Mozout

Gerado a 28/11/2026 14:51

BASE DE CONHECIMENTO · DOCUMENTO PÚBLICO

Codex CLI com o All in One AI: criar o config.toml, a chave e o modelo

All in One AI

CÓDIGO
RLX-AO-KB-8904

VERSÃO
v2026.11.28

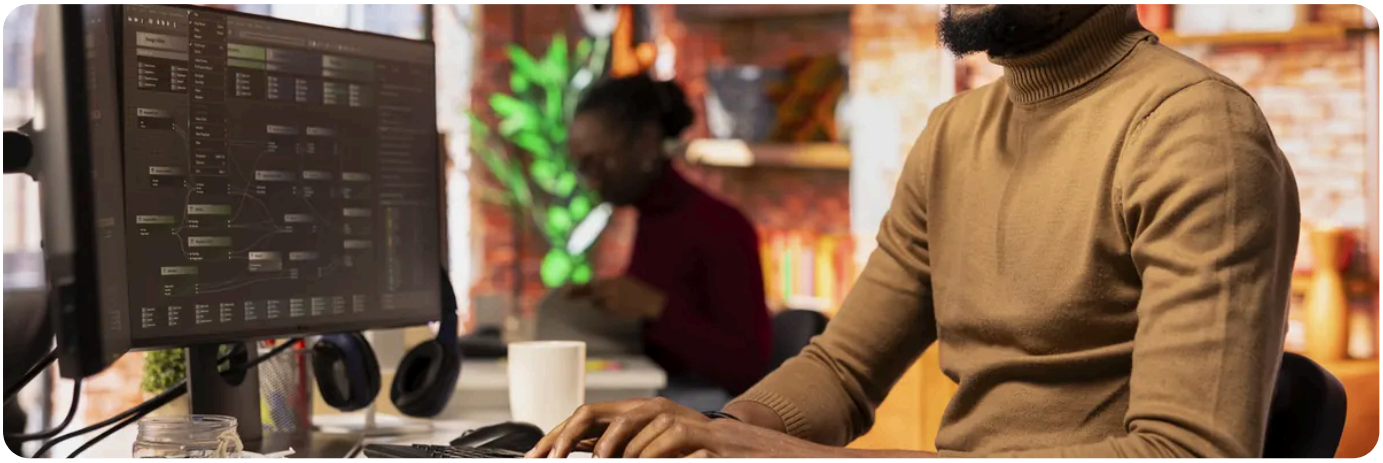
PAÍS · IDIOMA
Angola · Português

MozOut Angola
RLX Angola, Lda. · Grupo RLX Holdings
NIF 500 260 5724 · Matrícula 26464-25/250724

https://www.my.mozout.com/knowledgebase/8904

Codex CLI com o All in One AI: criar o config.toml, a chave e o modelo

All in One AI Criado em **28 de novembro de 2026** 4 min de leitura



O Codex CLI é o agente de programação de terminal da OpenAI. Fala o formato Responses, que a nossa API também aceita, por isso trabalha com os modelos do All in One AI. A configuração é um ficheiro TOML com um fornecedor próprio. Aqui está tudo, do zero.

Antes de começar

1. Crie uma chave no **API Center** com as permissões `ai:read` e `ai:write`. Use uma chave `sandbox_` para confirmar a configuração sem gastar e uma `prod_` para trabalhar. Os campos estão em **criar uma chave da API**.
2. Tenha créditos de IA ou um Coding Plan activo no **All in One AI**.
3. Instale o Codex CLI. Uma das formas que o README oficial indica: `npm install -g @openai/codex`.

1. Guardar a chave numa variável

No Windows, abra a Linha de Comandos ou o PowerShell e escreva (com a sua chave entre aspas):

```
setx MOZOUT_API_KEY "prod_..."
```

O `setx` grava a variável para as janelas que abrir **depois**. Feche o terminal e a ferramenta e volte a abri-los.

No Linux e no macOS, escreva no terminal:

```
export MOZOUT_API_KEY="prod_..."
```

Para a variável existir em cada terminal novo, junte a mesma linha ao fim do `~/.bashrc` ou do `~/.zshrc`.

2. Criar a pasta e o ficheiro

Sistema	Ficheiro
Linux e macOS	`~/.codex/config.toml`
Windows	`%USERPROFILE%\.codex\config.toml`

No Linux e no macOS:

```
mkdir -p ~/.codex
nano ~/.codex/config.toml
```

No Windows, no PowerShell:

```
New-Item -ItemType Directory -Force "$HOME\.codex"
notepad "$HOME\.codex\config.toml"
```

O Bloco de Notas pergunta se quer criar o ficheiro: responda que sim. Assim o nome fica exacto. Se usar **Guardar como**, escolha **Todos os ficheiros (*.*)** no tipo: senão, o Windows junta `.txt` e o ficheiro fica `config.toml.txt`, que a ferramenta não lê. O Windows esconde essa terminação. Para confirmar, escreva `dir` na mesma pasta: o nome tem de acabar em `.toml`.

