

newsy-make — agregator newsów z syntezą AI na platformie Make

Projekt zaliczeniowy kursu Automation+AI · autor: Marcin Jabłoński · sierpień 2026 · plan Make Free (≤ 1 000 kredytów/mies.) · Gemini API (darmowy tier) · Google Sheets · hosting seohost

Nagranie prezentujące działanie automatyzacji (maks. 5 min):

<https://drive.google.com/file/d/1PThYOeeJmwluD5HP2OWZRv178257Aujp/view?usp=sharing>

Blueprint scenariusza: newsy-make-blueprint.json (załącznik) · **Repozytorium:** fork projektu *newsy agregator*, katalog *newsy-make*

1. Szczegółowy opis projektu

newsy-make to automatyzacja, która trzy razy dziennie buduje bezstronne wydanie najważniejszych tematów dnia na podstawie nagłówków mediów z całego spektrum politycznego. Celem było przeniesienie rdzenia większego, działającego produkcyjnie projektu — agregatora z pełnymi syntezami i fact-checkingiem, generującego ponad 100 artykułów dziennie (rozdział 2) — na platformę Make w darmowym planie, czyli zmieszczenie całego procesu w tysiącu operacji miesięcznie, bez żadnych płatnych usług.

Scenariusz działa w cyklu co 8 godzin. Dwa moduły RSS pobierają nagłówki z Google News: feed A łączy operatorem OR portale lewicowe, centrowe i zagraniczne (OKO.press, Krytyka Polityczna, RMF24, Polsat News, TVN24, BBC), feed B — prawicowe (Do Rzeczy, Niezależna, wPolityce, TV Republika). Podział na dwa feedy nie jest przypadkowy: podczas testów jeden łączony feed zwracał niemal wyłącznie treści dwóch największych portali i sport, a media prawicowe znikają z wyniku. Agregatory tekstu zamieniają kilkadziesiąt wpisów w dwa bloki tekstu, endpoint `tresci.php` dokłada do nich pełne treści artykułów z cache serwera i składa bezpieczne żądanie do Gemini API (model `gemini-3.6-flash`, darmowy tier). Model grupuje nagłówki w maksymalnie sześć tematów, pisze do każdego pełną syntezę własnymi słowami na podstawie tych treści, wskazuje obecne orientacje źródeł i oznacza tematy relacjonowane jednostronnie (*blindspot*).

Odpowiedź modelu trafia przez HTTPS do endpointu `publish.php` na hostingu, który uwierzytelnia żądanie tokenem, twardo waliduje strukturę i renderuje statyczną stronę WWW (wydanie i archiwum). Ostatni moduł dopisuje wiersz logu do Google Sheets (data, liczba tematów, tytuły, status publikacji, zużyte tokeny). Każdy element, który może zawieść — źródło RSS, API modelu, serwer WWW, arkusz — ma własną ścieżkę obsługi błędów: awaryjny feed Bing News, ponowienia z opóźnieniem, alerty na Discordzie oraz filtry, które nie pozwalają wydać ani jednej operacji na pusty przebieg. Cały przebieg kosztuje dziewięć operacji, miesięcznie około 810 przy trzech przebiegach dziennie, co zostawia blisko jedną piątą limitu darmowego planu na ponowienia i testy.

2. Pochodzenie projektu: główny agregator newsów

newsy-make nie powstało od zera. Jest kursową, no-code'ową wersją większego systemu, który zbudowałem wcześniej i który działa produkcyjnie na tym samym hostingu: prywatnego agregatora newsów z syntezą i fact-checkingiem AI (newsy.marcinjablonski.com.pl, dostęp chroniony hasłem). To pipeline w Pythonie z bazą

SQLite, uruchamiany z croną na współdzielonym hostingu, bez własnego serwera — i to on wyznaczył cel i poprzeczkę jakości dla wersji na Make.

Jak działa dziś (stan na 26.08.2026):

- **Zbieranie:** co 30 minut z **46 źródeł** — polskie media lewicowe, centrowe i prawicowe oraz zagraniczne (BBC, Guardian, Telegraph, AP, DW, Fox News i inne), przez RSS i Google News; pełne teksty pobierane uczciwie (paywalle nigdy nie są obchodzone; przy ich wykryciu zostaje sam lead). Skala: 60–90 nowych artykułów na cykl, ok. 3 500 dziennie.
- **Grupowanie:** embeddingi artykułów i klastrowanie przyrostowe w tematy — na wektorach wycentrowanych (surowy cosinus newsów ma wspólne „tło” i sklejał różne wydarzenia), z automatyczną kontrolą spójności i podziałem klastrow, które zmieszały kilka wydarzeń. Ranking tematów: liczba unikalnych źródeł, różnorodność spektrum, świeżość.
- **Synteza i fact-checking:** dla każdego wybranego tematu model pisze artykuł od zera, własnymi słowami, z listą faktów oznaczonych w tekście znacznikami [F1], [F2]... i werdyktami (potwierdzone wielozródłowo / prawdopodobne / sporne / za wcześniej / niezwerifikowalne / fałszywe), wychwytuje rozbieżności między relacjami i osobno weryfikuje sprawdzalne wypowiedzi polityków (cytat musi być dosłowny — kod odrzuca każdy, którego nie ma w materiale). Twierdzenia sporne i jednoźródłowe idą do zewnętrznej weryfikacji z groundingiem w wyszukiwarce; obalone twierdzenie przepisuje artykuł, a nie tylko zmienia etykietę.
- **Bezstronność:** limit nowych tematów na godzinę z osobnymi slotami dla wydarzeń relacjonowanych wyłącznie przez jedną stronę sporu (żeby jednostronne tematy dostawały równie bezstronną syntezę), zakaz „przeglądów” łączących wiele wydarzeń, oznaczanie blindspotów rozumianych jako jednolity, nieneutralny przekaz wszystkich źródeł — z uwzględnieniem stronniczości mediów zagranicznych.
- **Cykl życia tematu:** nowe artykuły dołączają do istniejących tematów; aktualizacje są punktowe (podmiana nieaktualnych fragmentów), z dziennikiem „co nowego” i historią wersji; po 48 godzinach temat jest zamrażany, a nowa fala doniesień dostaje nowy artykuł. Duplikaty są scalane, nie syntetyzowane ponownie.
- **Warstwa dla czytelnika:** statyczna strona (wydanie, strony tematów, archiwum), strona fact-checkingowa z możliwością zlecenia dogłębnej weryfikacji pojedynczego twierdzenia, śledzenie przeczytanych tematów synchronizowane między urządzeniami, newsletter o 8:00 i 21:00 oraz alerty, gdy temat przekroczy 100 artykułów, panel administratora (limity na godzinę, wybór modelu, budżet, dodatkowe zadania AI).
- **Skala i koszt:** system generuje **ponad 100 artykułów dziennie** z fact-checkingiem (25.08.2026: ok. 250 syntez i weryfikacji z groundingiem oraz blisko 70 aktualizacji istniejących tematów). Generowanie idzie przez subskrypcję Google AI Pro (Antigravity CLI), embeddingi przez darmowy tier API, więc dzienny koszt API to grosze — poniżej 1 zł.

