Managing dependencies with Cargo.toml

In PHP

`composer require monolog/monolog` writes the package into `composer.json` and the exact version into `composer.lock`. Packagist is the index. You commit `composer.lock` for an application so the next install resolves the same versions. A library often does not commit it. `require-dev` is for PHPUnit and friends, not for production.

In Rust

crates.io is Packagist. `Cargo.toml` is `composer.json`. `Cargo.lock` is `composer.lock`.

Add a crate with Cargo. Do not type the version in by hand unless you are fixing one.

crate root

```
cargo add serde --features derive
```

Cargo writes the full current version it just resolved, for example:

Cargo.toml · the dependency Cargo just added

```
[dependencies]
serde = { version = "1.0.228", features = ["derive"] }
```

A bare version is a caret requirement: `"1.0.228"` means `^1.0.228`, i.e. `>=1.0.228, <2`. The lockfile picks the exact build you compiled against.

The exact version you compiled against goes in `Cargo.lock`. Commit that file for an application, including this course's CLIs and the API you will build later. Cargo now also says libraries may commit `Cargo.lock` so contributors and CI share the same graph — that differs from the old Composer habit of leaving a library unlocked. `cargo update` moves dependencies inside the ranges in `Cargo.toml` and rewrites the lockfile. CI should build with `cargo build --locked` so a missing or edited lockfile fails the build instead of quietly resolving something new.

Dev-only tools use `[dev-dependencies]`, which is `require-dev`.

crate root

```
cargo add --dev pretty_assertions
```

Those crates are available to tests, not to the shipped binary.

> **Watch for:** committing an application without `Cargo.lock`, then wondering why two machines compiled different versions.

> **Idiomatic Rust:** `cargo add`, then commit the lockfile for anything you run. Use `--locked` in CI.

Side by side

	PHP	Rust
Manifest	`composer.json`	`Cargo.toml`
Exact versions	`composer.lock`	`Cargo.lock`
Add a package	`composer require`	`cargo add`

Try it

In the hello-cargo project, run `cargo add serde --features derive` and open `Cargo.toml` and `Cargo.lock`. You should see serde in both.

Key takeaways

- `cargo add` edits `Cargo.toml`. `Cargo.lock` records the exact build.
- Commit `Cargo.lock` for applications. CI uses `--locked`.
- `[dev-dependencies]` is Composer's `require-dev`.

What's next: Stable, nightly, and pinning a version