Selecting .venv/bin/python as an active interpreter in VSCode doesn't enable IDE features #9637

Open

oulianov opened on Dec 4, 2024 · edited by oulianov

Edits ···

Maintainer edit: Please set `python.usePythonEnvironmentsExtension: true` and check if the problem persists. See #9637 (comment) for more detail.

Hello,

I develop my Python app with uv to handle dependencies. I want to benefit from intellisense, type hints, mypy and ruff in VSCode while I develop my app. VSCode is the most popular IDE for Python developers.

Usually, I create the a virtual env with `python -m venv`. I then select the virtual env in VSCode as an "Active Interpreter". This adds support from all the VSCode features mentioned.

With uv, a folder .venv is created. However, when I select this .venv as an active interpreter in VSCode, **nothing changes in my editor.** Intellisense, type hints, and mypy don't work. Installed modules are not found. Custom modules are not found either.

When I run "Reload Window", this results in a toast on the bottom right "Reactivating terminals" that never completes.

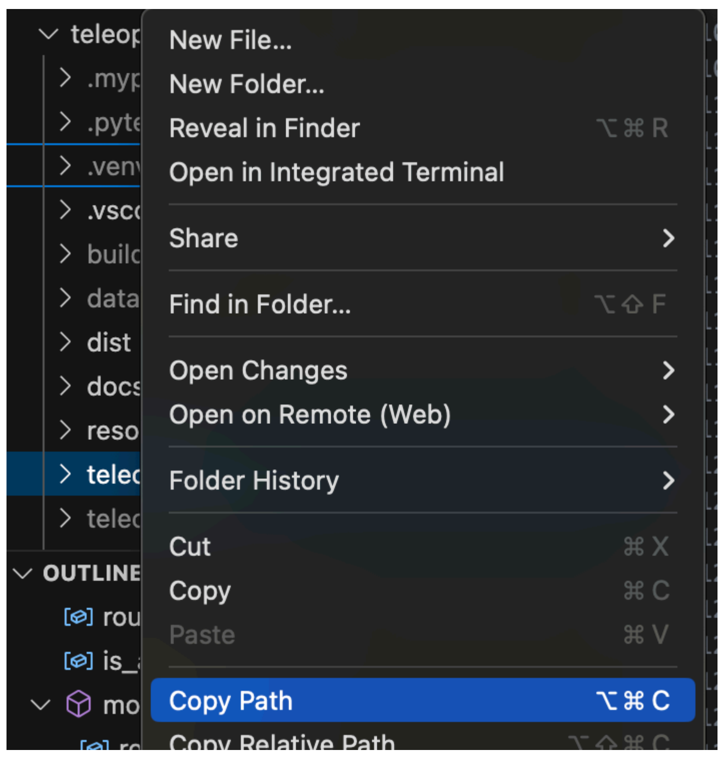
Restarting VSCode with the config doesn't make it work.

This means for now I need to have multiple virtual environment running. One for uv, another for development. This voids the usefulness of uv for package management.

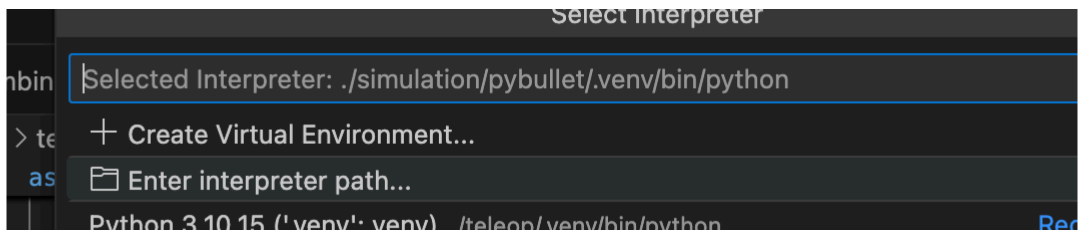
How to make uv virtual environments work with VSCode?

[EDIT] HERE IS A WORKAROUND! Here's a foolproof technique my buddy **@Jeerhz** showed me to select a uv `.venv` as an active interpreter in vscode.

1. First, copy the absolute path to the `.venv`. **Not** to the /bin/python, only the .venv folder.



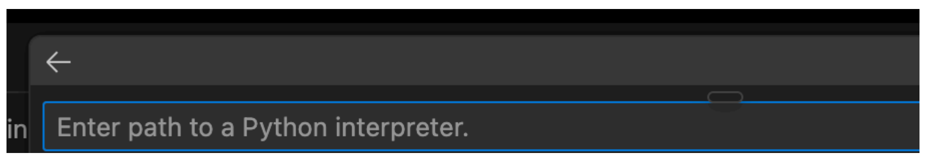
2. Do Cmd+Shift+P, then look for "Python: Select interpreter". Press Enter. Select the option: "Enter interpreter path".



Selected Interpreter: ./simulation/pybullet/.venv/bin/python

Python 3.10.15 ('.venv': venv) ./teleop/.venv/bin/python

3. Now in this field you copy paste the absolute path to the .venv folder.



Enter path to a Python interpreter.

Open **Selecting .venv/bin/python as an active interpreter in VSCode doesn't enable IDE features** #9637

Press enter and now you have the right Python environment with all the dependencies.

Selecting .venv as an active interpreter in VSCode works. But selecting .venv/bin/python doesn't.

Recommendation: use an extension like teticio.python-envy to automatically set the interpreter path based on the file you're currently looking at, compare #9637 (comment) . This is what I currently use and it has been working quite well in my monorepo setup.

👍 96 ❤️ 14 🚀 15

oulianov changed the title ~~Selecting .venv/bin/python as an active interpreter in VSCode doesn't enable features~~ Selecting .venv/bin/python as an active interpreter in VSCode doesn't enable IDE features on Dec 4, 2024