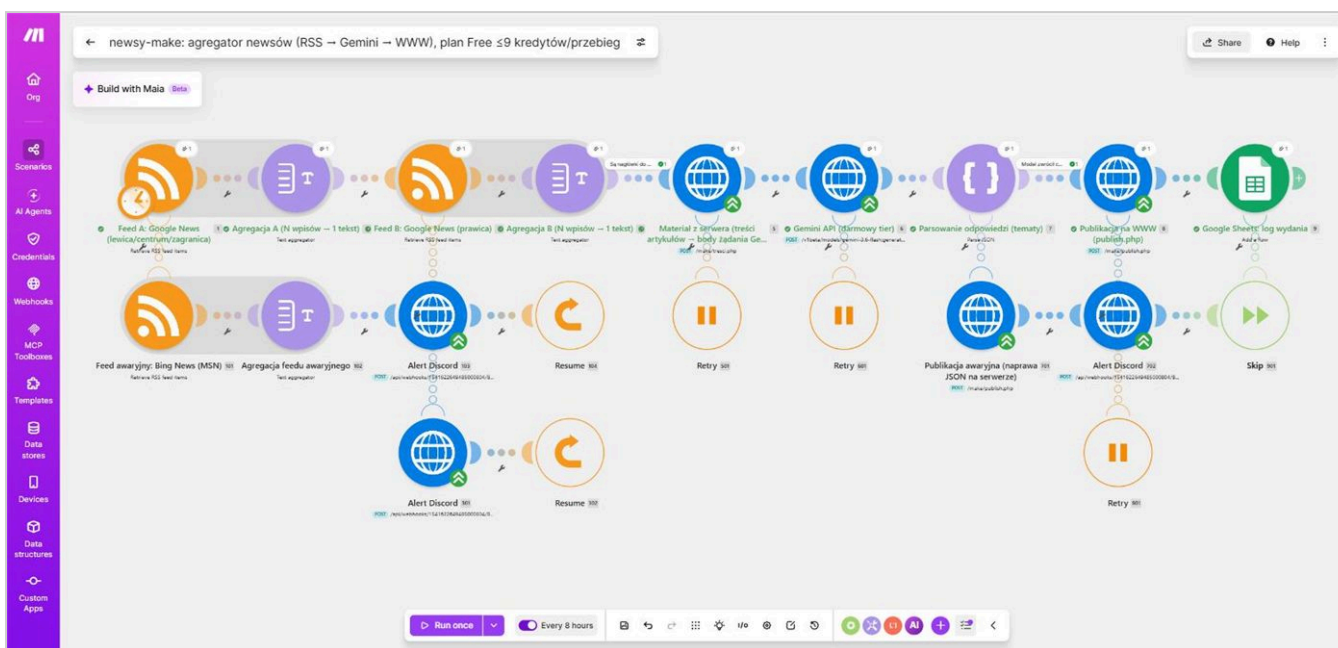
Wersja na Make celowo przenosi tylko rdzeń tego procesu — zbieranie, grupowanie, synteza, publikacja, log, alerty — na platformę no-code w limicie 1 000 operacji miesięcznie. Jedyne punkty styku obu systemów to endpoint `tresci.php`: zamiast pobierać artykuły w Make (operacja na artykule), scenariusz korzysta z cache pełnych tekstów, który główny agregator i tak utrzymuje dla tych samych portali (wyzwanie 9).

3. Architektura i przepływ danych

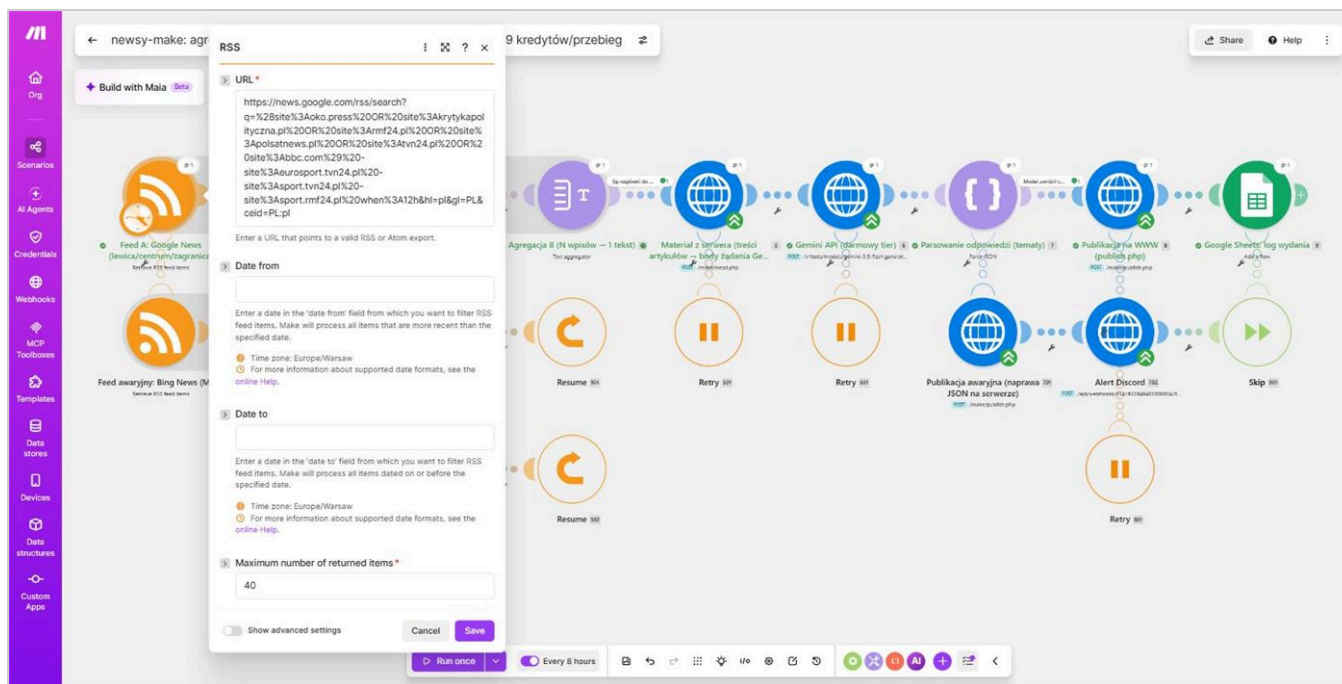
harmonogram Make (co 480 min)