3. Colar a configuração e guardar

```
model = "moonshotai/kimi-k2.7-code"
model_provider = "api"

[model_providers.api]
name = "MozOut"
base_url = "https://api.mozout.com/v1"
env_key = "MOZOUT_API_KEY"
wire_api = "responses"
```

Se o ficheiro já tiver outras definições, ponha as linhas `model` e `model_provider` no topo, antes do primeiro `[...]`. No TOML, o que vem depois de uma tabela fica dentro dela.

Campo	O que faz
`model`	O modelo de arranque, com o id do catálogo.
`model_provider`	O fornecedor a usar. Tem de ser igual ao nome em `[model_providers.api]`.
`base_url`	O nosso endereço, com `/v1`.
`env_key`	O **nome** da variável com a chave. O Codex lê-a ao arrancar.

Campo	O que faz
`wire_api`	`"responses"` é o único valor que o Codex aceita.

O id do fornecedor (`api`) pode ter outro nome, mas não `openai`, `ollama` nem `lmstudio`, que estão reservados. O fornecedor só se define neste ficheiro do utilizador: o `.codex/config.toml` de um projecto não o pode mudar.

4. Abrir o Codex e confirmar

```
codex
```

Abra um terminal novo, entre na pasta do projecto e escreva `codex`. Mande um «olá». Com uma chave `sandbox_`, a resposta é sempre «Resposta de teste (sandbox).». Depois troque a variável pela chave `prod_` e abra um terminal novo.

Outro modelo

O Codex usa o modelo escrito em `model`. Para usar outro só numa sessão: `codex -m zai-org/glm-5.2`. Para mudar de vez, troque o `model` no ficheiro. Use o id do catálogo tal como está, e um modelo que aceite ferramentas:

Modelo	Contexto	Para quê
`moonshotai/kimi-k2.7-code`	262 144 tokens	Feito para programar: o ponto de partida para o dia-a-dia.
`deepseek-ai/deepseek-v4-pro`	1 048 576 tokens	Raciocínio forte, para mudanças grandes e erros difíceis.
`zai-org/glm-5.2`	1 048 576 tokens	Pensado para agentes: usa bem as ferramentas em tarefas longas.
`deepseek-ai/deepseek-v4-flash`	1 048 576 tokens	Rápido e barato, para tarefas de fundo e perguntas curtas.

Os ids estão na aba «Modelos» do **All in One AI** ou na lista da API. No Linux e no macOS:

```
curl "https://api.mozout.com/v1/models?type=text" -H "Authorization: Bearer $MOZOUT_API_KEY"
```

No PowerShell do Windows:

```
curl.exe "https://api.mozout.com/v1/models?type=text" -H "Authorization: Bearer $env:MOZOUT_API_KEY"
```

Cada modelo vem com o `id` (sempre com a barra), o `context_length` e o preço em `pricing`. Ver a lista não gasta créditos.

Cada resposta fica guardada 30 dias, para o Codex continuar a conversa. Para juntar imagens, vídeos e voz ao Codex, veja **o servidor MCP do All in One AI**. Se algo falhar, veja **os erros mais comuns**.

RECOMENDADO

Quer programar com IA no Codex? Veja a página Programar com IA.

Ver Programar com IA →

VEJA TAMBÉM

- **Erros ao associar uma ferramenta ao All in One AI: 401, modelo não encontrado, sem créditos, o /v1 e ficheiros recusados**
- **O servidor MCP do All in One AI: imagens, vídeos e voz no Claude Code, no Codex e no ChatGPT**
- **Que ferramenta de programação usar com o All in One AI: OpenCode, Claude Code, Codex, Cursor, Cline, Kilo Code ou Aider**
- **Coding Plan: qual escolher, quanto custa e como assinar**
- **Programar com IA no All in One AI**

Mozout

MozOut Angola

RLX Angola, Lda. · Grupo RLX Holdings

NIF	500 260 5724
MATRÍCULA	26464-25/250724
SEDE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACTOS	info@mozout.com · +244 942 011 191 · www.mozout.com
HORÁRIO	2ª. à 6ª. Feira - 08 à 16 Horas Sábado & Domingo - Fechado

Versão online deste documento

RLX-AO-KB-8904 · v2026.11.28 · Gerado a 28/11/2026 14:51