

Cargo.toml Best Practices (Crates vs Dependencies)

SCRT Labs and Crates.io

Crates.io serves as a centralized repository for Rust packages, also known as "crates." When building applications or libraries with Rust, developers often need to use external libraries for functionality not provided by the standard library, many of which are included on Crates.io. Below is an example of how to import [Secret Labs' crates](#), such as `secret-cosmwasm-std`, a fork of the original `cosmwasm-storage` repository adapted for use in SecretNetwork's Secret Contracts.

...

```
Copy [dependencies] cosmwasm-std={ package="secret-cosmwasm-std", version="1.1.10"} cosmwasm-storage={
package="secret-cosmwasm-storage", version="1.1.10"} secret-toolkit-storage="0.9.0" secret-toolkit={
git="https://github.com/scrtlabs/secret-toolkit", tag="v0.8.0", default-features=false, features=[ "storage", "viewing-key",
"crypto", "utils", ] } schemars={ version="0.8.11"} serde={ version="1.0"} thiserror={ version="1.0"} cosmwasm-
schema="1.0.0"
```

...

Further Reading: Crates vs Dependencies

You can think of "crates" as individual packages or modules in Rust, while "dependencies" refer to external crates that your project relies on for additional functionality.

When developing a Rust project, you will often have dependencies on other crates. These dependencies can come from crates.io (the default Rust package registry), a Git repository (such as one hosted by Secret Labs!), or a local path on your system. The Rust package manager, Cargo, is responsible for managing these dependencies, ensuring that the correct versions are fetched and built, and handling transitive dependencies (i.e., the dependencies of your dependencies).

- Crates: A crate is a package or a module in Rust containing code and resources, like libraries or applications. Crates are the building blocks of Rust projects and are intended to be easily shared and reused. A crate can be a binary (application) or a library. Each crate has a unique name and a `Cargo.toml` file containing metadata, such as the crate's name, version, authors, and dependencies.
- Dependencies: Dependencies refer to external crates that your Rust project or library relies on for additional functionality. These external crates are not part of your project's codebase but are required for your project to build and function correctly. Dependencies are declared in the `Cargo.toml` file of your project under the `[dependencies]` section.
-

Last updated 7 months ago On this page * [SCRT Labs and Crates.io](#) * [Further Reading: Crates vs Dependencies](#)

Was this helpful? [Edit on GitHub](#) [Export as PDF](#)