

Join GitHub today

Dismiss

GitHub is home to over 31 million developers
working together to host and review code,
manage projects, and build software together.

Sign up

Working with packages and environments in Spyder

C.A.M. Gerlach edited this page on 12 Sep 2018 · 18 revisions

While relatively straightforward once you're familiar with it, the interaction between Spyder and other packages and environments can sometimes be confusing for first-time users. Spyder 4 will make this process much easier with integrated, interactive GUI package and environment management, but in the meantime—particularly with the changes released in Spyder 3.3.0—we'd like to clarify how that relationship works.

The most common problem: Using newly-installed packages inside Spyder

After installing a package (let's call it `foo`) outside Spyder, users may encounter an error trying to import it inside the IDE:

```
In [1]: import foo
Traceback (most recent call last):

  File "<ipython-input-4-7f58dd7fb72e>", line 1,
    import foo

ModuleNotFoundError: No module named 'foo'
```

This happens because `foo` was installed (with either `conda` or `pip`) in a different `conda` or `venv/virtualenv` environment than the one in which Spyder is currently running.

To confirm this is the problem, you need to:

1. Activate the environment (e.g. `myenv`) in which you installed the package `foo` (e.g. with `source activate myenv` on macOS/Linux or `activate myenv` on Windows, `workon myenv` for `virtualenv/venv`, etc)
2. Start a Python interpreter there by running the command `python`.

▼ Pages 37

[Home](#)
[Anaconda stopped funding Spyder](#)
[API decoupling](#)
[Beta version changelog](#)
[Contributing to Spyder](#)
[Current Funding and Development Status](#)
[Dev: Coding Style](#)
[Dev: Cookbook](#)
[Dev: Debugging Spyder](#)
[Dev: Github Workflow](#)
[Dev: Index](#)
[Dev: Run from source](#)
[Dev: Spyder Internals](#)
[Dev: Testing](#)
[Dev: Translations](#)
[Show 22 more pages...](#)

Clone this wiki locally



