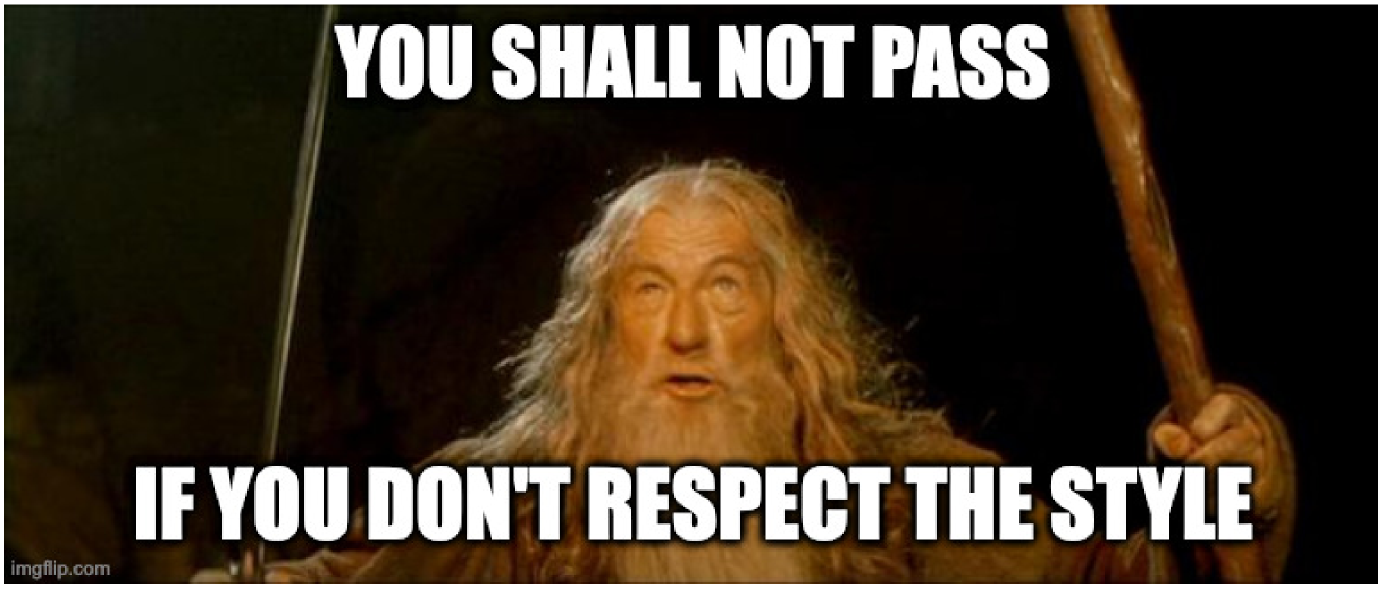
DataScientest Code Style



Disclaimer:

This code style applies to all Python repositories inside the DataScientest organization. For other languages like Javascript, another confluence can be created.

***This confluence page is brought to you by the PEP8 and the Black code style.* Please read those before continuing**

Template repository

A template repository has been created to simplify creation of micro-services and some if not most of the code will be taken from there.

First steps when creating a new repository

When creating a new repository, there’s a few things that always need to be done (and will be detailed in the next sections)

- Installing pre-commit and enabling it
- Sentry (if the repo is a service)
- Setting up your CI/CD

pre-commit

What is *pre-commit* ?

From the official git documentation:

> *The pre-commit hook is run first, before you even type in a commit message. It's **used to inspect the snapshot that's about to be committed**, to see if you've forgotten something, to make sure tests run, or to examine whatever you need to inspect in the code.*

And from https://pre-commit.com/ :

> *Git hook scripts are useful for identifying simple issues before submission to code review. We run our hooks on every commit to automatically point out issues in code such as missing semicolons, trailing whitespace, and debug statements. By pointing*

> *these issues out before code review, this allows a code reviewer to focus on the architecture of a change while not wasting time with trivial style nitpicks.*

As you've guessed it, pre-commit allows us to check the code before even commiting is, to check if it fits with the code style. In our case, `pre-commit` runs `black`, `flake8` and `reorder-python-imports`.

⚠ After running pre-commit when commiting, changes might be made and need to be added again.

black

> *Black is the uncompromising Python code formatter. By using it, you agree to cede control over minutiae of hand-formatting. In return, Black gives you speed, determinism, and freedom from pycodestyle nagging about formatting. You will save time and mental energy for more important matters.*

It will for example wrap too long lines or replace `'` with `"` etc… It allows us to automate a big part of our code style without having to worry about it. You can find more info about it here

flake8

> *Flake8 is a wrapper around these tools:*
>
> *PyFlakes*
>
> *pycodestyle*
>
> *Ned Batchelder's McCabe script*

Which basically means that flake8 checks your code *but doesn't modify it automatically*. Usually black fixes most of flake8's errors but you do need to fix some yourself.

Flake8 will for example find unused imports, variables etc but won't delete them for you.

reorder-python-imports

From the doc:

> *Tool for automatically reordering python imports. Similar to `isort` but uses static analysis more.*

This will reorder and sort your imports to a nice and clean style.

Sentry

Sentry is a tool used at DataScientest allowing quick access and solving of issues. Every event, every crash is captured and sent to http://sentry.io

⚠ Only do this if the repository is a service/micro service. Sentry won't work/is useless on private packages.

To configure sentry, you need to first create a new project here

Then copy the DSN and place it in the `.env` file and update the rest of sentry's code accordingly

datadog

logs instead of comment

git "flow"

CI/CD

github actions

code owners

review process