- 1 RSS Google News – feed A (lewica/centrum/zagranica) — błąd → RSS Bing News (MSN) ► agregacja ► alert Discord ► RESUME
- 2 Text aggregator A (=40 wpisów → 1 tekst)
- 3 RSS Google News – feed B (prawica) ————— błąd → alert Discord ► RESUME (bez feedu B)
- 4 Text aggregator B
- 5 [filtr: są nagłówki] HTTP POST → tresci.php (treści z cache → body żądania) — błąd → BREAK: 2 ponowienia co 5 min
- 6 HTTP POST → Gemini API (tryb JSON, body z serwera) – błąd → BREAK: 3 ponowienia co 15 min
- 7 Parse JSON → tematy[] ————— błąd → publish.php (tryb napraw) ► alert Discord ► IGNORE
- 8 [filtr: ≥ 1 temat] HTTP POST → publish.php (token) – błąd → BREAK: 3 ponowienia co 10 min
- 9 Google Sheets – wiersz logu wydania ————— błąd → IGNORE (publikacja już się odbyła)

4. Zrzuty ekranu kluczowych elementów



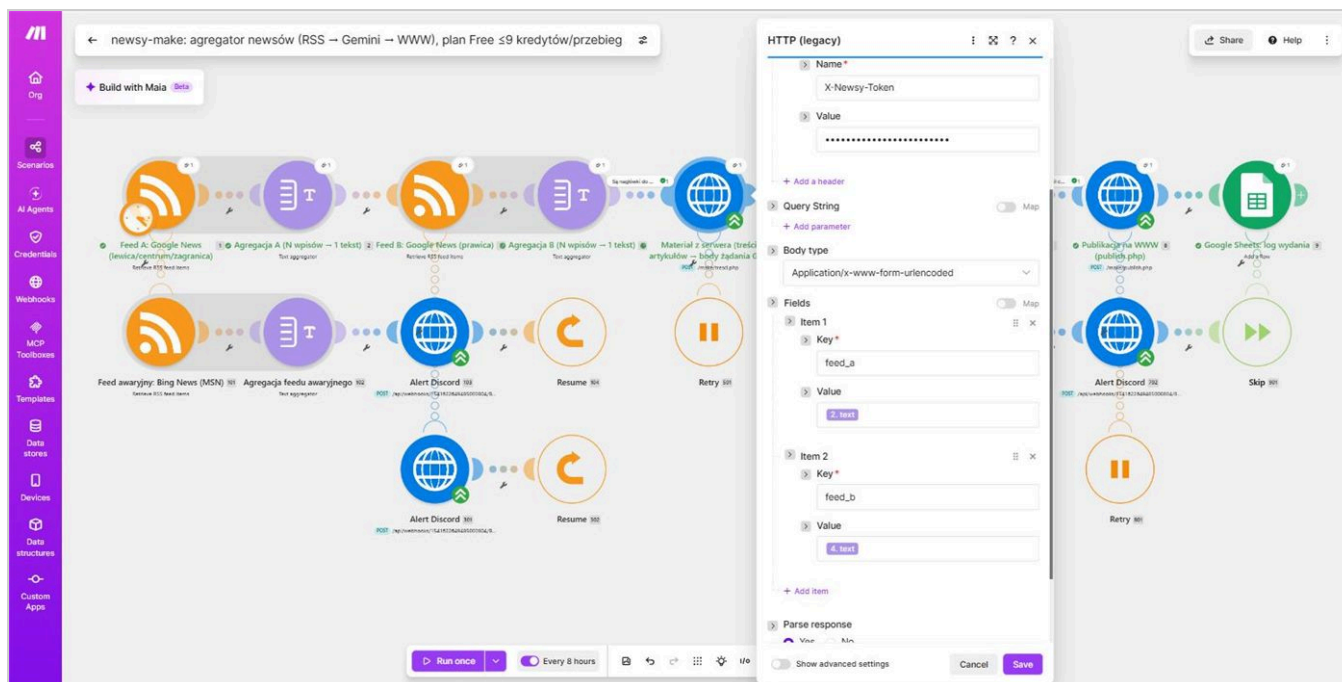
Rys. 1. Cały scenariusz. Łańcuch 9 modułów; nad modułami widoczne liczniki operacji po przebiegu testowym (po 1 na moduł — efekt agregacji od razu za RSS). Ścieżki obsługi błędów odchodzą w dół od modułów 1, 3, 6, 7, 8 i 9.



