

npm-registry

The JavaScript Package Registry

DESCRIPTION

To resolve packages by name and version, npm talks to a registry website that implements the CommonJS Package Registry specification for reading package info.

npm is configured to use npm, Inc.'s public registry at <https://registry.npmjs.org> by default. Use of the npm public registry is subject to terms of use available at <https://www.npmjs.com/policies/terms>.

You can configure npm to use any compatible registry you like, and even run your own registry. Use of someone else's registry may be governed by their terms of use.

npm's package registry implementation supports several write APIs as well, to allow for publishing packages and managing user account information.

The npm public registry is powered by a CouchDB database, of which there is a public mirror at <https://skimdb.npmjs.com/registry>. The code for the couchapp is available at <https://github.com/npm/npm-registry-couchapp>.

The registry URL used is determined by the scope of the package (see **npm-scope**). If no scope is specified, the default registry is used, which is supplied by the **registry** config parameter. See **npm-config** , **npmrc** , and **npm-config** for more on managing npm's configuration.

Does npm send any information about me back to the registry?

Yes.

When making requests of the registry npm adds two headers with information about your environment:

- **Npm-Scope** – If your project is scoped, this header will contain its scope. In the future npm hopes to build registry features that use this information to allow you to customize your experience for your organization.
- **Npm-In-CI** – Set to “true” if npm believes this install is running in a continuous integration environment, “false” otherwise. This is detected by looking for the following environment variables: **CI** , **TDDIUM** , **JENKINS_URL** , **bamboo.buildKey** . If you'd like to learn more you may find the **original PR** interesting. This is used to gather better metrics on how npm is used by humans, versus build farms.

The npm registry does not try to correlate the information in these headers with any authenticated accounts that may be used in the same requests.

Can I run my own private registry?

Yes!

The easiest way is to replicate the couch database, and use the same (or similar) design doc to implement the APIs.

If you set up continuous replication from the official CouchDB, and then set your internal CouchDB as the registry config, then you'll be able to read any published packages, in addition to your private ones, and by default will only publish internally.

If you then want to publish a package for the whole world to see, you can simply override the `--registry` option for that `publish` command.

I don't want my package published in the official registry. It's private.

Set `"private": true` in your `package.json` to prevent it from being published at all, or `"publishConfig":`

`{"registry": "http://my-internal-registry.local"}` to force it to be published only to your internal registry.

See [package.json](#) for more info on what goes in the `package.json` file.

Will you replicate from my registry into the public one?

No. If you want things to be public, then publish them into the public registry using npm. What little security there is would be for nought otherwise.

Do I have to use couchdb to build a registry that npm can talk to?

No, but it's way easier. Basically, yes, you do, or you have to effectively implement the entire CouchDB API anyway.

Is there a website or something to see package docs and such? §

Yes, head over to <https://www.npmjs.com/>