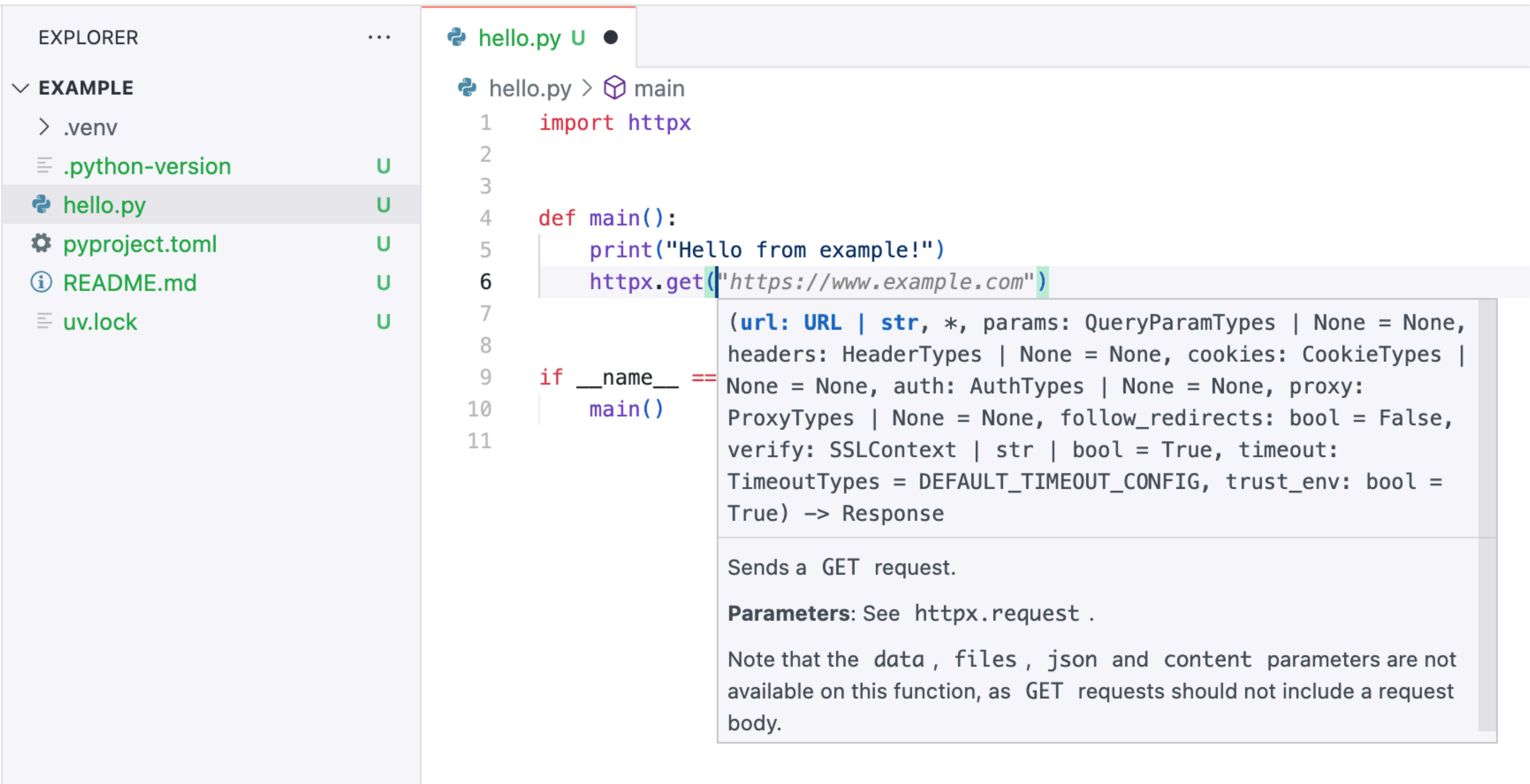
zanieb on Dec 4, 2024

Member ···

I don't know what to say here, this is pretty basic functionality and I cannot reproduce it

```
❯ uv init example
Initialized project `example` at `/Users/zb/workspace/uv/example`
❯ code example
❯ cd example
❯ uv add httpx
Using CPython 3.12.7
Creating virtual environment at: .venv
Resolved 8 packages in 113ms
```

```
Installed 7 packages in 13ms
 + anyio==4.6.2.post1
 + certifi==2024.8.30
 + h11==0.14.0
 + httpcore==1.0.7
 + httpx==0.28.0
 + idna==3.10
 + sniffio==1.3.1
```



```python
import httpx


def main():
    print("Hello from example!")
    httpx.get("https://www.example.com")


if __name__ == 
    main()

```

```
(url: URL | str, *, params: QueryParamTypes | None = None,
headers: HeaderTypes | None = None, cookies: CookieTypes |
None = None, auth: AuthTypes | None = None, proxy:
ProxyTypes | None = None, follow_redirects: bool = False,
verify: SSLContext | str | bool = True, timeout:
TimeoutTypes = DEFAULT_TIMEOUT_CONFIG, trust_env: bool =
True) -> Response
```

Sends a `GET` request.

Parameters: See `httpx.request` .

Note that the *`data`* , `files` , `json` and `content` parameters are not available on this function, as `GET` requests should not include a request body.

Can you share more details?

zanieb added **needs-mre** on Dec 4, 2024

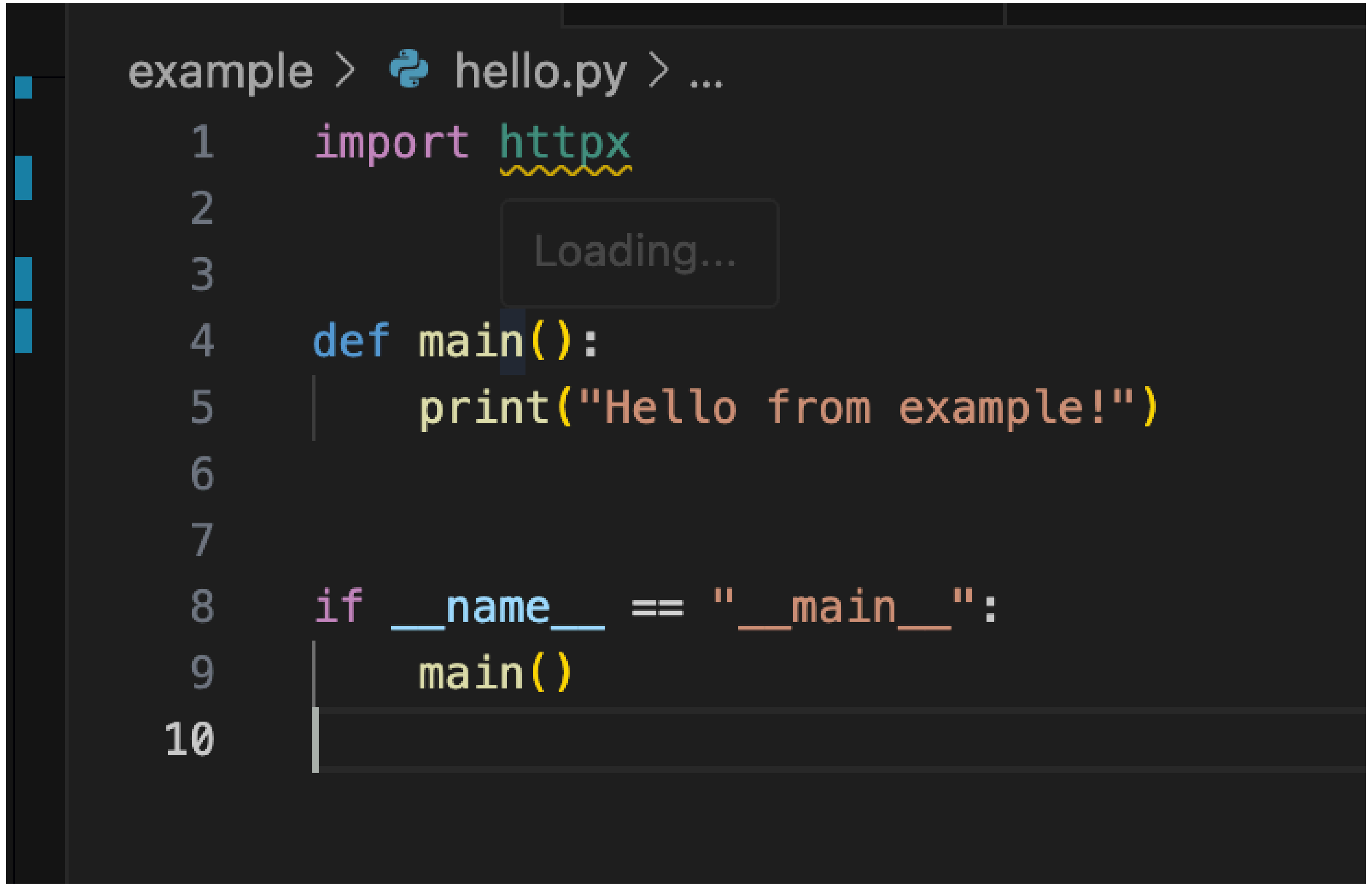
oulianov on Dec 5, 2024 · edited by oulianov

Edits Author ···

Hello, Zanieb!

