

Introducing mypy in code with lots of issues

Tom Rochette <tom.rochette@coreteks.org>

December 21, 2025 — [77e1b28a](#)

1 Problem

I want to include `mypy` as part of my CI pipeline but my existing code contains a lot (> 100 , but < 500) of issues. How can I get started?

2 Solution

Create a minimalist configuration of `mypy` such that it will list issues that need to be fixed and return a non-zero exit code. Based on the problem definition, we assume that at this step you have more than 100 issues that are listed and that fixing those issues will take many hours you'd rather invest in improving the code than to fix typing issues.

Add a step in your CI pipeline that runs `mypy` and list all those issues. Verify that it indeed breaks the build.

Once you've satisfied yourself that CI fails, we will “fix” the `mypy` issues by adding the `#type: ignore` and/or `# noqa` comment after the offending lines with issues. This will have the effect of resolving all the currently found `mypy` issues, such that `mypy` should now return a zero exit code. With this, any future code that fails to pass the `mypy` check will break the build. This will allow you to use `mypy` from this point forward to check your types.

I suggest adding an additional comment such as `# FIXME: TICKET-ID`, where `TICKET-ID` refers to the id of a ticket in your issue tracking system that explains that you need to take care of this technical debt.

Always prefer to fix the issues instead of ignoring them. However, also consider whether fixing those issues is an appropriate use of your time when you want to introduce `mypy` (which should be as soon as possible in my opinion).

3 References

- https://mypy.readthedocs.io/en/stable/common_issues.html#silencing-linters