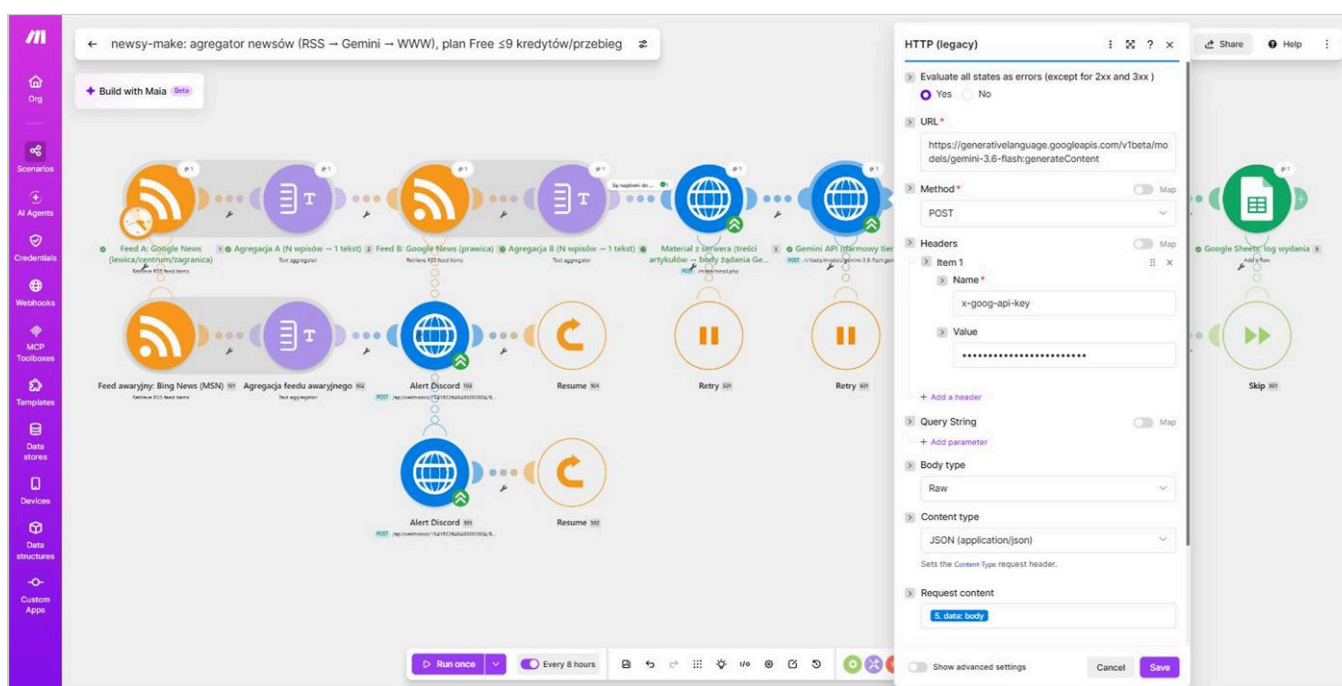
Rys. 2. Moduł RSS — feed A. Jeden adres Google News z operatorem OR obsługuje sześć portali i wyklucza serwisy sportowe (-site:); when:12h ogranicza wpisy do okna między przebiegami, co zastępuje deduplikację.



Rys. 3. Ścieżka błędu feedu A. Gdy Google News nie odpowiada, moduł RSS Bing News (odpowiednik MSN) dostarcza nagłówki zastępcze, wysyłany jest alert na Discorda, a dyrektywa Resume podstawia zebrany tekst jako wynik modułu 1 — przebieg trwa dalej.



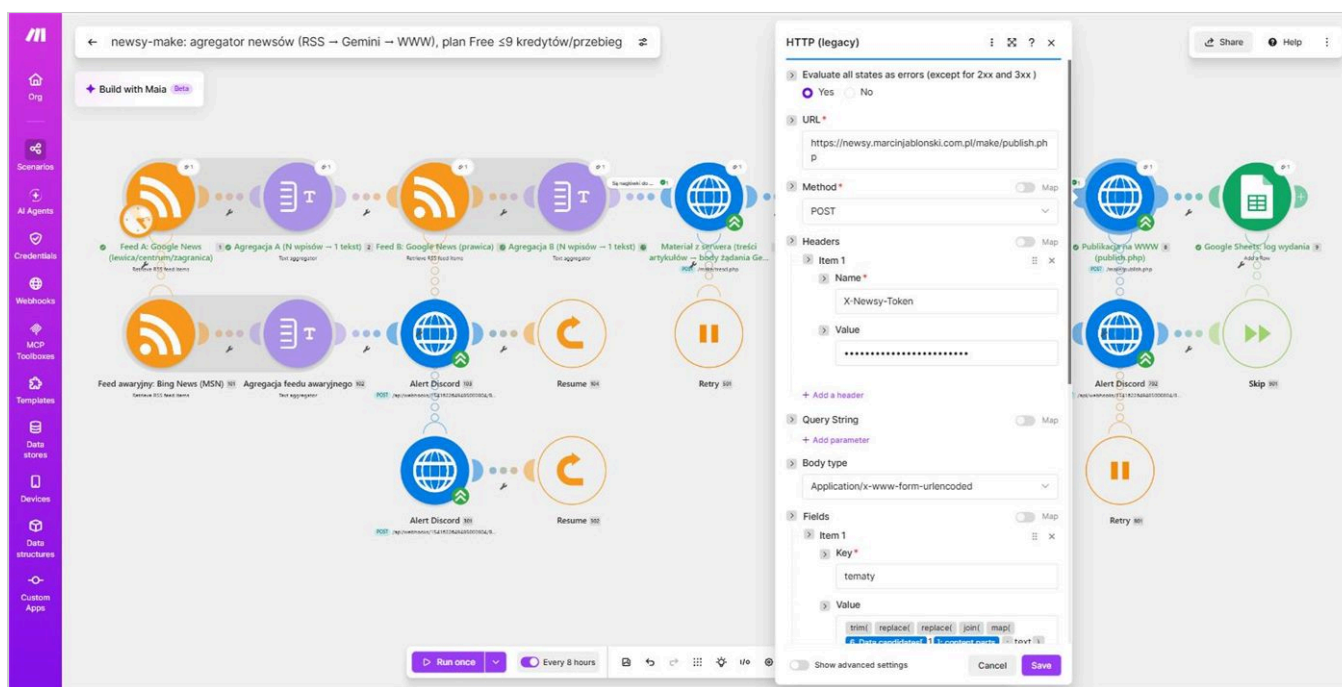
Rys. 4. HTTP → trecsi.php. Oba bloki nagłówków idą jako pola formularza; serwer dopasowuje je do treści artykułów w cache, skleja prompt z materiałem i zwraca gotowe ciało żądania Gemini (5.data.body). JSON składa PHP (json_encode), więc cudzystowy i nowe linie w nagłówkach niczego nie rozbijają.



