

3. Czat z Orin, Kuźnia, Muzyka i Stream — Rozrywka w Warsztacie

“ Źródło: [repo afterlife/docs/api/03-czat-kuznia-muzyka-stream-rp.md](#) — edytuj w repo, nie tutaj.

Czat z Orin, Kuźnia, Muzyka i Stream — Rozrywka w Warsztacie



Trąca łapką mikrofon i podkręca suwaki na mikserze! Słyszysz mnie dobrze, nyah~? Kiedy nie zjeżdżamy wózkem do kopalni **Nidarelu**, w naszym Warsztacie i tak cały czas coś gra, świeci albo maluje obrazy! Na tej stronie naszej książki zebrałam wszystkie trzynaście ścieżek API, którymi rozmawiasz ze mną w okienku czatu, rozpalasz ogień w **Kuźni** (ComfyUI), wrzucasz kawałki na wspólną kolejkę muzyczną i oglądasz strumienie wideo na żywo! *strzyże uszami*

Czat ze mną i Kuźnia obrazów (4 ścieżki)

Moje okienko czatu na stronie (`OrinAIWindow`) oraz komendy na Discordzie korzystają ze wspólnego mózgu i wspólnej kolejki! Ponieważ karta graficzna Majstra obsługuje jednocześnie modele językowe (Ollama), generowanie grafiki (ComfyUI) oraz same gry, nad wszystkim czuwa bramka **Kuźni** (`lib/forge.ts`).

Endpoint	Metody	Dostęp	Opis działania i parametry
----------	--------	--------	----------------------------

<code>/api/chat</code>	GET, POST	Ciasteczko sesji Discord	Główny punkt rozmowy ze mną! GET zwraca historię kontekstu i status; POST przyjmuje wiadomość, sprawdza czy pytasz o status serwerów/Vikunji, steruje światłami Home Assistant (dla uprawnionych), generuje obrazek lub odpowiada przez wybrany model AI. Jeśli GPU jest zajęte, wrzuca zadanie do kolejki <code>orinQueue</code> .
<code>/api/chat/queue/[id]</code>	GET, POST	Ciasteczko sesji Discord	Obsługa kolejki moich odpowiedzi! GET pozwala przeglądarce lub botowi odpytywać o status zadania (<code>pending</code> , <code>resolved</code> , <code>error</code>), tekst odpowiedzi i wygenerowany obrazek. Zwraca 404, jeśli zadanie nie należy do ciebie! POST oznacza odpowiedź jako dostarczoną (<code>markOrinQueueItemDelivered</code>), żebym po restarcie nie wysłała jej na Discordzie dwa razy.
<code>/api/generate</code>	POST	Sesja Discord + uprawnienie w grafie AI / rola <code>Zwłoki w Taczce</code>	Bezpośrednie wykuwanie mediów w Kuźni (wywoływane m.in. przez komendę <code>/generate</code>). Przyjmuje <code>{ preset, prompt }</code> . Sprawdza wolne miejsce na dysku AI (blokada powyżej 95%), rezerwuje slot w Kuźni (<code>acquireForgeSlot</code>) i zwraca surowe bajty wygenerowanego pliku!
<code>/api/forge-request</code>	GET, POST	Sesja Discord (GET), Uprawnieni do generowania / Majster (POST)	Sterowanie bramką Kuźni w trybie GAMING! GET zwraca stan Kuźni (<code>state</code>), liczbę czekających (<code>waiters</code>) i oczekującą prośbę (<code>pending</code>). POST pozwala otworzyć lub zamknąć Kuźnię — a jeśli Majster akurat w coś gra (<code>isLikelyGaming</code>), wysła mu prywatną wiadomość z przyciskami zatwierdzenia lub odrzucenia prośby!

Jak działa ochrona Kuźni i dysku?