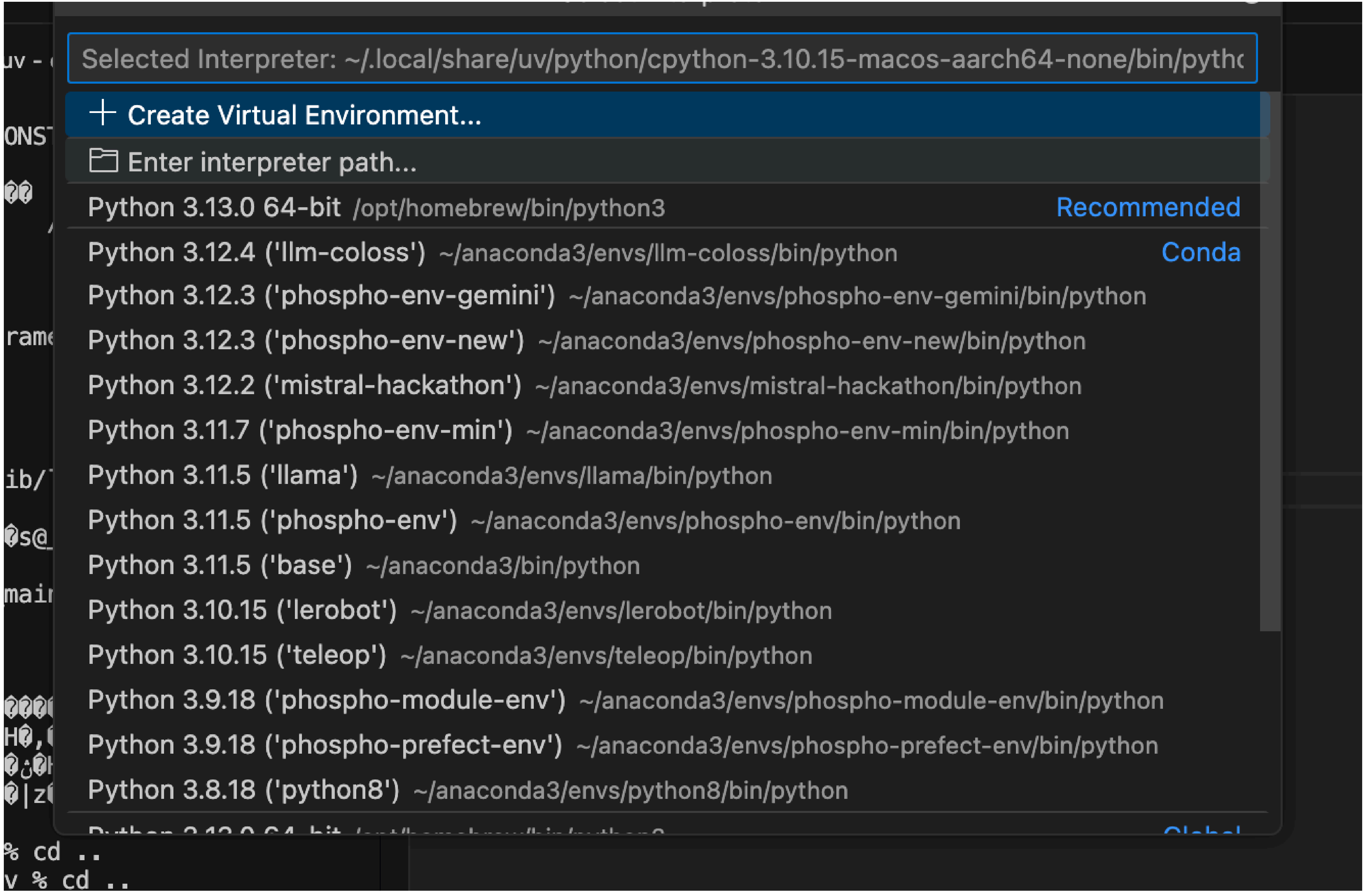
OK, this is a relief to know this is supposed to work. Here are more details:

Following the same steps as you, this is what I get after running `uv add httpx` in VSCode : "Can't resolve httpx import".
This is ok : I need to activate the python environment.



```python
import httpx


def main():
    print("Hello from example!")


if __name__ == "__main__":
    main()

```

I do Command+Shift+P. This lists the interpreters, but the local .venv doesn't show up. This is alright, I can still enter the path to it manually.

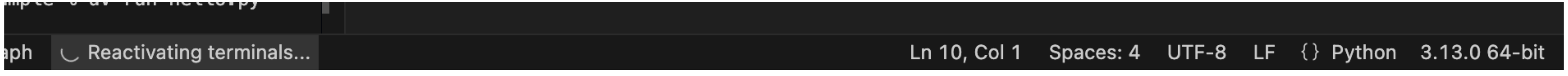


Selected Interpreter: ~/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/pytho

Python 3.13.0 64-bit /opt/homebrew/bin/python3 Recommended
Python 3.12.4 ('llm-coloss') ~/anaconda3/envs/llm-coloss/bin/python Conda
Python 3.12.3 ('phospho-env-gemini') ~/anaconda3/envs/phospho-env-gemini/bin/python
Python 3.12.3 ('phospho-env-new') ~/anaconda3/envs/phospho-env-new/bin/python
Python 3.12.2 ('mistral-hackathon') ~/anaconda3/envs/mistral-hackathon/bin/python
Python 3.11.7 ('phospho-env-min') ~/anaconda3/envs/phospho-env-min/bin/python
Python 3.11.5 ('llama') ~/anaconda3/envs/llama/bin/python
Python 3.11.5 ('phospho-env') ~/anaconda3/envs/phospho-env/bin/python
Python 3.11.5 ('base') ~/anaconda3/bin/python
Python 3.10.15 ('lerobot') ~/anaconda3/envs/lerobot/bin/python
Python 3.10.15 ('teleop') ~/anaconda3/envs/teleop/bin/python
Python 3.9.18 ('phospho-module-env') ~/anaconda3/envs/phospho-module-env/bin/python
Python 3.9.18 ('phospho-prefect-env') ~/anaconda3/envs/phospho-prefect-env/bin/python
Python 3.8.18 ('python8') ~/anaconda3/envs/python8/bin/python

```
% cd ..
v % cd ..
```

Then, I manually select the path to the .venv created by uv: `.venv/bin/python` by clicking on "Enter Interpreter path".
This doesn't enable intellisense.
I can select any other environment (either conda, either created with python -m venv) normally.
A message saying "Reactivating terminals" in the bottom left runs infinitely. Reloading the window doesn't stop this. I would need to quit and restart vscode.