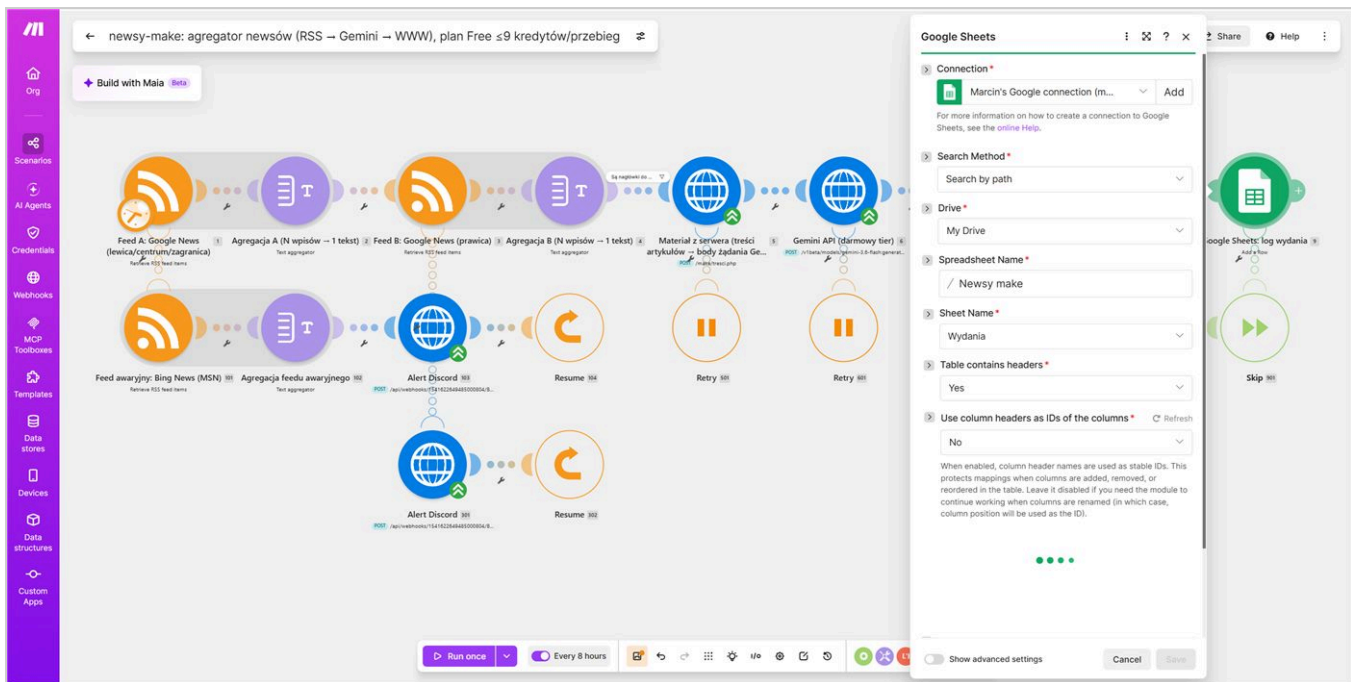
Rys. 5. HTTP → Gemini API. Klucz w nagłówku x-goog-api-key (nie w URL, żeby nie trafił do logów), Parse response włączone, weryfikacja certyfikatu włączona, limit czasu 300 s (synteza z pełnych tekstów trwa ok. minuty). Dyrektywa Break ponawia wywołanie 3× co 15 min przy błędach 429/5xx.



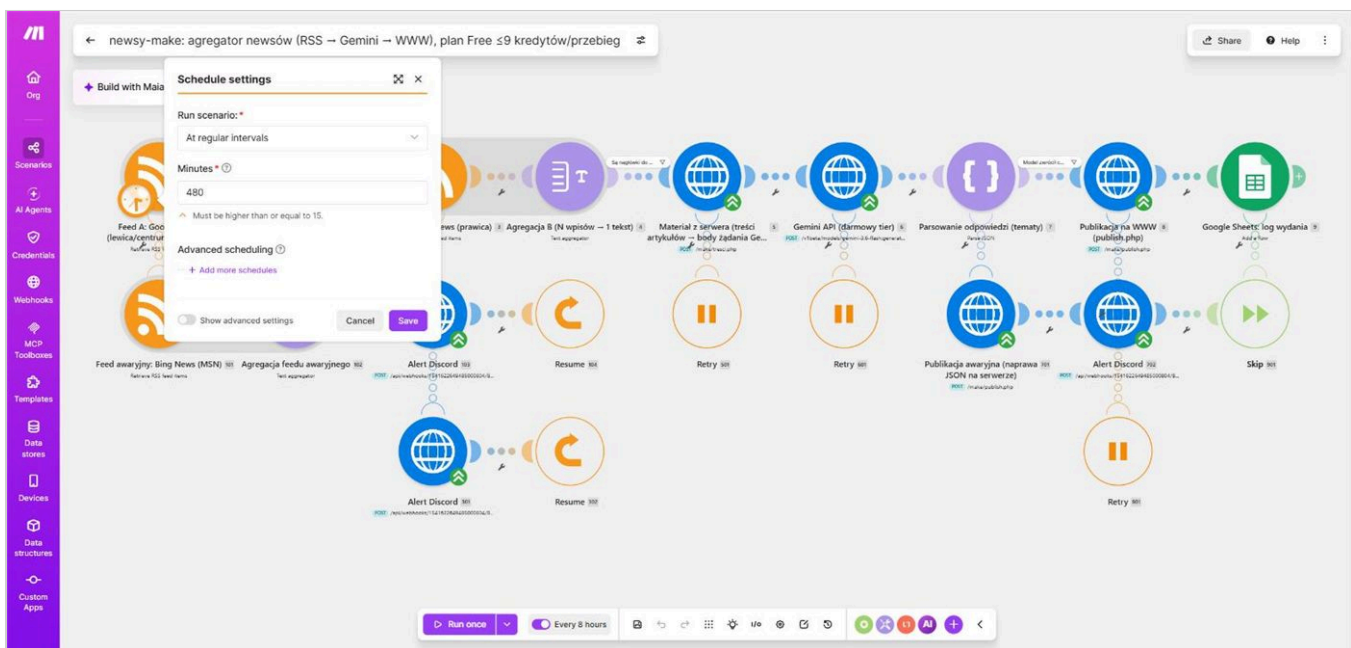
Rys. 6. Filtr „model zwrócił co najmniej 1 temat”. Filtry nie kosztują operacji — pusty przebieg kończy się po 4 zamiast 9 operacji.



Rys. 7. Publikacja. Odpowiedź modelu jest przekazywana jako pole formularza (razem z listą nagłówek, po której serwer dopasowuje linki) do `publish.php` z tokenem w nagłówku `X-Newsy-Token`; serwer sam renderuje HTML.



Rys. 8. Google Sheets — log wydań. Jeden wiersz na przebieg (1 operacja) zamiast wiersza na temat (8 operacji). Dyrektywa *Ignore*: awaria logu nie cofa publikacji.



Rys. 9. Harmonogram i ustawienia. Co 480 minut; włączone przechowywanie niedokończonych przebiegów (warunek działania ponowień), maksymalnie 3 kolejne błędy.

Ostatni przebieg: 25.08.2026, 15:08 · tematów: 5

lewica centrum prawica zagranica

Spór o finanse państwa po wetach prezydenta Karola Nawrockiego

Według wycień Ministerstwa Finansów, sześć wet prezydenta Karola Nawrockiego obniży tegoroczne dochody budżetowe o ponad 8 mld zł. Przedstawiciele obozu rządzącego krytykują decyzje prezydenta, podczas gdy Kancelaria Prezydenta przekonuje, że powstrzymanie podwyżek podatków pozostawia pieniądze w kieszeniach obywateli.

