



Creating Packages

Home

Creating Packages

[bower.json](#)[Specification](#)[Maintaining dependencies](#)[Register](#)

API

Configuration

Pluggable Resolvers

Tools

About

[!\[\]\(19d44b37fb4fa155bf9d60c77a3d3cb2_img.jpg\) Bower on GitHub](#)[!\[\]\(5a351309c3b87e4420622c1f0e57efc0_img.jpg\) @bower](#)[!\[\]\(bff896c19919791b89ab521f039b410a_img.jpg\) Get support on discord](#)[!\[\]\(23a2e9ddc7bb0ef55393d38b772a848d_img.jpg\) Support on Bountysource](#)

bower.json

Packages are defined by a manifest file `bower.json`. This is similar to Node's `package.json` or Ruby's `Gemfile`.

Interactively create a `bower.json` with [bower init](#)

```
$ bower init
```

Specification

Detailed specification of `bower.json` file can be found in [bower/spec](#) repository.

§ Maintaining dependencies

Using `bower install <package> --save` will add `<package>` to your project's `bower.json` dependencies array.

```
# install package and add it to
# bower.json dependencies
$ bower install <package> --save
```

Similarly, using `bower install <package> --save-dev` will add `<package>` to your project's `bower.json devDependencies` array.

```
# install package and add it to  
bower.json devDependencies  
$ bower install <package> --save-dev
```

Register

Registering your package allows others to install it with a short name, like `bower install <my-package-name>`.

To register a new package:

- The package name **must** adhere to the [bower.json spec](#).
- There **must** be a valid `bower.json` in the project's root directory.
- Your package should use [semver](#) Git tags. The `v` prefix is allowed.
- Your package **must** be publically available at a Git endpoint (e.g., GitHub). Remember to push your Git tags!
- For private packages (e.g. GitHub Enterprise), please consider running a private [Bower registry](#).

Then use `bower register`:

```
$ bower register <my-package-name>  
<git-endpoint>  
# for example  
$ bower register example  
git://github.com/user/example.git
```

Now anyone can run `bower install <my-package-name>`, and get your library installed. The Bower registry does not have authentication or user management at this point in time. It's on a first come, first served basis.

Bower doesn't support GitHub-style namespaces (`org/repo`), however you are encouraged to namespace related packages with `-`, for example, `angular-` and `paper-`.

Please do not squat on package names. Register your package and claim your name after you have a valid public repo with working code.

For package name transfers, intellectual property and other disputes, please try to resolve with the owner first. If no resolution, please submit a ticket in the [Bower Registry repo](#) and the Bower Core Team will assist.

Unregister

You can unregister packages with `bower unregister`. You first need to authenticate with GitHub with `bower login` to confirm you are a contributor to the package repo.

```
bower login
# enter username and password
? Username:
? Password:
# unregister packages after successful
Login
bower unregister <package>
```

You'll likely want to `bower cache clean` after your change. Please remember it is generally considered bad behavior to remove versions of a library that others are depending on. Think twice :) If the above doesn't work for you, you can [request a package be unregistered manually](#).

[Help improve these docs. Open an issue or pull request.](#)