interweb

Generated on 28/11/2026 14:46

KNOWLEDGE BASE · PUBLIC DOCUMENT

Codex CLI with All in One AI: creating config.toml, the key and the model

All in One AI

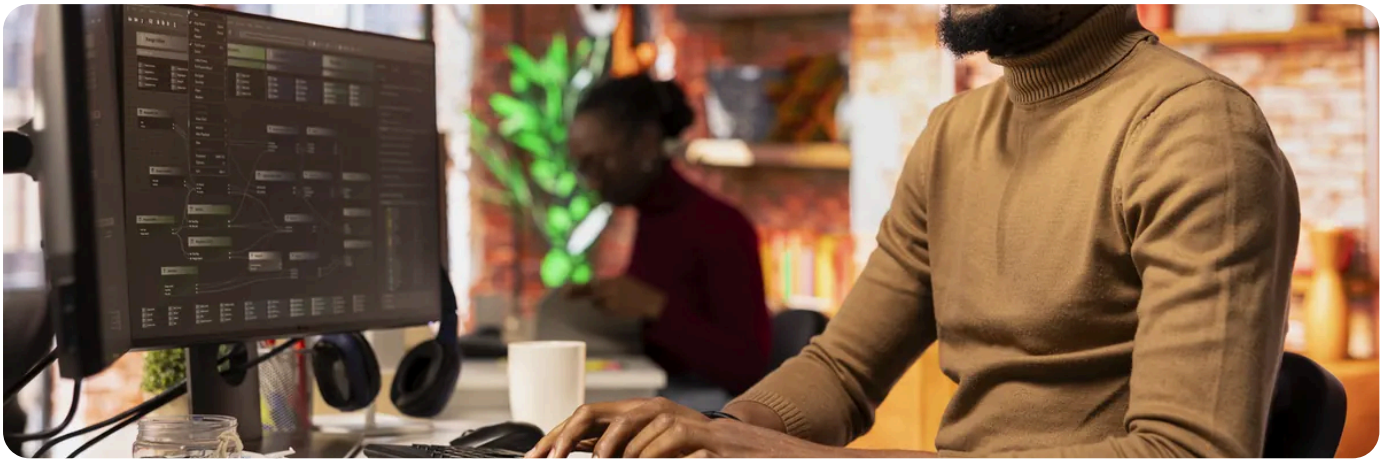
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-2940	v2026.11.28	Angola · English

Interweb Angola
RLX Angola, Lda. · Group RLX Holdings
Tax ID (NIF) 500 260 5724 · Commercial registration no. 26464-25/250724

https://www.meu.interweb.ao/knowledgebase/2940

Codex CLI with All in One AI: creating config.toml, the key and the model

All in One AI Created on **November 28, 2026** 4 min read



Codex CLI is OpenAI's terminal coding agent. It speaks the Responses format, which our API also accepts, so it works with All in One AI models. Its configuration is a TOML file with your own provider. This article creates it from scratch.

Before you start

1. Create a key in the **API Center** with the `ai:read` and `ai:write` permissions. Use a `sandbox_` key to check the setup without spending and a `prod_` key to work. The fields are explained in **creating an API key**.
2. Have AI credits or an active Coding Plan in **All in One AI**.
3. Install Codex CLI. One way from the official README: `npm install -g @openai/codex`.

1. Keep the key in a variable

On Windows, open Command Prompt or PowerShell and type (with your key in quotes):

```
setx INTERWEB_API_KEY "prod_..."
```

`setx` saves the variable for the windows you open **afterwards**. Close the terminal and the tool, then open them again.

On Linux and macOS, type in the terminal:

```
export INTERWEB_API_KEY="prod_..."
```

To have the variable in every new terminal, add the same line at the end of `~/.bashrc` or `~/.zshrc`.

