feature overview:

npm Enterprise
Enterprise-grade JavaScript development

Join some of the world’s biggest companies in a preview of the platform for frictionless, safe JavaScript development.

www.npmjs.com/enterprise
Contents

1 Security

Authentication
Invitations—username/password
SSO—OAuth 2.0 OIDC
Custom registry URL/Single-tenant registry
Token management
Two-factor authentication
npm Audit

9 Collaboration

Organizations
Package pages
Package publication
registry.npmjs.org package proxy
Search and discovery

15 Support

Support offerings
Basic
Premium
Security
Safe code discovery and reuse

npm Enterprise provides visibility and control over how development teams share code.
Authentication

Invitations—username/password

By default, npm Enterprise is configured with username/password-based authentication. An administrator sends invitations to users within their organization to grant access to a private registry instance:

Quick testing: can preview the npm Enterprise experience without needing to configure a single-sign-on solution.
SSO—OAuth 2.0 OIDC

npm Enterprise supports logging in with OIDC-compliant OAuth 2.0 identity providers. These include Okta, Auth0, and the Google Identity Platform. SAML authentication is available upon request.

Once SSO is enabled, users logging into npm Enterprise are directed to an external identity provider, rather than prompted for a username and password:

✓ Eliminates the need for associating a username and password with npm Enterprise. From a security standpoint, this is one less password that can be hacked.

✓ A company can plug in an entire existing directory of employees with no need for an administrator to provision each account.
Custom registry URL / Single-tenant registry

Every npm Enterprise customer is given a custom registry URL for accessing their single-tenant npm registry.

- Fully segregates a company’s data on its own infrastructure.
- Provides a single URL for an OpSec team to allow through the corporate firewall.

npm Audit

npm Audit notifies developers about security vulnerabilities, both when new dependencies are added to JavaScript projects and when the command `npm audit` is explicitly run.
In addition, npm audit fix allows developers to automatically fix critical vulnerabilities within their JavaScript project:

```
Benjamins-MacBook-Pro:bencorp benjamincoe$ npm audit

=== npm audit security report ===

# Run npm install marked@0.5.1 to resolve 5 vulnerabilities

<table>
<thead>
<tr>
<th>High</th>
<th>Sanitization bypass using HTML Entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>marked</td>
</tr>
<tr>
<td>Dependency of</td>
<td>marked</td>
</tr>
<tr>
<td>Path</td>
<td>marked</td>
</tr>
<tr>
<td>More info</td>
<td><a href="https://nodesecurity.io/advisories/101">https://nodesecurity.io/advisories/101</a></td>
</tr>
</tbody>
</table>
```

```
In addition, npm audit fix allows developers to automatically fix critical vulnerabilities within their JavaScript project:

found 5 vulnerabilities (2 moderate, 3 high) in 2 scanned packages
run `npm audit fix` to fix 5 of them.
Benjamins-MacBook-Pro:bencorp benjamincoe$ npm audit fix
npm WARN @bencorp/bencorp@1.0.1 No repository field.
+ marked@0.5.1
updated 1 package in 1.628s
fixed 5 of 5 vulnerabilities in 2 scanned packages
Benjamins-MacBook-Pro:bencorp benjamincoe$ npm audit

=== npm audit security report ===

found 0 vulnerabilities
in 2 scanned packages
```

✓ Provides access to npm's community security feed.

✓ Provides a pre-publication feed of newly-discovered vulnerabilities.

✓ npm Audit is integrated directly into npm's command-line interface, providing a frictionless workflow to find and fix security vulnerabilities.

✓ Draws developers' attention to security vulnerabilities earlier in the application development cycle.
Token management

When logging into npm Enterprise from the npm command-line interface, a user will be directed to a website that allows them to mint a Read-Only or Read and Write token for the .npmrc file on their computer.

The credentials in this file are what the npm CLI uses when publishing and installing.

From the CLI a user can also use the command npm token create to generate a deploy token in their .npmrc. When running this command, a user can specify an additional --cidr argument, allowing them to generate a deploy token that can only publish npm modules from a specific CIDR range.

Benjamins-MacBook-Pro:npmrc benjaminco$ npm token create --cidr=0.0.0.0/9
npm password:

<table>
<thead>
<tr>
<th>token</th>
<th>45fc434d-ad04-412c-a3ca-3a62934acda6</th>
</tr>
</thead>
<tbody>
<tr>
<td>cidr_whitelist</td>
<td>0.0.0.0/9</td>
</tr>
<tr>
<td>readonly</td>
<td>false</td>
</tr>
<tr>
<td>created</td>
<td>2018-09-21T22:00:31.6562</td>
</tr>
</tbody>
</table>
✓ Web-based login allows npm Enterprise to support complex authentication strategies such as OAuth 2.0 OIDC and SAML.

✓ Authentication tokens make it easy to integrate npm Enterprise with continuous integration/continuous deployment processes.

✓ Limiting authentication tokens by CIDR range provides additional security for CI/CD systems: even if a token is leaked, packages can only be published from a specific IP address.

