interweb

Generated on 28/11/2026 14:46

KNOWLEDGE BASE · PUBLIC DOCUMENT

Kilo Code with All in One AI: the custom provider in VS Code and kilo.jsonc in the CLI

All in One AI

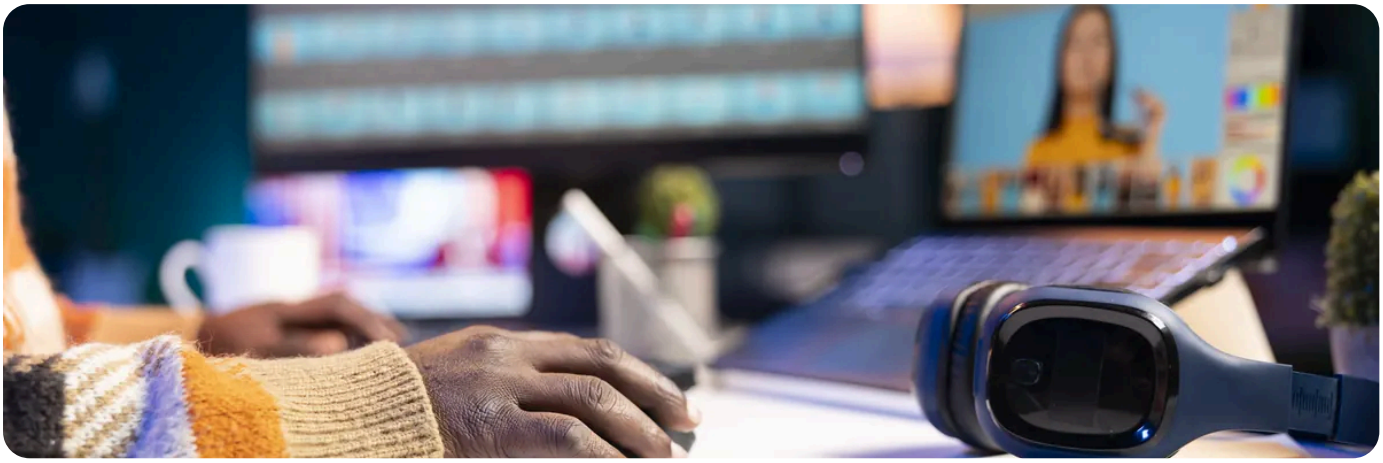
CODE	VERSION	COUNTRY · LANGUAGE
RLX-AO-KB-2946	v2026.11.28	Angola · English

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/2946

Kilo Code with All in One AI: the custom provider in VS Code and kilo.jsonc in the CLI

All in One AI Created on **November 28, 2026** 9 min read



Kilo Code comes as a VS Code extension and as a command-line program. Both accept your own OpenAI-compatible provider, so they use All in One AI models with the Interweb key. In the extension it is a form. On the command line it is a file, which this article creates 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**.

In the VS Code extension

1. Open the settings (the gear icon) and the **Providers** tab.
2. Scroll to the end of the list and click **Custom provider**.
3. Fill in **Provider ID** (for example `api`), **Display name** (Interweb), **Provider API** (**OpenAI Compatible**), **Base URL** (`https://api.interweb.ao/v1`) and **API key**.
4. Under **Models**, choose the models. Kilo Code reads the list from our catalogue, at `GET /v1/models`. Click **Submit**.

Here you do not need to type the ids: they come from the list. Choose models that support tools, such as `moonshotai/kimi-k2.7-code` or `zai-org/glm-5.2`.

On the command line: 1. Install it and keep the key

Install the program with `npm install -g @kilocode/cli`. Then 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	`~/.config/kilo/kilo.jsonc`
Windows	`C:\Users\<your name>\.config\kilo\kilo.jsonc`

On Linux and macOS:

```
mkdir -p ~/.config/kilo
nano ~/.config/kilo/kilo.jsonc
```

On Windows, in PowerShell:

```
New-Item -ItemType Directory -Force "$HOME\.config\kilo"
notepad "$HOME\.config\kilo\kilo.jsonc"
```

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 `kilo.jsonc.txt`, which the tool does not read. Windows hides that ending. To check, type `dir` in the same folder: the name must end in `.jsonc`.