[Czytaj pełną syntezę →](#)

lewica centrum prawica zagranica

Afera korupcyjna na Ukrainie i międzynarodowe reakcje wokół otoczenia Zełenskigo

Była wiceszefowa Biura Prezydenta Ukrainy Iryna Mudra została aresztowana na 60 dni w związku ze śledztwem dotyczącym prania pieniędzy. Komisja Europejska przypomniała Kijowowi o zasadzie zerowej tolerancji dla korupcji, podczas gdy komentatorzy wskazują na rosnące spory polityczne i spadek zaufania do prezydenta Wołodymyra Zełenskigo.

3 źródeł · 330 słów syntezy

lewica centrum prawica zagranica

Nowy sondaż Sejmu i deklaracje utworzenia nowej partii przez Mateusza Morawieckiego

Sondaż United Surveys dla Wirtualnej Polski wskazuje, że do Sejmu weszłoby sześć ugrupowań, a KO utraciłaby większość rządową na rzecz koalicji prawicy. Biorący udział w badaniu były premier Mateusz Morawiecki zapowiedział utworzenie nowej partii Rozwój Plus i zadeklarował chęć ponownego objęcia urzędu szefa rządu.

2 źródeł · 313 słów syntezy

lewica centrum prawica zagranica

Prokuratura przygotowała wnioski o uchylenie immunitetów Hołowni i Kobosce

Śląscy prokuratorzy skierowali do Prokuratury Krajowej poprawione projekty wniosków o uchylenie immunitetów Szymonowi Hołowni oraz Michałowi Kobosce. Sprawa ma związek ze śledztwem dotyczącym nieprawidłowości w funkcjonowaniu uczelni Collegium Humanum.

1 źródło · 279 słów syntezy

lewica centrum prawica zagranica

Grzegorz Braun wezwany do IPN w charakterze podejrzanego

Posel Grzegorz Braun został wezwany do oddziału IPN w Krakowie na

Rys. 10. Wynik — strona WWW. Wydanie z tematami, paskiem pokrycia orientacji (lewica / centrum / prawica / zagranica) i oznaczeniem tematów jednostronnych.

Newsy make ☆ 📁 ☁					
Plik Edytuj Widok Wstaw Formatuj Dane Narzędzia Rozszerzenia Pomoc					
🔍 Menu ↶ ↷ 🖨 📄 100% ▾ zł % ⬇ ⬆ 123 Domy... ▾ - 10 + B I ↺ A 🗑 📑 📊 📈					
G3	fx				
	A	B	C	D	E
1	Data	Tematów	Tytuły	Publikacja OK	Tokeny
2	2026-08-25 15:04	5	Wezwanie Grzegorza Brauna do IPN w charakterze podejrzanego o negowanie zbrodni w Auschwitz Wylczenia Ministerstwa Finansów dotyczące skutków wet prezydenta Karola Nawrockiego i spór polityczny Konfrontacja polityczna wokół działań ministra Waldemara Żurka na rzecz ścigania mowy nienawiści Aresztowanie Iryny Mudrej w Kijowie i ostrzeżenia Komisji Europejskiej ws. praworządności na Ukrainie Prokuratura przygotowuje wnioski o uchylenie immunitetów Szymonowi Hołowni i Michałowi Kobosce		43282
3	2026-08-25 15:08	5	Spór o finanse państwa po wetach prezydenta Karola Nawrockiego Afera korupcyjna na Ukrainie i międzynarodowe reakcje wokół otoczenia Zełenskigo Nowy sondaż Sejmu i deklaracje utworzenia nowej partii przez Mateusza Morawieckiego Prokuratura przygotowała wnioski o uchylenie immunitetów Hołowni i Kobosce Grzegorz Braun wezwany do IPN w charakterze podejrzanego		36069
4					

Rys. 11. Arkusz Google. Log wydań z liczbą tematów i użytymi tokenami — podstawa do kontroli kosztów.

History								
Started ▾	Run name ⓘ ▾	Trigger / Activity	Status ▾	Duration ▾	Credits ▾	Operations ▾	Data transfer ▾	Source run
25 sie 2026, 15:07:21		🕒 Schedule	Success	56 seconds	9	9	423,1 KB	🔄 ...
25 sie 2026, 15:05:18		🕒 Scenario was activated by Marcin Jablonski.						
25 sie 2026, 15:03:10		▶ Manual run by Marcin Jablonski	Success	1 minute	9	9	451,9 KB	🔄 ...
25 sie 2026, 15:02:56		✎ Scenario was edited by Marcin Jablonski.						
25 sie 2026, 14:55:35		▶ Manual run by Marcin Jablonski	Warning	1 second	5	5	291,0 KB	🔄 ...
25 sie 2026, 14:55:31		✎ Scenario was edited by Marcin Jablonski.						

Rys. 12. Historia przebiegów. Zużycie operacji na przebieg widoczne w Make.