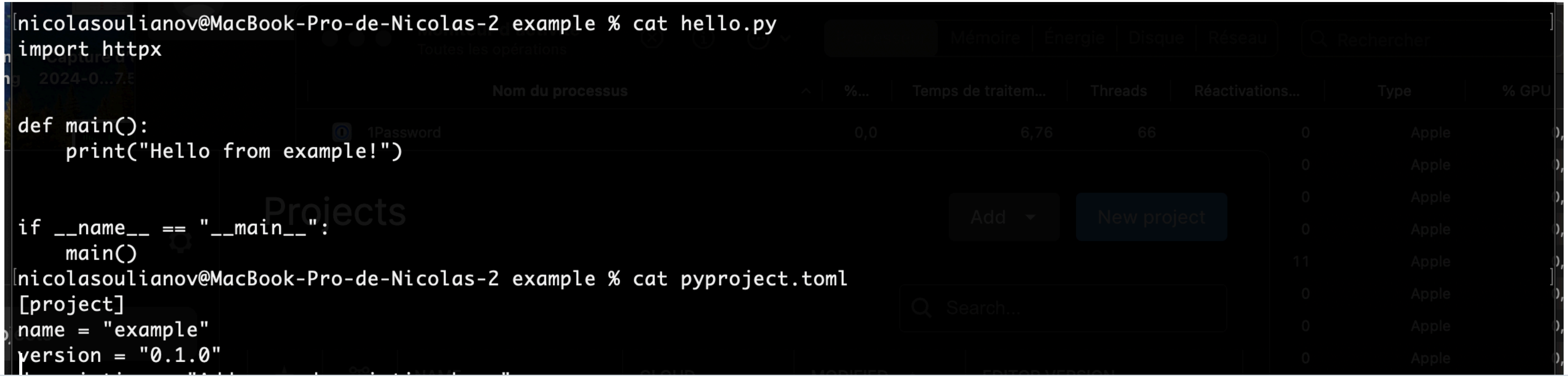


If I try to do `uv run hello.py` in the VSCode terminal, the process hangs and never completes. If I Ctrl+C to stop the execution and instead do `python hello.py`, the program runs normally.

Running `uv run hello.py` in a new terminal results in this:

```
error: Querying Python at `/Users/nicolasoulianov/test/example/.venv/bin/python3` failed with exit status signal: 9 (SIGKILL)
--- stdout:

--- stderr:

---
```

This error occurs whether VSCode is open or not.

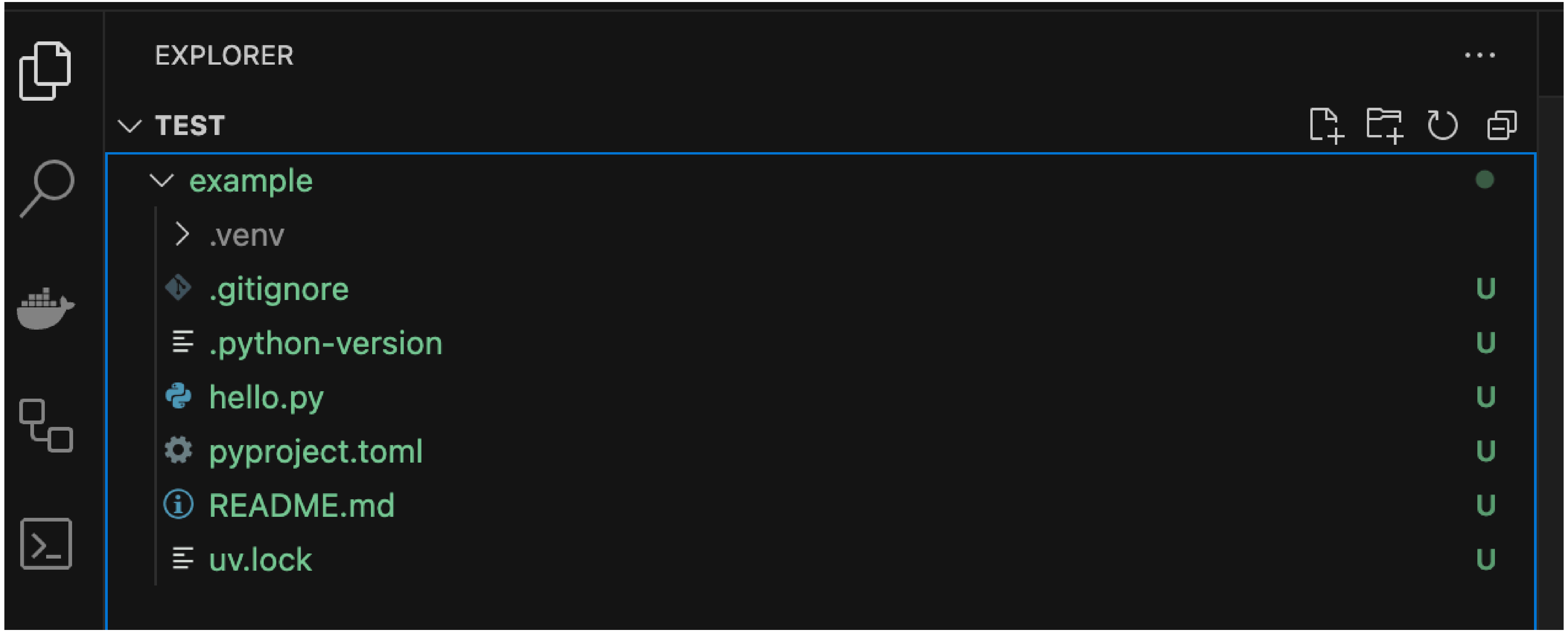


```
nicolasoulianov@MacBook-Pro-de-Nicolas-2 example % cat hello.py
import httpx


def main():
    print("Hello from example!")


if __name__ == "__main__":
    main()
nicolasoulianov@MacBook-Pro-de-Nicolas-2 example % cat pyproject.toml
[project]
name = "example"
version = "0.1.0"
```