3. Paste the configuration and save

```
{
  "provider": {
    "api": {
      "npm": "@ai-sdk/openai-compatible",
      "name": "Interweb",
      "options": {
        "baseURL": "https://api.interweb.ao/v1",
```

```
        "apiKey": "{env:INTERWEB_API_KEY}"
      },
      "models": {
         "anthropic/claude-fable-5": { "name": "Claude Fable 5", "limit": { "context":
1000000, "output": 16000 } },
            "anthropic/claude-fable-5.1": { "name": "Claude Fable 5.1", "limit": {
"context": 1000000, "output": 16000 } },
            "openai/gpt-6-astra-codex": { "name": "GPT 6 Astra Codex", "limit": {
"context": 1050000, "output": 16000 } },
         "anthropic/claude-fable-5.2-ccmax": { "name": "Claude Fable 5.2", "limit": {
"context": 1000000, "output": 16000 } },
           "openai/gpt-6.2-astra": { "name": "GPT 6.2 Astra", "limit": { "context":
1050000, "output": 16000 } },
          "openai/gpt-6.2-astra-codex": { "name": "GPT 6.2 Astra Codex", "limit": {
"context": 1050000, "output": 16000 } },
        "openai/gpt-5.5": { "name": "GPT 5.5", "limit": { "context": 1050000, "output":
16000 } },
           "anthropic/claude-opus-4.5-20251101-coding": { "name": "Claude Opus 4.5",
"limit": { "context": 200000, "output": 16000 } },
        "anthropic/claude-opus-4.6": { "name": "Claude Opus 4.6", "limit": { "context":
1000000, "output": 16000 } },
        "anthropic/claude-opus-4.7": { "name": "Claude Opus 4.7", "limit": { "context":
1000000, "output": 16000 } },
        "anthropic/claude-opus-4.8": { "name": "Claude Opus 4.8", "limit": { "context":
1000000, "output": 16000 } },
         "anthropic/claude-opus-5": { "name": "Claude Opus 5", "limit": { "context":
1000000, "output": 16000 } },
        "anthropic/claude-opus-5.5": { "name": "Claude Opus 5.5", "limit": { "context":
1000000, "output": 16000 } },
          "moonshotai/kimi-k3.1-pro-highspeed": { "name": "Kimi K3.1 Pro HighSpeed",
"limit": { "context": 1048576, "output": 16000 } },
            "openai/gpt-5.6-sol-codex": { "name": "GPT 5.6 Sol Codex", "limit": {
"context": 1050000, "output": 16000 } },
           "anthropic/claude-sonnet-4.6": { "name": "Claude Sonnet 4.6", "limit": {
"context": 200000, "output": 16000 } },
          "moonshotai/kimi-k3": { "name": "Kimi K3", "limit": { "context": 1048576,
"output": 16000 } },
        "google/gemini-3.1-pro-preview": { "name": "Gemini 3.1 Pro Preview", "limit": {
"context": 1000000, "output": 16000 } },
        "anthropic/claude-sonnet-5": { "name": "Claude Sonnet 5", "limit": { "context":
1000000, "output": 16000 } },
         "anthropic/claude-sonnet-5.5-aws": { "name": "Claude Sonnet 5.5", "limit": {
"context": 1000000, "output": 16000 } },
         "moonshotai/kimi-k3.1-pro": { "name": "Kimi K3.1 Pro", "limit": { "context":
1048576, "output": 16000 } },
```

```
          "openai/gpt-6-sol-codex": { "name": "GPT 6 Sol Codex", "limit": { "context":
1050000, "output": 16000 } },
             "openai/gpt-6.1-sol-codex": { "name": "GPT 6.1 Sol Codex", "limit": {
"context": 1050000, "output": 16000 } },
             "openai/gpt-6.2-sol-codex": { "name": "GPT 6.2 Sol Codex", "limit": {
"context": 1050000, "output": 16000 } },
           "google/gemini-2.5-pro": { "name": "Gemini 2.5 Pro", "limit": { "context":
1000000, "output": 16000 } },
         "google/gemini-3.5-flash": { "name": "Gemini 3.5 Flash", "limit": { "context":
1048576, "output": 16000 } },
           "qwen/qwen3.7-max": { "name": "Qwen3.7 Max", "limit": { "context": 1000000,
"output": 16000 } },
           "google/gemini-4-argon": { "name": "Gemini 4 Argon", "limit": { "context":
1048576, "output": 16000 } },
         "google/gemini-3.6-flash": { "name": "Gemini 3.6 Flash", "limit": { "context":
1048576, "output": 16000 } },
         "xai/grok-4.5": { "name": "Grok 4.5", "limit": { "context": 500000, "output":
16000 } },
         "xai/grok-4.6": { "name": "Grok 4.6", "limit": { "context": 500000, "output":
16000 } },
         "xai/grok-4.7": { "name": "Grok 4.7", "limit": { "context": 500000, "output":
16000 } },
           "google/gemini-4-carbon-developer": { "name": "Gemini 4 Carbon", "limit": {
"context": 1048576, "output": 16000 } },
        "anthropic/claude-haiku-4.5-20251001": { "name": "Claude Haiku 4.5", "limit": {
"context": 200000, "output": 16000 } },
           "openai/gpt-6.1-terra-codex": { "name": "GPT 6.1 Terra Codex", "limit": {
"context": 1050000, "output": 16000 } },
             "zai-org/glm-5.3": { "name": "GLM 5.3", "limit": { "context": 1048576,
"output": 16000 } },
           "qwen/qwen3.8-max": { "name": "Qwen3.8 Max", "limit": { "context": 1000000,
"output": 16000 } },
             "bytedance/doubao-seed-2.1-pro-260628": { "name": "Doubao Seed 2.1 Pro",
"limit": { "context": 262144, "output": 16000 } },
        "zai-org/glm-5.1": { "name": "GLM 5.1", "limit": { "context": 202752, "output":
