

Cargo Package Registry



- [Requirements](#)
 - [Index Repository](#)
 - [Configuring the package registry](#)
 - [Credentials](#)
 - [Git vs Sparse](#)
 - [Publish a package](#)
 - [Install a package](#)
 - [Supported commands](#)
-

Publish [Cargo](#) packages for your user or organization.

Requirements

To work with the Cargo package registry, you need [Rust and Cargo](#).

Cargo stores information about the available packages in a package index stored in a git repository. This repository is needed to work with the registry. The following section describes how to create it.

Index Repository

Cargo stores information about the available packages in a package index stored in a git repository. In Forgejo, this repository has the special name `_cargo-index`. After a package was uploaded, its metadata is automatically written to the index. The content of this repository should not be manually modified.

The user or organization package settings page allows to create the index repository along with the configuration file. If needed, this action will rewrite the configuration file. This can be useful if, for example, the Forgejo instance domain was changed.

If the case arises where the packages stored in Forgejo and the information in the index repository are out of sync, the settings page allows to rebuild the index repository. This action iterates all packages in the registry and writes their information to the index. If there are lot of packages this process may take some time.

Configuring the package registry

To register the package registry the Cargo configuration must be updated. Add the following text to the configuration file located in the current users home directory (for example:

`~/.cargo/config.toml`):

```
[registries.{registry_name}]
index = "sparse+https://forgejo.example.com/api/packages/{owner}/cargo/" # Sparse index
# index = "https://forgejo.example.com/{owner}/_cargo-index.git" # Git

# [net]
# git-fetch-with-cli = true
```

Parameter	Description
<code>owner</code>	The owner of the package.
<code>registry_name</code>	The name used to refer to the registry in commands.

Credentials

If the registry is private or you want to publish new packages, you must use your credentials to access it.

The Cargo registry requires an [API token](#) with the "write:packages" permission. To obtain a token usable by Cargo, prepend the string "Bearer " to your API token.

To store the token in your configuration, add the credentials section to the credentials file located in the current users home directory (for example `~/.cargo/credentials.toml`):

```
[registries.{registry_name}]
token = "Bearer {token}"
```

Parameter	Description
-----------	-------------

<code>token</code>	Your personal access token
<code>registry_name</code>	The name used to refer to the registry in commands.

Git vs Sparse

Currently, cargo supports two ways for fetching crates in a registry: Git index & sparse index. Sparse index is the newest method and offers better performance when updating crates compared to git. Since Rust 1.68, sparse is the default method for crates.io.

Publish a package

Publish a package by obtaining an [API token](#) and running the following command in your project:

```
cargo publish --registry {registry_name}
```

Parameter	Description
<code>registry_name</code>	The name of the registry in <code>.cargo/config.toml</code> .

You cannot publish a package if a package of the same name and version already exists. You must delete the existing package first.

Install a package

To install a package from the package registry, execute the following command:

```
cargo add {package_name} --registry {registry_name}
```

Parameter	Description
<code>package_name</code>	The package name.
<code>registry_name</code>	The name of the registry in <code>.cargo/config.toml</code> .

Supported commands

cargo publish

cargo add

cargo install

cargo yank

cargo unyank

cargo search

📦 #3

📦 cocobook-sync 📦 2026-10-28 16:38:31 UTC

📦 cocobook-sync 📦 2026-10-29 01:09:31 UTC