Two-factor authentication (2FA)

One-time password-based 2FA authentication can be enabled for both login and package publication. If 2FA is enabled for publication, users will be prompted for a one-time password from the CLI when publishing a package:

```
Benjamins-MBP-2:demo-publish benjamincoe$ npm publish

’demo-corp-2/demo-publish@1.0.0’

299B package.json

299B package.json

| name: @demo-corp-2/demo-publish |
| version: 1.0.0 |
| package size: 291 B |
| unpacked size: 299 B |
| shasum: 4d48683238c655da7b15d4f03f3e4f2b218e9546 |
| integrity: sha512-9+QI4MbüFWX/[...]SKKoNl+INoQua= |
| total files: 1 |

npm ERR! publish Failed PUT 401
There was an error while trying authentication due to OTP (One-Time-Password). The One-Time-Password is generated via applications like Authy or Google Authenticator, for more information see: https://docs.npmjs.com/getting-started/using-two-factor-authentication

Enter OTP: 3944
```
If 2FA is enabled for login, users will be prompted for an OTP token when logging in on the website:

✓ If 2FA is enabled on publication, stealing a developer's `.npmrc` file will not allow hackers to publish updates to packages.

✓ If 2FA is enabled on login, stealing a developer's username/password will not allow hackers to mint new access tokens on the website.
Collaboration
JavaScript development efficiency

Develop software faster with modular, reusable code.
Organizations

Organizations allow npm Enterprise users to create multiple namespaces for package permissions that model the structure of their company.

Users logging into the npm Enterprise instance can be assigned to specific organizations depending on their role within the company:

Multiple teams can be created within an organization.
✓ Creating an organization structure that matches how a company actually works makes it less likely that developers step on each other’s toes, e.g., by overwriting code that was owned by another development team.

✓ Teams allow for fine-grained access control, for example, when many people may install a package but only a few can update it.

Package pages

Package pages provide a high-level summary for developers regarding the packages they're considering using in an application.

Code’s README files are rendered, providing a description of a library and information about the APIs it exposes:

Testing Publication

This is an example README.

```javascript
const a = 99
const foo = (n) => {
  console.info('hello world')
}
```

Keywords

demo  publish

Meta information, such as the number of versions released, package collaborators, and dependency and dependent counts, is also displayed:
Package pages are only visible to users who are authenticated on the npm Enterprise website, and have read access through their organizational ACLs.

- Hosting documentation in a centrally accessible location makes code reuse and sharing easier.
- Metrics, such as dependency count and dependent count, make it easier for developers to evaluate the quality of the code they consider.

## Package publication

npm Enterprise users can publish any number of private packages to their instance using the npm CLI tool.

npm Enterprise allows private packages to be published using the namespace of any user created on the instance, e.g., `@bencoe/my-test-package`, and of any organization created on the instance, e.g., `@frontend/my-test-package`.

Packages published to npm Enterprise can also be made “public” to the enterprise: packages will be available for installation to anyone who is authenticated on the npm Enterprise instance, regardless of the organizations and teams to which they belong.
✓ Developers are able to share private packages within their organization using the exact same command-line tool they use for installing open-source JavaScript code.

✓ Developers are able to organize their private code across a multitude of private scopes—whatever best suits their company’s structure.

registry.npmjs.org package proxy

Packages from the public npm registry can be transparently installed through a private npm Enterprise instance.

✓ A company’s OpSec team only needs to allow a single URL through their corporate firewall to get the full benefit of the npm public registry and private packages.

✓ Developers can simply set the URL of their npm Enterprise registry as their default registry; no complex `.npmrc` configuration is necessary.

Search and discovery

From an npm Enterprise website, developers can search for packages published by other developers and teams. Various fields within the `package.json` are indexed for search: keywords, author information, and description.

Search results are returned for both private packages written within an organization and public packages on the npmjs.com registry.
An enterprise’s custom registry homepage displays a list of recently updated packages, providing a live stream of how users interact with the registry:

- Powerful search helps developers discover packages that have already been written, encouraging more code reuse throughout an organization.
- A live feed of published packages encourages developers to publish regularly, and to frequently check their npm Enterprise homepage to learn of new and newly updated code.
- Search differentiates npm Enterprise from other tools’ rudimentary discovery functionality.
Support
Options for every team

Back a performant, reliable development solution with support from the global leaders in modular JavaScript.
Support offerings

Basic

Basic support is provided to all npm Enterprise customers.

Customers are asked to categorize their requests into critical and regular priority inquiries, which are handled as follows:

- **Critical priority** requests are responded to within 8 business hours, and updates are made every business day.
- **Regular priority** requests are responded to within 16 business hours.

- Access to enterprise-grade support from the world’s leading experts in modular JavaScript development.
- A dedicated email address for request identification.

Premium

Customers may add a premium support offering for an additional monthly fee.

- **Critical priority** requests are responded to within 3 business hours, and updates are made every 2 business hours.
- **Regular priority** requests are responded to within 8 business hours.

- Faster response times than basic support.
- A dedicated email address and license identification to programmatically triage support requests.
- Dedicated resources for mission-critical development tooling.