16000 } },
             "deepseek-ai/deepseek-v4-pro": { "name": "DeepSeek V4 Pro", "limit": {
"context": 1048576, "output": 16000 } },
           "moonshotai/kimi-k2.6": { "name": "Kimi K2.6", "limit": { "context": 262144,
"output": 16000 } },
         "moonshotai/kimi-k2.7-code": { "name": "Kimi K2.7 Code", "limit": { "context":
262144, "output": 16000 } },
         "google/gemini-3.7-flash": { "name": "Gemini 3.7 Flash", "limit": { "context":
1048576, "output": 16000 } },
         "google/gemini-3.8-flash": { "name": "Gemini 3.8 Flash", "limit": { "context":
```

```
1048576, "output": 16000 } },
          "zai-org/glm-5": { "name": "GLM 5", "limit": { "context": 202752, "output":
16000 } },
         "qwen/qwen3.5-397b-a17b": { "name": "Qwen3.5 397BA17B", "limit": { "context":
262144, "output": 16000 } },
            "zai-org/glm-5.2": { "name": "GLM 5.2", "limit": { "context": 1048576,
"output": 16000 } },
        "xai/grok-4.3": { "name": "Grok 4.3", "limit": { "context": 1000000, "output":
16000 } },
           "bytedance/doubao-seed-2.0-pro-260215": { "name": "Doubao Seed 2.0 Pro",
"limit": { "context": 262144, "output": 16000 } },
        "google/gemini-3-flash-preview": { "name": "Gemini 3 Flash Preview", "limit": {
"context": 200000, "output": 16000 } },
          "qwen/qwen3.8-27b": { "name": "Qwen3.8 27B", "limit": { "context": 1048576,
"output": 16000 } },
          "tencent/hy4-preview": { "name": "Hunyuan 4 Preview", "limit": { "context":
1048576, "output": 16000 } },
          "moonshotai/kimi-k2.5": { "name": "Kimi K2.5", "limit": { "context": 262144,
"output": 16000 } },
        "google/gemini-2.5-flash": { "name": "Gemini 2.5 Flash", "limit": { "context":
1048576, "output": 16000 } },
         "google/gemini-3.5-flash-lite": { "name": "Gemini 3.5 Flash Lite", "limit": {
"context": 1048576, "output": 16000 } },
         "qwen/qwen3.5-plus": { "name": "Qwen3.5 Plus", "limit": { "context": 1000000,
"output": 16000 } },
          "bytedance/doubao-seed-2.1-turbo-260628": { "name": "Doubao Seed 2.1 Turbo",
"limit": { "context": 262144, "output": 16000 } },
        "qwen/qwen3.5-122b-a10b": { "name": "Qwen3.5 122B A10B", "limit": { "context":
262144, "output": 16000 } },
          "qwen/qwen3.5-27b": { "name": "Qwen3.5 27B", "limit": { "context": 262144,
"output": 16000 } },
         "qwen/qwen3.6-plus": { "name": "Qwen3.6 Plus", "limit": { "context": 1000000,
"output": 16000 } },
           "bytedance/doubao-seed-2.0-lite-260428": { "name": "Doubao Seed 2.0 Lite",
"limit": { "context": 262144, "output": 16000 } },
         "qwen/qwen3.7-plus": { "name": "Qwen3.7 Plus", "limit": { "context": 1000000,
"output": 16000 } },
         "google/gemini-3.1-flash-lite": { "name": "Gemini 3.1 Flash Lite", "limit": {
"context": 1048576, "output": 16000 } },
        "deepseek-ai/deepseek-v4.1-flash": { "name": "DeepSeek V4.1 Flash", "limit": {
"context": 1048576, "output": 16000 } },
         "meituan-longcat/longcat-2.0": { "name": "LongCat 2.0", "limit": { "context":
1048756, "output": 16000 } },
           "xiaomi/mimo-v2.5-pro": { "name": "MiMo V2.5 Pro", "limit": { "context":
1024000, "output": 16000 } },
```

```
          "qwen/qwen3.6-35b-a3b": { "name": "Qwen3.6 35B A3B", "limit": { "context":
262144, "output": 16000 } },
          "bytedance/doubao-seed-character-260628": { "name": "Doubao Seed Character",
"limit": { "context": 131072, "output": 16000 } },
         "tencent/hy3": { "name": "Hunyuan 3", "limit": { "context": 262144, "output":
16000 } },
          "zai-org/glm-5.3-flash": { "name": "GLM 5.3 Flash", "limit": { "context":
1048576, "output": 16000 } },
          "deepseek-ai/DeepSeek-V3.2-Exp": { "name": "DeepSeek V3.2 Exp", "limit": {
"context": 163840, "output": 16000 } },
         "deepseek-ai/deepseek-v3.2": { "name": "DeepSeek V3.2", "limit": { "context":
163840, "output": 16000 } },
          "anthropic/claude-haiku-5.5-ccmax": { "name": "Claude Haiku 5.5", "limit": {
"context": 1000000, "output": 16000 } },
        "openai/gpt-6-luna-codex": { "name": "GPT 6 Luna Codex", "limit": { "context":
1050000, "output": 16000 } },
          "bytedance/doubao-seed-2.0-mini-260428": { "name": "Doubao Seed 2.0 Mini",
"limit": { "context": 262144, "output": 16000 } },
         "google/gemini-2.5-flash-lite": { "name": "Gemini 2.5 Flash Lite", "limit": {
"context": 1048576, "output": 16000 } },
        "qwen/qwen3.5-flash": { "name": "Qwen3.5 Flash", "limit": { "context": 1000000,
"output": 16000 } },
          "deepseek-ai/deepseek-v4-flash": { "name": "DeepSeek V4 Flash", "limit": {
"context": 1048576, "output": 16000 } },
         "dots-studio/dots-3-note-prev-free": { "name": "Dots 3 Note Preview", "limit":
{ "context": 512000, "output": 16000 } }
      }
    }
  },
  "model": "api/moonshotai/kimi-k2.7-code"
}
```