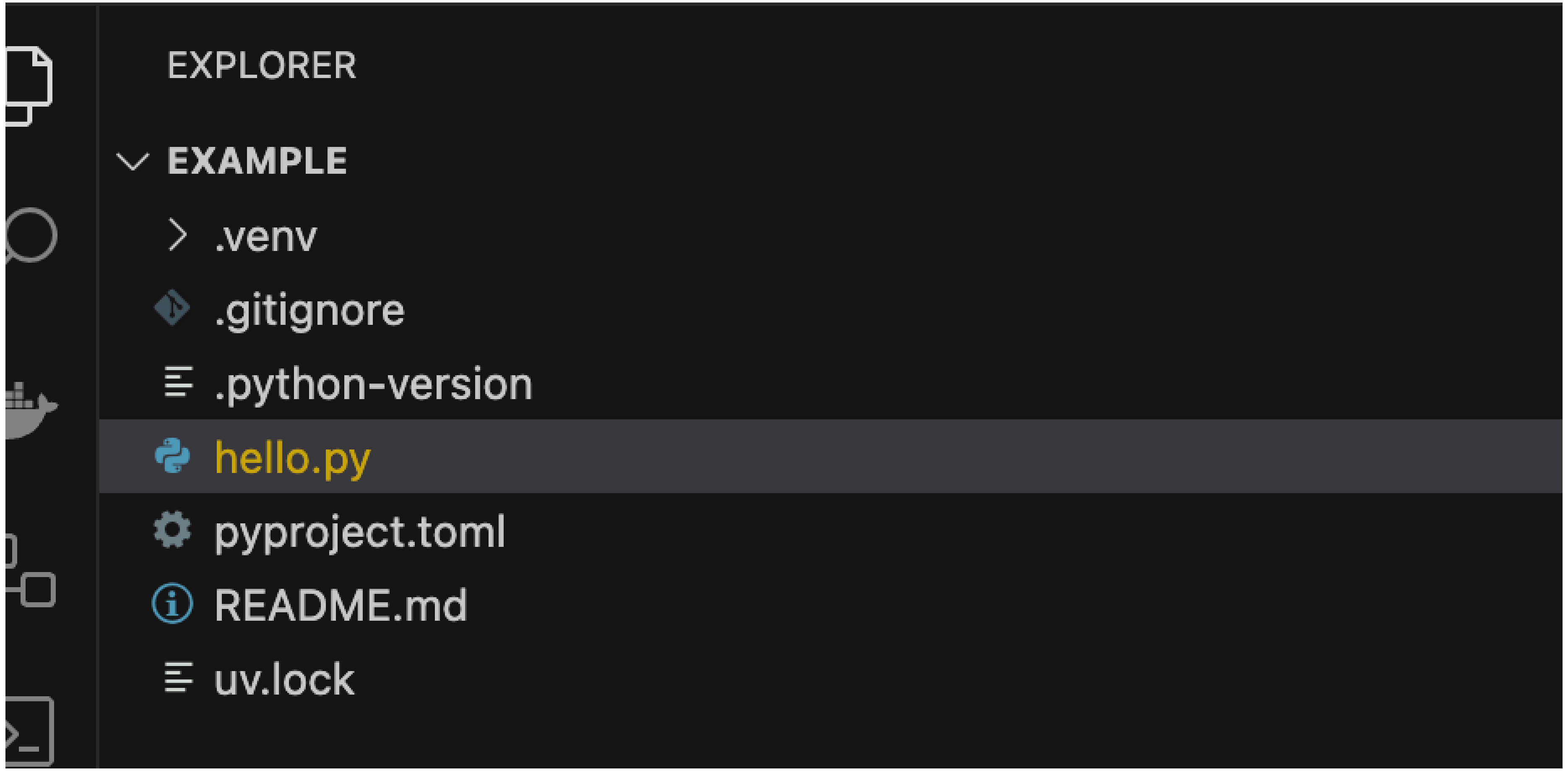
oulianov on Dec 5, 2024 Author

I think I figured this out: the uv .venv can't be selected in VSCode when the .venv is not at the root of the VSCode workspace.

For example, this doesn't work: .venv is in example/.venv



However, this works: .venv is at the root of the VSCode workspace.



This is pretty weird. I'm working with a lot of monorepo so I'm used to have multiple Python projects with different configs in the same VSCode window.

Do you know of a way to circumvent this behavior ? So that I can still have intellisense, even if the uv `.venv` is not at the root of the VSCode workspace ?

zanieb on Dec 5, 2024 Member

That's.. very weird. I'm not sure what would cause that.

Perhaps we should narrow down `uv run python` failing? Can you produce that consistently?

Can you invoke Python directly? e.g.

```
❯ uv python find --python-preference only-managed --system
/Users/zb/.local/share/uv/python/cpython-3.12.7-macos-aarch64-none/bin/python3.12
❯ /Users/zb/.local/share/uv/python/cpython-3.12.7-macos-aarch64-none/bin/python3.12 --version
Python 3.12.7
```

Can you create a virtual environment with it and use it?

```
❯ /Users/zb/.local/share/uv/python/cpython-3.12.7-macos-aarch64-none/bin/python3.12 -m venv ./venv
❯ ./venv/bin/python --version
Python 3.12.7
```

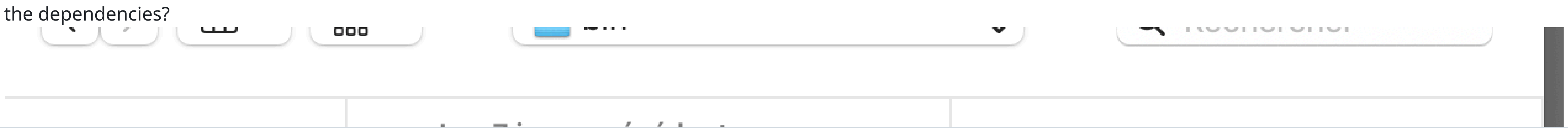
oulianov on Dec 5, 2024 Author

Unfortunately, I can't reproduce the issue with just `uv run python`

I can invoke Python directly. I can create venv with it and use it.

```
nicolasoulianov@MacBook-Pro-de-Nicolas-2 teleop % uv python find --python-preference only-managed --system
/Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10
nicolasoulianov@MacBook-Pro-de-Nicolas-2 teleop % /Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 --version
Python 3.10.15
nicolasoulianov@MacBook-Pro-de-Nicolas-2 teleop % /Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 -m venv
./venv2
nicolasoulianov@MacBook-Pro-de-Nicolas-2 teleop % ./venv2/bin/python --version
Python 3.10.15
```

Here is another piece of information. When I select the interpreter in .venv/bin/python, I see that I select a shortcut towards the true interpreter (localized at `/Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10`). Maybe this information is not enough for VSCode to make sense of the dependencies?

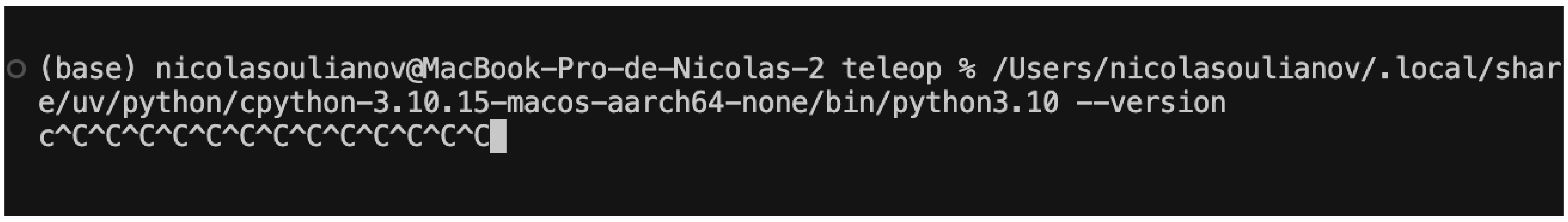


oulianov on Dec 5, 2024 Author

I opened a new VSCode window and tried again activating the .venv environment. Now, when I run