5. Mechanizmy bezpieczeństwa (odporność na błędy)

Punkt awarii	Mechanizm	Efekt
Google News nie odpowiada / zmienia format (feed A)	Ścieżka błędu: RSS Bing News jako źródło awaryjne → agregacja → alert Discord → Resume	przebieg kończy się poprawnym wydaniem z zapasowego źródła; administrator wie o awarii
Feed B (prawica) niedostępny	alert Discord → Resume z komunikatem zastępczym	wydanie powstaje z feedu A; model dostaje informację o braku feedu B i nie „zgaduje” drugiej strony
Brak nagłówków (np. noc, oba feedy puste)	filtr „są nagłówki” przed modułem HTTP → <code>tresci.php</code>	koniec przebiegu po 4 operacjach; żadnego wywołania API
Gemini: limit 429, błąd 5xx, timeout	Break z automatycznym ponowieniem 3× co 15 min; kody 4xx/5xx traktowane jako błąd modułu; timeout 300 s; przechowywanie niedokończonych przebiegów	chwilowe problemy API nie tracą przebiegu; trwale widać w historii jako niedokończone
Model zwraca niepoprawny JSON	ścieżka błędu Parse JSON: surowa odpowiedź → <code>publish.php</code> w trybie „napraw” (deterministyczny skaner nawiasów) → alert → Ignore	wydanie powstaje z naprawionej odpowiedzi (oznaczone na stronie); gdy naprawa się nie uda, poprzednie wydanie zostaje
Model zwraca 0 tematów	filtr „≥ 1 temat” przed publikacją	brak pustej publikacji i brak operacji na Sheets
Serwer WWW niedostępny	Break 3× co 10 min	publikacja ponawiana, wydanie nie ginie
Awaria Google Sheets / wygaśnięte połączenie	Ignore	publikacja już się odbyła; log uzupełni następny przebieg
Nieautoryzowane żądania do endpointu	token w nagłówku porównywany stałoczasowo (<code>hash_equals</code>), HTTPS, basic auth na stronie, <code>config.php</code> i archiwum niedostępne z sieci	tylko Make może publikować
Złośliwa lub błędna treść od modelu (XSS, złe URL-e)	walidacja typów i długości, biała lista orientacji, tylko <code>http(s)://</code> , limit 12 tematów × 8 źródeł, limit 200 KB, render wyłącznie przez <code>htmlspecialchars</code>	model nigdy nie wstrzykuje HTML-u; przetestowane ładunkiem z <code></code> , cudzysłowami i <code>ftp://</code>
Wyciek sekretów	klucz Gemini i token wyłącznie w polach modułów (blueprint zawiera placeholder); klucz w nagłówku, nie w URL	eksport blueprintu i logi nie zawierają sekretów

6. Optymalizacja wykorzystania zasobów

Decyzja	Bez optymalizacji	Z optymalizacją
Agregacja tekstu od razu za modułem RSS	każdy z ~40 wpisów przechodzi przez 7 kolejnych modułów: ~280 op./przebieg	9 op./przebieg
Jeden feed Google News z OR dla kilku portali	6 modułów RSS + 6 agregatorów: 12 op.	2 + 2 = 4 op. (i lepszy balans źródeł niż 1 feed)
Jedno wywołanie modelu na przebieg (wszystkie nagłówki naraz)	1 wywołanie na temat: 8 op. + 8 zapytań API	1 op., 1 zapytanie
Odpowiedź modelu przekazywana 1:1 do serwera	Iterator + moduły budujące HTML: ~10 op.	0 op. — HTML renderuje PHP
Treści artykułów z cache serwera (1 moduł HTTP zamiast Create JSON)	pobieranie każdego artykułu modułem HTTP: 40–80 op.	1 op. — i model pisze z pełnych tekstów
Jeden wiersz logu na przebieg	Iterator + wiersz na temat: 8 op.	1 op.
Deduplikacja przez when:12h w zapytaniu	Data store: 2 op. (odczyt + zapis) na przebieg	0 op.
Filtry zamiast routerów	—	filtry nie liczą operacji
Naprawa uszkodzonego JSON na serwerze (skaner nawiasów w PHP)	drugie wywołanie modelu „popraw JSON”: +2 op. i zapytanie API	+1 op. tylko przy awarii, 0 zapytań
Harmonogram 3 × dobę	co 15 min (minimum planu): ~26 000 op./mies.	≈ 810/mies. + zapas 19 % na ponowienia

Bilans miesięczny: 9 op. × 3 przebiegi × 30 dni = **810 kredytów**. Ścieżki błędów kosztują tylko wtedy, gdy się wykonują (1–3 op.), ponowienia Break maksymalnie 3 na przebieg. Przy awariach w co piątym przebiegu zużycie nie przekracza ~950 kredytów; wariant oszczędny 2 × dobę to 540. Gemini: 3 zapytania dziennie przy limicie darmowego tieru liczonym w setkach. Transfer danych: ~150 KB na przebieg wobec 512 MB limitu planu.

7. Proces myślowy: wyzwania, decyzje, alternatywy

Wyzwanie 1 — jeden feed nie oddaje spektrum

Plan zakładał jeden feed Google News z listą wszystkich portali (1 operacja). Sonda pokazała, że Google zwraca maksymalnie 100 pozycji uporządkowanych według własnej trafności: w wyniku dominowały TVN24 i Eurosport (43 + 39 pozycji), a portale prawicowe nie pojawiły się wcale. Alternatywy: (a) osobny moduł RSS dla każdego portalu — 6–8 operacji i utrata budżetu; (b) jeden feed z węższą listą — nadal niesymetryczny; (c) **dwa feedy podzielone według orientacji** plus wykluczenie serwisów sportowych operatorem `-site:.` Wybrałem (c): koszt 2 operacji więcej, ale gwarantowana obecność obu stron sporu — bez tego bezstronna synteza nie ma sensu.

Wyzwanie 2 — mnożenie operacji przez pakiety (bundles)

Moduł RSS zwraca każdy wpis jako osobny pakiet, więc kolejne moduły wykonałyby się kilkadziesiąt razy. Rozwiązaniem jest agregator ustawiony bezpośrednio za RSS; dwa feedy w jednym łańcuchu działają, bo po pierwszym agregatorze płynie już tylko jeden pakiet, więc drugi RSS wykonuje się raz. To najważniejsza decyzja kosztowa projektu (9 zamiast ~280 operacji).

Wyzwanie 3 — bezpieczne budowanie żądania do API

Nagłówki zawierają cudzysłowy i znaki specjalne; ręcznie sklepany JSON w module HTTP psuł się na pierwszym tytule z cytatem. Alternatywy: funkcje `replace()` (kruche), `escapeHTML()` (zmienia treść), przeniesienie wywołania modelu na serwer PHP (ukrywa logikę poza Make — sprzeczne z celem zadania). Wybrałem moduł **Create JSON** ze strukturą danych: 1 operacja, poprawne escapowanie gwarantowane przez platformę, a klucz API zostaje w Make. W wersji 3 tę rolę przejął serwer: `tresci.php` zwraca gotowe ciało żądania złożone przez `json_encode`, a moduł Create JSON zniknął (wyzwanie 9) — klucz API nadal jest wyłącznie w Make.

Wyzwanie 4 — gdzie renderować stronę

Budowanie HTML w Make wymagałoby iteratora po tematach (operacja na temat). Zamiast tego Make wysyła surowy JSON, a niewielki endpoint PHP na istniejącym hostingu waliduje go i renderuje stronę. Bonus bezpieczeństwa: tekst modelu nigdy nie staje się kodem HTML. Alternatywa (Google Sheets jako „baza” i strona czytająca arkusz w JavaScript) była tańsza w operacjach, ale gorsza w kontroli treści i wymagała publicznego arkusza.