`{env:INTERWEB_API_KEY}` reads the key from the variable and only works in this global file, in `~/.config/kilo/`. A project's `kilo.jsonc` does not read environment variables, so a repository cannot read your key. We reserve credits for the maximum output the tool asks for in each answer. That is why the examples ask for 16,000 tokens: it is enough for coding and needs a smaller balance than a higher value.

There must be at least one model in `models`, and only the ones listed there show. To add one, add a line with the catalogue id and put a comma at the end of the line before it.

4. Open it and check

Close Kilo Code if it is open and start it again: it only reads the file when it starts. Type `/models` and check that the Interweb provider's models show. Say "hello". With a `sandbox_` key, the answer is always "Resposta de teste (sandbox).".

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.

If you get an error, see **errors when setting up a tool with All in One AI**. The file format is the same as OpenCode's: **the OpenCode article** explains each part.

RECOMMENDED

Want our models in Kilo Code? Visit the Coding with AI page.

See Coding with AI →

SEE ALSO

- **OpenCode with All in One AI: creating opencode.json from scratch, the key and the models**
- **Cline with All in One AI: the OpenAI Compatible provider, field by field**
- **Errors when setting up a tool with All in One AI: 401, model not found, no credits, /v1 and rejected config files**
- **Which coding tool to use with All in One AI: OpenCode, Claude Code, Codex, Cursor, Cline, Kilo Code or Aider**
- **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-2946 · v2026.11.28 · Generated on 28/11/2026 14:46