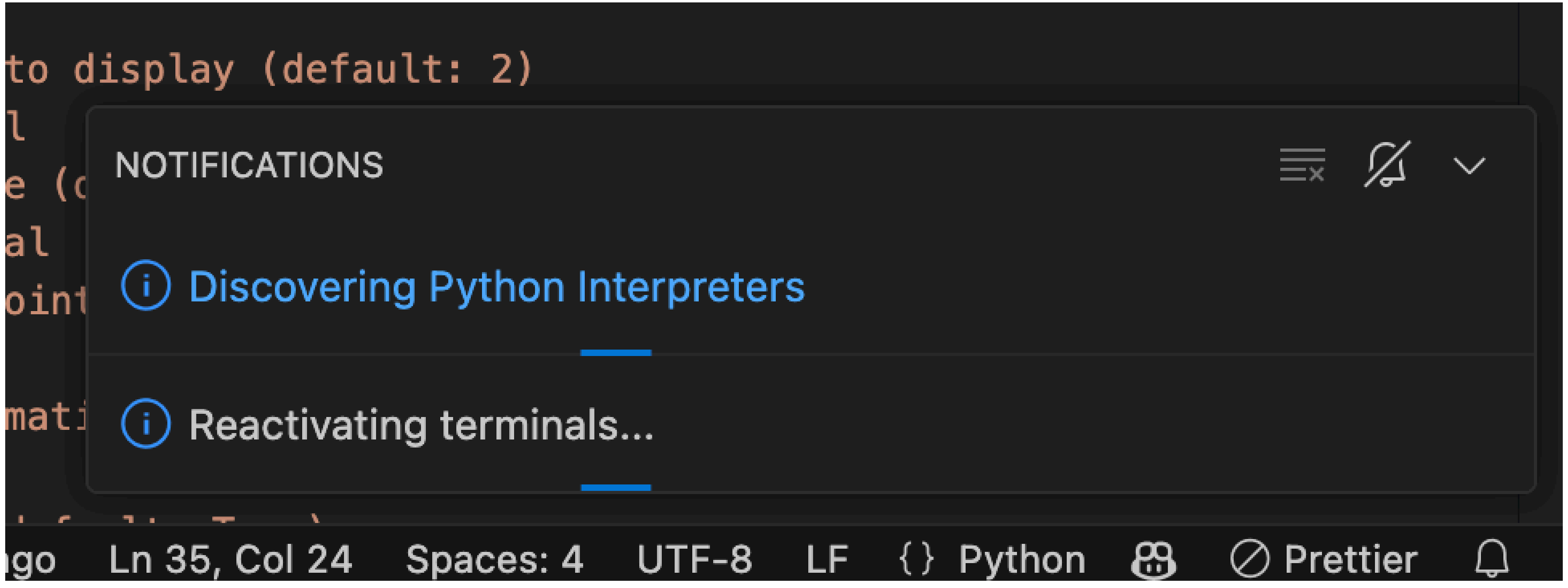
```
/Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 --version
```

The program hangs. I can't even Ctrl+C to stop it.



```
○ (base) nicolasoulianov@MacBook-Pro-de-Nicolas-2 teleop % /Users/nicolasoulianov/.local/shar
  e/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 --version
  c^C^C^C^C^C^C^C^C^C^C^C^C^C
```

I have the infinite "Reactivating terminals" loading bar



NOTIFICATIONS

ⓘ Discovering Python Interpreters

ⓘ Reactivating terminals...

I need to restart my computer to be able to use everything normally again.

👍 1

zanieb on Dec 5, 2024 Member

Does everything work if you do `uv venv --python-preference only-system` to skip using our managed interpreters?

Do you have an antivirus or similar?

Can you look at the VSCode Python debug logs?

oulianov on Dec 5, 2024 · edited by oulianov Edits Author

I enabled VSCode Python debug logs.

I don't have an antivirus. However, I'm running Vanta Agent and 1Password.

Using managed venv

If I select the managed uv .venv/bin/python as an active interpreter, I have a null error code.

```
2024-12-05 16:37:48.507 [info] > ~/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 -I ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/get_output_via_markers.py ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py
2024-12-05 16:37:48.511 [error] Error: Command failed: /Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 -I /Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/get_output_via_markers.py /Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py

    at genericNodeError (node:internal/errors:984:15)
    at wrappedFn (node:internal/errors:538:14)
    at ChildProcess.exithandler (node:child_process:423:12)
    at ChildProcess.emit (node:events:531:35)
    at maybeClose (node:internal/child_process:1105:16)
    at Socket.<anonymous> (node:internal/child_process:457:11)
    at Socket.emit (node:events:519:28)
    at Pipe.<anonymous> (node:net:339:12) {
  code: null,
  killed: false,
  signal: 'SIGKILL',
  cmd: '/Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 -I /Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/get_output_via_markers.py /Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py'
}
2024-12-05 16:37:50.513 [info] > ~/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10 -I ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/get_output_via_markers.py ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py
```

I have then the "Reactivating terminals error"

Running

```
/Users/nicolasoulianov/.local/share/uv/python/cpython-3.10.15-macos-aarch64-none/bin/python3.10
```

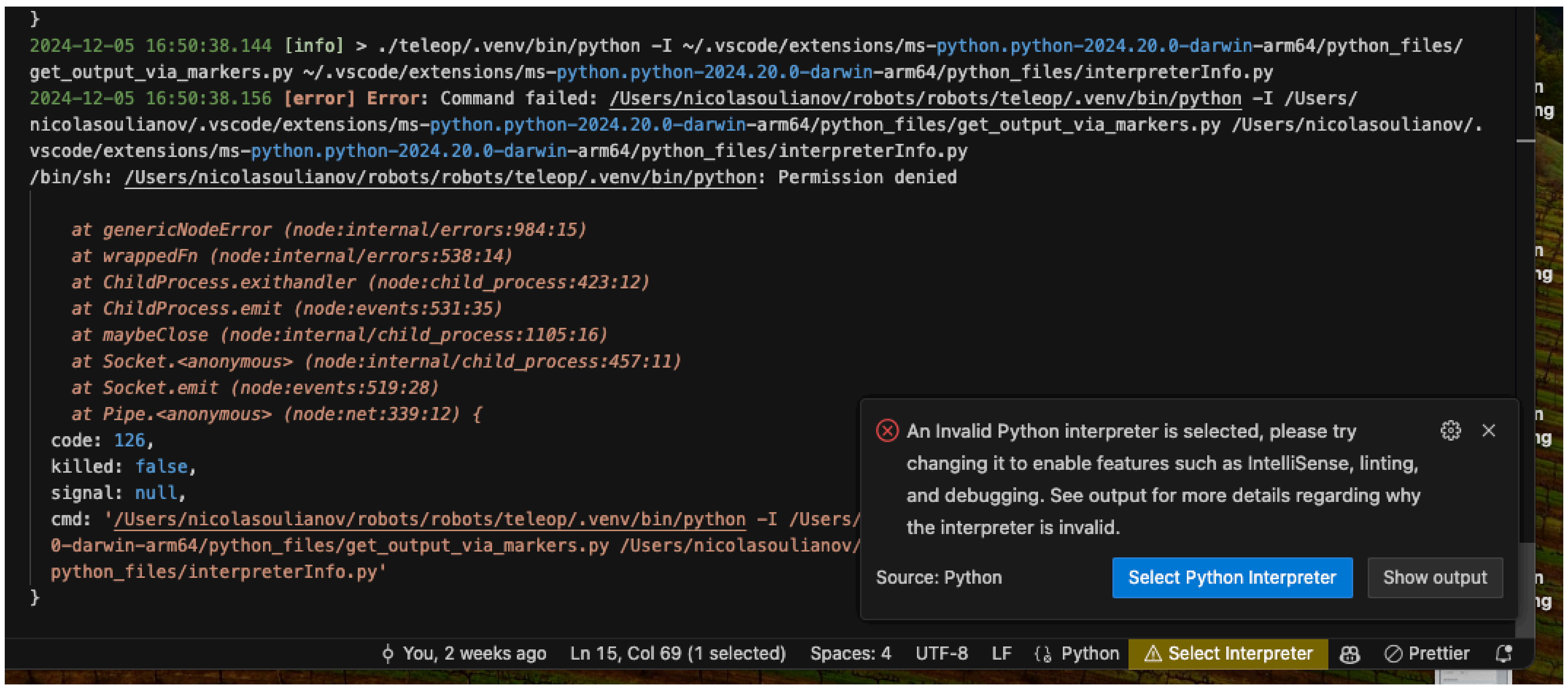
In a terminal hangs the process. It doesn't show the Python console.

