Mozout

Generated on 29/11/2026 04:04

KNOWLEDGE BASE · PUBLIC DOCUMENT

# Why the key only works in the global kilo.jsonc, and not in a project's file

Kilo Code

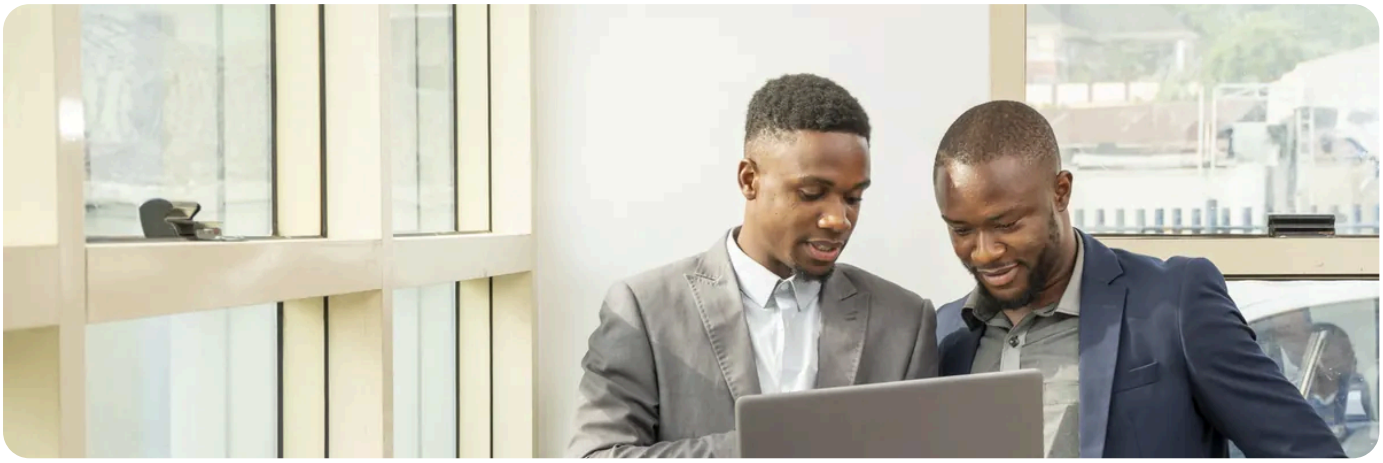
| CODE | VERSION | COUNTRY · LANGUAGE |
| --- | --- | --- |
| RLX-INT-KB-9696 | v2026.11.28 | International · English |

**MozOut International**
RLX Holdings (Pty) Ltd. · Group RLX Holdings
Tax no. (SARS) 943 907 6234 · Registration no. (CIPC) 2025/667953/07

https://www.my.mozout.com/knowledgebase/9696

# Why the key only works in the global kilo.jsonc, and not in a project's file

Kilo Code Created on **November 28, 2026** 2 min read



A common case: you put the MozOut provider in the project's `kilo.jsonc`, with `"apiKey": "{env:MOZOUT_API_KEY}"`, and Kilo Code behaves as if there were no key. The setup is not wrong. Kilo Code does not read variables in project files, on purpose.

## The rule

Kilo Code only replaces `{env:...}` in configuration it trusts: the global file (in `~/.config/kilo/`), a file given in the `KILO_CONFIG` or `KILO_CONFIG_CONTENT` variables, and configuration managed by your organisation. In a project's `kilo.jsonc` or `opencode.json`, the reference is ignored and Kilo Code writes a warning in its log.

The reason: a project file can come from any repository. This way, a repository you cloned cannot read your key.

## What happens in practice

The provider has no key and the requests fail with `401`. Kilo Code may show "Invalid API Key".

## The fix

1. Move the MozOut provider block (with `options` and `models`) to the global file: `~/.config/kilo/kilo.jsonc`, or `C:\Users\<your name>\.config\kilo\kilo.jsonc` on Windows.

2. In the project file, leave only what does not need the key, such as the model: `"model": "api/zai-org/glm-5.2"`. Project settings win over the global ones.

3. Check that the `MOZOUT_API_KEY` variable exists (`setx` on Windows, `export` on Linux and macOS) and open a new terminal.

4 Close Kilo Code and start it again.

The full provider block is in the article on **the global kilo.jsonc file**.

## What about writing the key in the project file?

Do not do it. The Kilo Code documentation itself warns that these files must not hold keys (`provider.*.options.apiKey`), because they usually go into the repository, and tells you to use environment variables.

Kilo Code accepts `{file:...}` in a project file, but only for files inside the project folder, which is the one that goes to Git. So it is no good for the key either.

OpenCode is different: its project file reads variables and can go into the repository. See **sharing an OpenCode setup without sharing the key**.

RECOMMENDED

Want Kilo Code with your key kept safe? Visit the Coding with AI page.

**See Coding with AI →**

SEE ALSO

- **Kilo Code with All in One AI: the custom provider in VS Code and kilo.jsonc in the CLI**
- **Kilo Code on the command line: the global kilo.jsonc file**
- **An API key or token has leaked: what to do, in the right order**
- **Sharing an OpenCode setup in your repository without sharing the key**
- **Coding with AI in All in One AI**

Mozout

## MozOut International

**RLX Holdings (Pty) Ltd. · Group RLX Holdings**


| | |
|---|---|
| TAX NO. (SARS) | 943 907 6234 |
| REGISTRATION NO. (CIPC) | 2025/667953/07 |
| REGISTERED OFFICE | 1111 Olive Street<br>Saint Louis 63101, Missouri, United States |
| CONTACT | info@mozout.com · +27 78 823 3325 · www.mozout.com |
| OPENING HOURS | Monday to Friday - 08 AM to 04 PM<br>Saturday & Sunday - Closed |


Online version of this document

RLX-INT-KB-9696 · v2026.11.28 · Generated on 29/11/2026 04:04