie zmienia ani nie usuwa.

Q. Etap drugi — jak wybierany jest rozdział?

Przez wybór z ponumerowanej listy, nie przez nazywanie. Model dostaje istniejące rozdziały vaulta, ponumerowane, i ma odpowiedzieć numerem — albo poprosić o podrozdział pod jednym z nich.

Czyta skróconą wersję drzewa, a nie całość: skrót ograniczony do około 4 000 znaków, żeby decyzja pozostała opłacalna w vaultcie z tysiącami notatek.

Nowa gałąź powstaje wyłącznie, gdy nic nie pasuje, a nazwa vaulta nigdy nie jest rozdziałem — to dokładałoby korzeń nad korzeniem. Rozdziały o nazwach różniących się tylko ogonkami są scalane przy odczycie drzewa, więc przypadkowe *Warzywa* i *warzywa* nie stają się dwiema półkami.

Q. Etap trzeci — co porównuje sprawdzanie duplikatów?

Najpierw kandydatów. Wyszukiwanie pełnotekstowe i podobieństwo trigramowe w obrębie vaulta dają krótką listę notatek, które mogą być o tym samym. To tanie i działa w milisekundach nawet przy tysiącach notatek.

Potem ocenę. Drugi model — pomocniczy, nie ten, z którym rozmawiasz — czyta kandydata i nowy materiał i zwraca **ocenę od 0 do 1 z pisemnym uzasadnieniem**. Tanie wyszukiwanie zawęża pole, kosztowny czytelnik podejmuje decyzję.

Co dzieje się dalej, zależy od ustawienia vaulta: pytaj, scalaj automatycznie albo zawsze twórz nową. W trybie pytania decyzja może poczekać — nie blokuje Cię przed dalszą rozmową.

Q. Co dokładnie robi scalanie z istniejącą notatką?

Dopisuje do niej. Nowy materiał dołącza do notatki, a nowe pytanie i odpowiedź trafiają do materiału źródłowego obok tych, które już tam były.

Czego scalanie nie robi nigdy, to edytowanie ani usuwanie istniejącego źródła. Jest to wymuszone w kodzie, a nie umową: serwis i repozytorium po prostu nie mają dla niego ścieżki aktualizacji ani usuwania.

Powód jest dowodowy. Scalona notatka to twierdzenie złożone z kilku rozmów, a jedyne, co czyni takie twierdzenie sprawdzalnym później, to nieprzepisany zapis tego, co każda z tych rozmów faktycznie powiedziała.

Q. Etap czwarty — co łąduje na dysku?

Plik Markdown, nazwany slugiem tytułu, w katalogu `notes/` vaulta. Jeśli nazwa jest zajęta, aplikacja dokleja `(1)`, `(2)` i tak dalej — i pyta o to magazyn, a nie indeks.

Plik niesie frontmatter i treść notatki, i jest kompletny sam w sobie: wszystko, co wie indeks, da się z niego odbudować. Robi to operacja przeindeksowania, a w sporze plik jest traktowany jako poprawny.

Regenerowany jest też `base.md` vaulta, więc spis treści na dysku pozostaje zgodny z tym w aplikacji, z względnymi odnośnikami działającymi w dowolnym narzędziu Markdown.

Q. Skąd biorą się odnośniki w notatce?

Z tego samego etapu, który pisze treść. Model dostaje mapę do ośmiu kandydatów — tytuł do nazwy pliku — wybranych z wyników wyszukiwania, i ma wpleść odnośnik w zdanie, w którym naprawdę pasuje.

Każdy cel `.md` spoza tej mapy jest wycinany przy odbiorze. Model nie ma prawa wymyślić nazwy pliku, która wygląda sensownie, bo odnośnik do nieistniejącego pliku jest gorszy niż jego brak. Zewnętrzne odnośniki `http` zostają nietknięte.

Jest też akcja *przebuduj odnośniki* na poziomie vaulta, która przechodzi wszystko naraz — to narzędzie po imporcie większej partii materiału.

Q. Czy mogę wyłączyć automatyczne notatki?

Tak, per rozmowa. Część rozmów to myślenie na głos i nie powinna niczego po sobie zostawiać — przełącznik istnieje dokładnie po to.

Sama rozmowa nadal jest zapisywana — pytania i odpowiedzi, w kolejności — więc nic nie ginie. Nie uruchamia się pipeline: bez szkicu, bez odkładania, bez sprawdzania duplikatów, bez pliku.

W praktyce większość ludzi zostawia go włączonego i polega na scalaniu, które trzyma szum w ryzach, ale przełącznik jest, kiedy z góry wiesz, że sesja jest brudnopisem.