Creating a new venv

I create a .venv with `uv venv --python-preference only-system` to skip the managed interpreters.

I try to select it but nothing happens, not even an error message. I delete the conda environment I had which was named `teleop` , the same name as the folder containing the .venv.

I try again to select the .venv. Now, I have an error when selecting it. The error code is 126. I see "Permission denied". I also have a visible error toaster in VSCode. This is the first time it has been the case.



```
}
2024-12-05 16:50:38.144 [info] > ./teleop/.venv/bin/python -I ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/
get_output_via_markers.py ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py
2024-12-05 16:50:38.156 [error] Error: Command failed: /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python -I /Users/
nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/get_output_via_markers.py /Users/nicolasoulianov/.
vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py
/bin/sh: /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python: Permission denied

    at genericNodeError (node:internal/errors:984:15)
    at wrappedFn (node:internal/errors:538:14)
    at ChildProcess.exithandler (node:child_process:423:12)
    at ChildProcess.emit (node:events:531:35)
    at maybeClose (node:internal/child_process:1105:16)
    at Socket.<anonymous> (node:internal/child_process:457:11)
    at Socket.emit (node:events:519:28)
    at Pipe.<anonymous> (node:net:339:12) {
  code: 126,
  killed: false,
  signal: null,
  cmd: '/Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python -I /Users/
  0-darwin-arm64/python_files/get_output_via_markers.py /Users/nicolasoulianov/
  python_files/interpreterInfo.py'
}
```

An Invalid Python interpreter is selected, please try changing it to enable features such as IntelliSense, linting, and debugging. See output for more details regarding why the interpreter is invalid.

Source: Python

```
2024-12-05 16:50:38.156 [error] Error: Command failed: /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python -I
/Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/get_output_via_markers.py
/Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py
/bin/sh: /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python: Permission denied

    at genericNodeError (node:internal/errors:984:15)
    at wrappedFn (node:internal/errors:538:14)
    at ChildProcess.exithandler (node:child_process:423:12)
    at ChildProcess.emit (node:events:531:35)
    at maybeClose (node:internal/child_process:1105:16)
    at Socket.<anonymous> (node:internal/child_process:457:11)
    at Socket.emit (node:events:519:28)
    at Pipe.<anonymous> (node:net:339:12) {
  code: 126,
  killed: false,
  signal: null,
  cmd: '/Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python -I /Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-
arm64/python_files/get_output_via_markers.py /Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-
arm64/python_files/interpreterInfo.py'
```

I try to run the Python interpreter in my console, and I get permission denied as well.

```
(teleop) (base) nicolasoulianov@MacBook-Pro-de-Nicolas-2 teleop % /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python
zsh: permission denied: /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python
```

I then change the authorizations of the .venv folder

```
sudo chmod -R 777 .venv
```

I run this command. It doesn't show anything in the command line.

```
 /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python
```

I have this error in the VSCode Python extension logs : failed to parse json

```
2024-12-05 16:59:22.698 [info] > ./teleop/.venv/bin/python -I ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-
arm64/python_files/get_output_via_markers.py ~/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py
2024-12-05 16:59:22.701 [error] Failed to parse interpreter information for >> /Users/nicolasoulianov/robots/robots/teleop/.venv/bin/python -I
/Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/get_output_via_markers.py
/Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py << with Error: python
/Users/nicolasoulianov/.vscode/extensions/ms-python.python-2024.20.0-darwin-arm64/python_files/interpreterInfo.py returned bad JSON () (SyntaxError:
Unexpected end of JSON input)
```

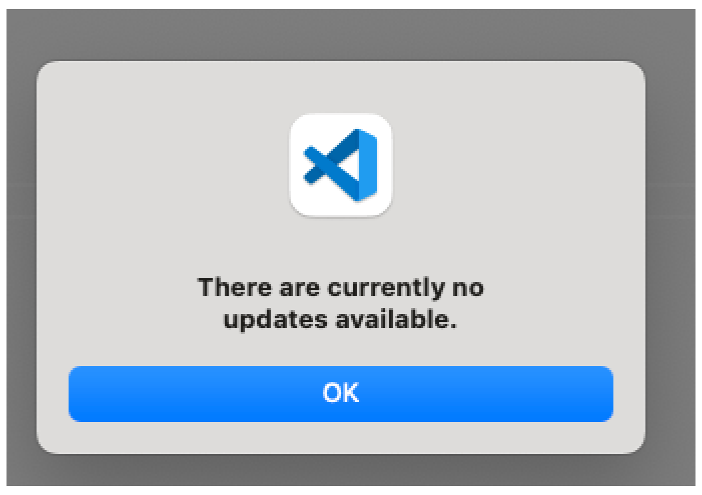
Could this be an authorization issue with uv, my home folder, MacOS "security features", or something else ?🤔

zanieb on Dec 6, 2024 Member ···

Have you tried updating VSCode to the latest version?

oulianov on Dec 6, 2024 Author ···

I am on the VSCode latest version available



There are currently no updates available.

```
Version: 1.95.3 (Universal)
Commit: f1a4fb101478ce6ec82fe9627c43efbf9e98c813
Date: 2024-11-13T14:50:04.152Z (3 wks ago)
Electron: 32.2.1
ElectronBuildId: 10427718
Chromium: 128.0.6613.186
Node.js: 20.18.0
```

```
V8: 12.8.374.38-electron.0
OS: Darwin arm64 23.6.0
```

zanieb on Dec 6, 2024 · edited by zanieb

Edits ▾ Member ···

I'm sorry I'm out of suggestions here if you've updated VS Code and restarted it. We can't reproduce this problem.

For reference, I'm using the exact same build

```
Version: 1.95.3
Commit: f1a4fb101478ce6ec82fe9627c43efbf9e98c813
Date: 2024-11-13T14:50:04.152Z (3 wks ago)
Electron: 32.2.1
ElectronBuildId: 10427718
Chromium: 128.0.6613.186
Node.js: 20.18.0
V8: 12.8.374.38-electron.0
OS: Darwin arm64 23.5.0
```

oulianov on Dec 10, 2024 · edited by oulianov

Edits ▾ Author ···

Hmm... This is all very strange.

I assume there must something cursed about my setup :

- multiple .venv in a single VSCode workspace
- nested python projects

I'll simplify my setup and keep you posted.

In the meanwhile, using an additional .venv for development and uv for deployment works good enough.

Thank you for the support!

NellyWhads on Dec 18, 2024

···

I'd just like to confirm that I'm using the following setup with success:

- Python monorepo
- Single uv workspace with multiple projects
- using `direnv` to set `UV_PROJECT_ENVIRONMENT` within each project directory such that `uv` creates a separate `.venv` in the project root instead of the workspace root
- VSCode opened to the workspace directory -> manually selecting the appropriate `.venv` -> `Reload Window`
- Testing integration also works so long as I point VSCode to the specific `pyproject.toml` which houses the `pytest` config for the specific project.

