

Czas: ok. 10 minut. Potrzebny klucz API do OmniRoute ([jak zdobyć](#)) i Node.js ≥ 18 .

Instalacja

Linux i Windows (PowerShell):

```
npm install -g @openai/codex
```

Sprawdzenie:

```
codex --version
```

Oczekiwany wynik: `codex-cli <wersja>`.

Konfiguracja

1. Zmienna z kluczem

Codex czyta klucz ze zmiennej środowiskowej `OMNIRoute_API_KEY`.

Linux (dopisz do `~/.bashrc` albo `~/.zshrc`, potem otwórz nowy terminal):

```
export OMNIRoute_API_KEY="<TWÓJ_KLUCZ_OMNIRoute>"
```

Windows (PowerShell, zapis na stałe dla użytkownika, potem otwórz nowe okno):

```
[Environment]::SetEnvironmentVariable("OMNIRoute_API_KEY",  
"<TWÓJ_KLUCZ_OMNIRoute>", "User")
```

2. Plik `config.toml`

System	Plik
Linux	<code>~/.codex/config.toml</code>
Windows	<code>%USERPROFILE%\codex\config.toml</code>

Wklej poniższy wzór. Klucza nie wpisujesz do pliku: `env_key` wskazuje zmienną z kroku 1.

```
model          = "gpt-6.1-sol"
model_provider = "omniroute"

[model_providers.omniroute]
name          = "OmniRoute"
base_url      = "https://omni.suncode.dev/v1"
env_key       = "OMNIROUTE_API_KEY"
requires_openai_auth = false
wire_api      = "responses"
```

Klucz	Znaczenie
model_provider	Wskazuje sekcję [model_providers.omniroute].
base_url	Adres OmniRoute z /v1.
env_key	Nazwa zmiennej z kluczem, nie sam klucz.
requires_openai_auth = false	Codex nie wymaga logowania kontem OpenAI.
wire_api = "responses"	Protokół Responses API.

Sprawdzenie

```
codex exec "odpowiedz jednym sowem: dziaa"
```

Oczekiwany wynik: odpowiedź modelu bez błędu. Błąd: [Rozwiązywanie problemów](#).

Pamięć długoterminowa (Hindsight)

Codex korzysta z tego samego firmowego serwera pamięci co OMP i Claude Code: <https://hindsight.suncode.dev>. Łączy się przez oficjalny instalator Hindsight dla agentów ([@vectorize-io/hindsight-coding-agents](https://github.com/vectorize-io/hindsight-coding-agents)). Instalator dopisuje do Codexa hooki, które przypominają fakty z poprzednich sesji i zapisują nowe ustalenia, oraz serwer MCP hindsight.

Wymagany Node.js (ten sam co do instalacji Codexa).

1. Zainstaluj (Linux i Windows PowerShell, jedna linia):

```
npx @vectorize-io/hindsight-coding-agents install codex
--server self-hosted --api-url
https://hindsight.suncode.dev
```

Instalator zmienia `~/.codex/hooks.json` i `~/.codex/config.toml` (sekcja `[mcp_servers]`). Nie musisz ich edytować ręcznie.

2. Uruchom ponownie codex.

Pamięć jest osobna dla każdego repozytorium git (bank `coding-agent::<nazwa-repo>`). Na serwerze firmowym nie ustawiasz kluczy OpenAI/Anthropic: fakty z rozmów wyciąga serwer.

Sprawdzenie: `codex mcp list` pokazuje serwer `hindsight`. Potem w sesji napisz `zapamiętaj : test pamięci`, otwórz nową sesję w tym samym repo i zapytaj `co pamiętasz o tym projekcie?`.

Aktualizacja: `npx @vectorize-io/hindsight-coding-agents update`. Usunięcie: `npx @vectorize-io/hindsight-coding-agents uninstall` `codex`.

Nie każ agentowi zapamiętywać haseł, kluczy API ani danych klientów. Pamięć jest przechowywana na serwerze.