

Clippy and linting

With `cargo check` you have an integrated linter in Rust. But additionally there is another linting tool called **clippy**. It is usually installed per default and can be executed using `cargo clippy`. It provides further linting options.

Adding linting rules

You can for example add linting rules by activating them in your `main.rs` or `lib.rs`:

```
#![warn(clippy::pedantic)]
#![warn(clippy::nursery)]
#![warn(clippy::unwrap_used)]
#![warn(clippy::expect_used)]
#![warn(clippy::todo)]
#![warn(clippy::panic)]
```

In this example, there are some additional warnings added, like when `unwrap` or `expect` is used somewhere. With this you can set additional rules to increase your code quality. This is useful e.g. when you use `unwrap` for prototyping, but you want to make sure to replace all `unwraps` with proper error handling later.

Clippy and VS Code

VS Code usually checks using the `check` command of `cargo`. To activate `clippy` instead, go to the settings JSON of your workspace and add the following command:

```
{
  "rust-analyzer.check.command": "clippy"
}
```

Afterwards, you probably need to restart the rust analyzer (CTRL+ALT+P and then search for **rust-analyzer: Restart Server**).

Revision #1

Created 2026-07-06 09:49:30 UTC by Matthias

Updated 2026-07-06 09:56:24 UTC by Matthias