👍 1

79 remaining items

Load more

Duc-NSH on Jul 23

···

> The only thing that worked for me was reverting to the older "js" python locator with this setting: `"python.locator": "js"` - from here. Obviously not a great long term solution, but nothing else worked.
>
> I reported this here

Thank you very much! This solution worked for me. Although I mostly use NeoVim, I still need to use VsCode for certain tasks. The bug where VsCode doesn't detect non-root .venv makes me dislike opening it every time.

hsiehbocheng on Jul 28

···

Not sure if I'm misunderstanding your issue, but I ran into something very similar.
VS Code couldn't find the Python interpreter in my .venv from uv, and I eventually discovered it was because the virtual-env lived inside a sub-folder (e.g. project/sub-module/.venv).

By default a VS Code workspace only scans the top-level folder for interpreters, so anything in deeper paths is ignored. The fix was to switch to a multi-root workspace (File ▸ Add Folder to Workspace...) and add the sub-folder as its own root. As soon as I did that, the .venv interpreter showed up.

Docs for reference: https://code.visualstudio.com/docs/editing/workspaces/multi-root-workspaces

❤️ 2

codethief on Jul 28 · edited by codethief

Edits ▾ ···

@Duc-NSH @hsiehbocheng

> The bug where VsCode doesn't detect non-root .venv

> I eventually discovered it was because the virtual-env lived inside a sub-folder (e.g. project/sub-module/.venv).
> By default a VS Code workspace only scans the top-level folder for interpreters, so anything in deeper paths is ignored.

Correct, and – with all due respect – we have discussed this various times already (see e.g. the 4th comment here). The bug/deficiency is in Pyright/Pylance[0], and has nothing to do with uv.

Overall, I feel like we're going in circles, so **let me summarize all the solutions & workarounds people here in this thread have identified so far**. I added a suffix `(?)` to solutions I haven't verified myself and where there hasn't been a lot of feedback yet.

Solution 1: Set the interpreter path by hand manually, as discussed in the very first post here as well as various others. Note that there seems to be an additional issue regarding symlinks, see 1, 2, 3. Either way, this solution doesn't work great if you're in a monorepo or often switch between multiple projects/workspaces in VSCode.

Solution 2: Use a multi-root workspace in VSCode, i.e. one root for every Python project. This isn't exactly great if you're in a monorepo with a lot of projects and collaborators, each of whom will have to set up their VSCode instance by hand, but it seems it is what the Pyright devs recommend.

Solution 3: Like 1 but use an extension like teticio.python-envy to automatically set the interpreter path based on the file you're currently looking at, compare #9637 (comment) . This is what I currently use and it has been working quite well in my monorepo setup.

Solution 4(?): Use the new ms-python.vscode-python-envs extension, as per #9637 (comment) .

Solution 5(?): Set `"python.locator": "js"` , compare #9637 (comment) .

Side note regarding solutions 1 and 3: You might also want to add a `pyrightconfig.json` to your workspace root and set different `executionEnvironments` for your different projects there. This should ensure that import suggestions work as expected. E.g.:

```
{
    "executionEnvironments": [
        {
            "root": "./python-project1"
        },
        {
            "root": "./python-project2"
        },
        // …
    ],
    // …
}
```

[0]: Related tickets:

- ⊘ Feature request: Determine Python project root & venv by looking for pyproject.toml (aka monorepo support) microsoft/pyright#10498
- ⊙ How to configure with multiple packages and uv microsoft/pylance-release#7151
- ✓ Support for Monorepo with multiple virtual environments microsoft/pylance-release#3899
- ⊙ Proposal: Use Nearest Configuration File microsoft/pylance-release#5564
- ⊙ Improved mono repo support microsoft/vscode-python#21204

❤️ 9

jyaros on Aug 9 ···

Personally i was able to add uvenv as an interpreter only once I reopened the git repo at the top level of the workspace so that the env folder was at root. If you dont **need** to open up a higher parent directory for your work, this is the easiest way to get around the issue ...

zanieb on Aug 9 Member ···

If there's a canonical way to configure this correctly, e.g., as described by **@codethief**, I think we should add it to an integration guide in the documentation.

konosubakonoakua on Aug 18 ···

> The only thing that worked for me was reverting to the older "js" python locator with this setting: `"python.locator": "js"` - from here. Obviously not a great long term solution, but nothing else worked.
>
> I reported this here

Thank you, work like a charm.

👍 1

eleanorjboyd on Aug 18 ···

Hello everyone! Wanted to hop in per **@konstin** suggestion. I am an engineer on the VS Code team specifically working on Python and environments. We are beginning to roll out a new extension called the Python Environments Extension which is a new extension dedicated to all environment and package management for Python in VS Code. This will serve as the backbone for Python support in VS Code, a required extension which all Python extensions (the core Python, the Python Debugger, and various linter/formatter extensions) will rely on.

With this being said, we encourage everyone encountering issues with their environments in VS Code to install the new extension and add to their user settings: `python.usePythonEnvironmentsExtension: true` . Once enabled you may find your issue is fixed (as we have made significant improvements to the environment experience) or it may still persist. If you still have it, please file an issue on this repo and I will get back to you to figure out and solve the bug. This will make sure your bug is fixed moving forward instead of on the soon-to-be deprecated code for environments which lives on the Python extension.

Feel free to reach out with any outstanding questions in that repository and I hope everyone adopts the new extension as we are excited about the new features which it enables!

❤️ 7

ggirelli on Aug 19 ···

> With this being said, we encourage everyone encountering issues with their environments in VS Code to install the new extension and add to their user settings: `python.usePythonEnvironmentsExtension: true` .

This fixed the issue for me. Thank you so much! 🙏

konstin added **external** and removed **needs-mre** on Aug 19

hsuabina on Aug 19 ···

> Solution 3: Like 1 but use an extension like teticio.python-envy to automatically set the interpreter path based on the file you're currently looking at, compare #9637 (comment) . This is what I currently use and it has been working quite well in my monorepo setup.

I'm also working on a monorepo (with packages of all sorts) and for someone that's not so used to work on python projects this solution was simple and effective.

codethief on Aug 21 · edited by codethief Edits ▾ ···

@eleanorjboyd I've seen your new extension be mentioned a few times already but so far no one ever spelled out what it does differently compared to the old Python extension. Could you elaborate? Does it support monorepos? (I.e. multiple Python projects / venvs per workspace.) Does it automatically recognize Python projects / venvs or do I have to configure things by hand? Etc.

👍 1

matthias-busch last month

> Solution 3: Like 1 but use an extension like teticio.python-envy to automatically set the interpreter path based on the file you're currently looking at, compare #9637 (comment) . This is what I currently use and it has been working quite well in my monorepo setup.

Gotta say this does work quite well in my monorepos. Thanks for the tip!

👍 2

mironnn 3 days ago

had the same issue.

Disabling this buggy sh..t solved the issue.
https://marketplace.visualstudio.com/items?itemName=ms-python.vscode-python-envs is recommended to be uninstalled.

eleanorjboyd 3 days ago

Hi @mironnn- would appreciate if you file an issue on our repo if you are seeing a problem so I can fix it! Thanks

Sign up for free **to join this conversation on GitHub.** Already have an account? Sign in to comment

Assignees

No one assigned

Labels

external

Type

No type

Projects