Zanim pozwolę ci wygenerować obrazek przez `/api/generate` albo `/api/chat`, sprawdzam trzy rzeczy, nyah~:

1. Czy w grafie uprawnień (`canInvokeOrLegacy`) twoja rola ma połączenie `can_invoke` do wybranego presetu `capability:orin.image.<preset>`.
2. Czy dysk roboczy AI nie przekroczył progu **95% zapęłnienia** (`AI_DISK_BLOCK_PCT`).
3. Czy **Kuźnia** nie jest wygaszona z powodu gry Majstra! Zamknąć Kuźnię po skończonej pracy może każdy uprawniony, ale zatwierdzić prośbę podczas rozgrywki może wyłącznie sam **Majster** (`isOwner`).

Wspólna Kolejka Muzyczna (6 ścieżek)

Jak już wiesz z poradnika na wiki, mam jedną wspólną kolejkę muzyczną dla strony **ariku.pl/music** i kanałów głosowych na Discordzie! *kiwa głową w rytm muzyki* Najpierw szukam utworu w naszej lokalnej bibliotece, a jeśli go nie ma — pobieram go z YouTube, indeksuję i dopiero puszczam w eter.

Endpoint	Metody	Dostęp	Opis działania i parametry
<code>/api/music/search</code>	<code>GET</code>	Ciasteczko sesji Discord	Przeszukuje naszą zaindeksowaną bibliotekę muzyczną po tytule lub artyście i zwraca gotowe do odtworzenia utwory.
<code>/api/music/youtube</code>	<code>GET</code>	Ciasteczko sesji Discord	Wyszukuje filmy lub playlisty na YouTube według wpisanej frazy albo wklejonego linku URL.
<code>/api/music/play</code>	<code>POST</code>	Ciasteczko sesji Discord	Dodaje wybrany utwór do kolejki! Jeśli pochodzi z naszej biblioteki — trafia tam od razu; jeśli z YouTube — uruchamia zadanie pobierania i indeksacji w tle.

Endpoint	Metody	Dostęp	Opis działania i parametry
<code>/api/music/queue</code>	GET, DELETE	Ciasteczko sesji Discord	GET zwraca aktualnie grany utwór oraz całą listę zaplanowanych piosenek w kolejce. DELETE pozwala wyczyścić kolejkę lub usunąć z niej własne utwory („usuń moje”).
<code>/api/music/control</code>	POST	Ciasteczko sesji Discord	Sterowanie odtwarzaczem na żywo: przyjmuje komendy pause (pauza), resume (wznów) oraz skip / next (następny utwór).
<code>/api/music/jobs</code>	GET	Ciasteczko sesji Discord	Zwraca status trwających zadań pobierania z YouTube (pobieram... → indeksuję... → w kolejce), dzięki czemu pasek postępu na stronie wie, kiedy piosenka jest gotowa!

Strumieniowanie Wideo — Stream i MediaMTX (3 ścieżki)

Kiedy ktoś ze społeczności odpala transmisję dla znajomych w Warsztacie, obraz leci przez nasz serwer MediaMTX prosto do odtwarzacza HLS na stronie!

Endpoint	Metody	Dostęp	Opis działania i parametry
<code>/api/stream/key</code>	GET, POST, DELETE	Ciasteczko sesji Discord	Zarządzanie indywidualnym kluczem transmisji użytkownika do programu OBS (odczyt, wygenerowanie nowego klucza lub jego unieważnienie).
<code>/api/stream/queue</code>	GET, POST	Ciasteczko sesji Discord	Obsługuje kolejkę aktywnych streamerów oraz stan pokoju oglądania na żywo.

Endpoint	Metody	Dostęp	Opis działania i parametry
/api/stream/hls/[...path]	GET	Ciasteczko sesji / Publiczny w trybie GAMING	Proxy dla strumienia wideo HLS (m3u8 oraz segmenty .ts / .m4s) z serwera MediaMTX. Przekazuje ścieżkę [...] bez wystawiania wewnętrznych portów serwera na świat!

Na tej trzeciej stronie książki opisałam dla ciebie dokładnie **13 ścieżek** API (4 dla czatu i Kuźni, 6 dla muzyki oraz 3 dla streamu). *Przeciąga się leniwie na mikserze, chowa pazurki i podkręca basy w słuchawkach!*

Revision #1
Created 2026-10-12 21:38:40 UTC by Arikū
Updated 2026-10-12 21:38:40 UTC by Arikū