Wyzwanie 5 — źródło zapasowe

Google News to pojedynczy punkt awarii. MSN nie ma już działającego RSS (DNS `rss.msn.com` nie istnieje, ścieżki zwracają 404), ale **Bing News** — silnik stojący za MSN — wystawia RSS z operatorami `site:`. Dyrektywa *Resume* pozwala podstawić dane z zapasowego feedu jako wynik modułu, który zawiódł, więc reszta scenariusza nie wie o awarii.

Wyzwanie 6 — model i darmowy tier

Modele `gemini-2.5-*` okazały się niedostępne dla nowych użytkowników („no longer available to new users”); aktualne to `gemini-3.7-flash`, `gemini-3.6-flash` i `gemini-3.5-flash-lite`. Pierwsza wersja używała wersji lite i wymuszonego JSON-a — wystarczyło to do grupowania nagłówków. Gdy wymaganiem stały się pełne syntezы, przeszedłem na `gemini-3.6-flash` w trybie JSON, bez narzędzi (3.7 w dniu testów zwracał z API 503 „high demand”); zmiana modelu to jedna nazwa w URL; 3 zapytania dziennie po ok. 25 tys. tokenów mieszczą się w darmowym tierze z dużym zapasem. Klucz pochodzi z projektu AI Studio bez billingu, więc koszt to 0 zł.

Wyzwanie 7 — dwa feedy zamiast jednego to inny prompt

Model dostaje oba bloki z etykietą orientacji i tabelę „portal → orientacja”, ma preferować tematy opisane przez ≥ 2 źródła i wprost oznaczać jednostronne (*blindspot*). Pobieranie pełnych tekstów artykułów nie mieści się w 1 000 operacji (moduł HTTP na każdy artykuł), dlatego pełne treści dokłada do promptu serwer z własnego cache (wyzwanie 9).

Wyzwanie 8 — model zwrócił niepoprawny JSON (incydent 25.08, 03:39)

Przebieg zatrzymał się na module Parse JSON: w jednym z obiektów `zrodla` brakowało nawiasu zamykającego, a model przepisywał 300-znakowe adresy Google News, co mnożyło okazje do błędu. Analiza wskazała dwie przyczyny: tryb JSON nie był aktywny (nazwa pola w konwencji SDK zamiast REST) i prompt kazał kopiować URL-e. Rozwiązanie w trzech warstwach: (1) model podaje tylko nazwę portalu i tytuł nagłówka, a link dopasowuje serwer po znormalizowanym tytule; (2) ścieżka błędu modułu Parse JSON wysyła surową odpowiedź do `publish.php` w trybie „napraw” — deterministyczny skaner nawiasów domyka struktury, wstawia brakujące przecinki, zamienia surowe znaki nowej linii na sekwencje `\n`, a w ostateczności obcina do ostatniego kompletnego tematu; (3) alert na Discordzie, że wydanie powstało z naprawionej odpowiedzi (widać to też na stronie). Alternatywa — drugie wywołanie modelu „popraw ten JSON” — kosztowałaby 2 operacje i zapytanie API przy każdej awarii; naprawa w PHP to 1 operacja i powtarzalny wynik.

Wyzwanie 9 — pełne syntezy zamiast streszczeń nagłówków

Wymaganie: trzy przebiegi dziennie z artykułami dogłębnymi i kompletnymi strukturalnie, nadal bez pobierania treści modułami Make (operacja na artykuł). Pierwszy pomysł — grounding `google_search` w wywołaniu Gemini — upadł na cenniku: darmowy tier zwraca dla niego 429 z zerowym limitem (sprawdzone na wszystkich modelach 3.x). Drugi — narzędzie `url_context`, w którym model sam czyta wskazane strony — działa na darmowym tierze, ale linki Google News to przekierowania, a po ich odkodowaniu portale (RMF24, TVN24) blokują fetcher Google (`URL_RETRIEVAL_STATUS_ERROR`). Rozwiązanie: **cache treści po stronie serwera**. Endpoint `tresci.php` dostaje oba bloki nagłówków, odrzuca strony tagów i autorów, dopasowuje tytuły (dokładnie, a potem rozmyto — pokrycie słów, indeks odwrotny) do artykułów w bazie głównego agregatora na tym samym koncie hostingowym, który co 30 minut pobiera pełne teksty tych samych portali, i zwraca gotowe ciało żądania: prompt + do 60 artykułów z treścią (~85 tys. znaków) + pozostałe nagłówki. Moduł HTTP wysyła je 1:1, model pracuje w trybie JSON (bez narzędzi), więc odpowiedź jest poprawna strukturalnie za pierwszym razem. Bilans: moduł Create JSON zastąpiony jednym modułem HTTP — nadal 9 operacji. Test 25.08: 46 z 72 nagłówków z pełną treścią, 5–6 syntez po 250–360 słów z atrybucją i linkami do źródeł.

Czego się nauczyłem

- Budżet operacji projektuje się na etapie architektury (miejsce agregatora, filtry, brak iteratorów), nie po fakcie.
- Każdy moduł zewnętrzny potrzebuje własnej odpowiedzi na pytanie „co, jeśli zawiedzie?” — i to jest inna odpowiedź dla źródła danych (fallback), API (ponowienie), logu (ignoruj).
- Sondowanie realnych danych (feedy, limity modeli, certyfikat serwera) zmieniło trzy założenia projektu, zanim powstał pierwszy moduł.
- Odpowiedź modelu językowego jest niezaufana także składniowo — walidację i naprawę robi deterministyczny kod po stronie serwera, nie kolejny prompt.
- „Darmowy tier” trzeba sprawdzić per funkcja, nie per model: generowanie jest darmowe, grounding nie — dowiedziałem się tego dopiero z kodu 429 w historii przebiegu.

8. Załączniki

- `newsy-make-blueprint.json` — blueprint scenariusza w formacie eksportu Make (sekrety zastąpione placeholderami `WPISZ_*`).

- `tresci.php` — endpoint dopasowujący nagłówki do treści artykułów i składający ciało żądania Gemini.
- `prompt.txt` — pełna treść promptu (wgrywana na serwer).
- `publish.php`, `.htaccess`, `styl.css` — endpoint publikujący z walidacją i styl strony.
- `narzedzia/` — generator blueprintu, uploader FTPS, testy endpointów.
- Wszystkie powyższe pliki w formie tekstowej: `newsy-make-zalacznik-kod.pdf` (drugi załącznik — formularz przyjmuje tylko PDF i obrazy).
- `README.md` — instrukcja importu i konfiguracji krok po kroku.