2. Create the folder and the file

System	File
Linux and macOS	`~/.codex/config.toml`
Windows	`%USERPROFILE%\.codex\config.toml`

On Linux and macOS:

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

On Windows, in PowerShell:

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

Notepad asks whether to create the file: answer yes. That way the name is exact. If you use **Save as**, choose **All files (*.*)** as the type: otherwise Windows adds `.txt` and the file becomes `config.toml.txt`, which the tool does not read. Windows hides that ending. To check, type `dir` in the same folder: the name must end in `.toml`.

3. Paste the configuration and save

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

[model_providers.api]
name = "Interweb"
base_url = "https://api.interweb.ao/v1"
env_key = "INTERWEB_API_KEY"
wire_api = "responses"
```

If the file already had other settings, put the `model` and `model_provider` lines at the top, before the first `[...]`. In TOML, whatever comes after a table goes inside it.

Field	What it does
`model`	The starting model, with the catalogue id.
`model_provider`	The provider to use. It must match the name in `[model_providers.api]`.
`base_url`	Our address, with `/v1`.
`env_key`	The **name** of the variable that holds the key. Codex reads it at startup.

Field	What it does
`wire_api`	`"responses"` is the only value Codex accepts.

The provider id (`api`) can have another name, but not `openai` , `ollama` or `lmstudio` , which are reserved. The provider is only set in this user file: a project's `.codex/config.toml` cannot change it.

4. Open Codex and check

```
codex
```

Open a new terminal, go to the project folder and type `codex` . Say "hello". With a `sandbox_` key, the answer is always "Resposta de teste (sandbox).". Then set the variable to the `prod_` key and open a new terminal.

Another model

Codex uses the model written in `model` . For another one in a single session: `codex -m zai-org/glm-5.2` . To change it for good, edit `model` in the file. Use the catalogue id exactly as it is, and a model that supports tools:

Model	Context	What for
`moonshotai/kimi-k2.7-code`	262,144 tokens	Built for coding: the starting point for everyday work.
`deepseek-ai/deepseek-v4-pro`	1,048,576 tokens	Strong reasoning, for big changes and hard bugs.
`zai-org/glm-5.2`	1,048,576 tokens	Made for agents: handles tools well in long tasks.
`deepseek-ai/deepseek-v4-flash`	1,048,576 tokens	Fast and cheap, for background tasks and short questions.

The ids are in the Models tab of **All in One AI** or in the API list. On Linux and macOS:

```
curl "https://api.interweb.ao/v1/models?type=text" -H "Authorization: Bearer $INTERWEB_API_KEY"
```

In Windows PowerShell:

```
curl.exe "https://api.interweb.ao/v1/models?type=text" -H "Authorization: Bearer $env:INTERWEB_API_KEY"
```

Each model comes with its `id` (always with the slash), its `context_length` and its price in `pricing` . Listing the models spends no credits.

Each response is kept for 30 days, so Codex can carry on the conversation. To add images, video and voice to Codex, see **the All in One AI MCP server**. If something fails, see **the most common errors**.

RECOMMENDED

Want to code with AI in Codex? Visit the Coding with AI page.

See Coding with AI →

SEE ALSO

- **Errors when setting up a tool with All in One AI: 401, model not found, no credits, /v1 and rejected config files**
- **The All in One AI MCP server: images, video and voice inside Claude Code, Codex and ChatGPT**
- **Which coding tool to use with All in One AI: OpenCode, Claude Code, Codex, Cursor, Cline, Kilo Code or Aider**
- **Coding Plan: which plan to choose, what it costs and how to subscribe**
- **Coding with AI in All in One AI**

interweb

Interweb Angola

RLX Angola, Lda. · Group RLX Holdings

TAX ID (NIF)	500 260 5724
COMMERCIAL REGISTRATION NO.	26464-25/250724
REGISTERED OFFICE	Rua Direita do Patriota, Bairro Dangereux Belas, Luanda, Angola
CONTACT	info@interweb.ao · +244 942 011 191 · www.interweb.ao
OPENING HOURS	Monday to Friday - 08 AM to 04 PM Saturday & Sunday - Closed

Online version of this document

RLX-AO-KB-2940 · v2026.11.28 · Generated on 28/11/2026 14:46