

Swift Packages Repository



- [Swift Packages Repository](#)
 - [Requirements](#)
 - [Configuring the package registry](#)
 - [Publish a package](#)
 - [Install a package](#)

Swift Packages Repository

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

Requirements

To work with the Swift package registry, you need to use [swift](#) to consume and a HTTP client (like `curl`) to publish packages.

Configuring the package registry

To register the package registry and provide credentials, execute:

```
swift package-registry set https://forgejo.example.com/api/packages/{owner}/swift
swift package-registry login https://forgejo.example.com/api/packages/{owner}/swift --username {username} --password {password}
```

Placeholder	Description
<code>owner</code>	The owner of the package.
<code>username</code>	Your Forgejo username.
<code>password</code>	Your Forgejo password. If you are using 2FA or OAuth use a personal access token instead of the password.

The login is optional and only needed if the package registry is private.

Publish a package

First you have to pack the contents of your package:

```
swift package archive-source
```

To publish the package perform a HTTP PUT request with the package content in the request body.

```
curl -X PUT --user {username}:{password} \
  -H "Accept: application/vnd.swift.registry.v1+json" \
  -F source-archive=@/path/to/package.zip \
  -F metadata={metadata} \
  https://forgejo.example.com/api/packages/{owner}/swift/{scope}/{name}/{version}
```

Placeholder	Description
<code>username</code>	Your Forgejo username.
<code>password</code>	Your Forgejo password. If you are using 2FA or OAuth use a personal access token instead of the password.
<code>owner</code>	The owner of the package.
<code>scope</code>	The package scope.
<code>name</code>	The package name.
<code>version</code>	The package version.
<code>metadata</code>	(Optional) The JSON-encoded metadata of the package, see Swift documentation

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

The server responds with the following HTTP Status codes.

HTTP Status Code	Meaning
<code>201 Created</code>	The package has been published.
<code>400 Bad Request</code>	The package is invalid.
<code>409 Conflict</code>	A package file with the same combination of parameters exists already.

Install a package

To install a Swift package from the package registry, add it in the `Package.swift` file dependencies list:

```
dependencies: [  
  .package(id: "{scope}.{name}", from:"{version}")  
]
```

Parameter	Description
<code>scope</code>	The package scope.
<code>name</code>	The package name.
<code>version</code>	The package version.

Afterwards execute the following command to install it:

```
swift package resolve
```

	#3	
	cocobook-sync	 2026-10-28 16:38:59 UTC
	cocobook-sync	 2026-10-29 01:09:32 UTC