



Basic NPM Commands

Installation

<code>npm install</code> or <code>npm i</code>	Installs all dependencies defined in <code>package.json</code> in the current directory.	<pre>npm install</pre>
<code>npm install <package_name></code>	Installs a specific package and adds it as a dependency to <code>package.json</code> .	<pre>npm install lodash</pre>
<code>npm install <package_name> --save-dev</code> or <code>-D</code>	Installs a package as a development dependency. Useful for tools like testing frameworks.	<pre>npm install jest --save-dev</pre>
<code>npm install <package_name> --global</code> or <code>-g</code>	Installs a package globally, making it available in the system's PATH. Often used for command-line tools.	<pre>npm install create-react-app --global</pre>
<code>npm install <git_repo></code>	Install a package directly from a git repository.	<pre>npm install git+ssh://git@github.com:user/repo.git</pre>
<code>npm ci</code>	Install dependencies with a clean slate, based on <code>package-lock.json</code> . Faster and ensures consistency across environments.	<pre>npm ci</pre>

Updating Packages

<code>npm update</code>	Updates all packages listed in <code>package.json</code> to the latest version that satisfies the version ranges specified.	<pre>npm update</pre>
<code>npm update <package_name></code>	Updates a specific package to the latest version that satisfies its version range.	<pre>npm update lodash</pre>
<code>npm audit fix</code>	Attempts to automatically fix known security vulnerabilities in your dependencies.	<pre>npm audit fix</pre>
<code>npm outdated</code>	Lists outdated packages.	<pre>npm outdated</pre>

Uninstalling Packages

<code>npm uninstall <package_name></code>	Uninstalls a package from the <code>node_modules</code> directory and removes it from <code>package.json</code> dependencies.	<pre>npm uninstall lodash</pre>
<code>npm uninstall <package_name> --save-dev</code>	Uninstalls a package from development dependencies.	<pre>npm uninstall jest --save-dev</pre>
<code>npm uninstall -g <package_name></code>	Uninstalls a globally installed package.	<pre>npm uninstall -g create-react-app</pre>
<code>npm clean-install</code>	Remove <code>node_modules</code> and install dependencies.	<pre>npm clean-install</pre>

Managing Dependencies

Package Information

<code>npm view <package_name></code>	Displays detailed information about a package from the npm registry.	<pre>npm view lodash</pre>
<code>npm view <package_name> version</code>	Shows the latest version of a specific package.	<pre>npm view lodash version</pre>

<code>npm view <package_name> dependencies</code>	Lists the dependencies of a specific package. <code>npm view lodash dependencies</code>
<code>npm ls</code>	Lists installed packages and their dependencies in a tree structure. <code>npm ls</code>

Working with package.json

<code>npm init</code>	Guides you through creating a new <code>package.json</code> file for your project. <code>npm init</code>
<code>npm init -y</code> or <code>npm init --yes</code>	Creates a <code>package.json</code> file with default values without prompting for input. <code>npm init -y</code>
<code>npm install --production</code>	Installs only production dependencies (excluding devDependencies) from <code>package.json</code> . <code>npm install --production</code>
<code>npm prune --production</code>	Removes extraneous packages. If the <code>--production</code> flag is set, only removes the packages that are not listed in <code>dependencies</code> . <code>npm prune --production</code>

Scripts

<code>npm run <script_name></code>	Executes a script defined in the <code>scripts</code> section of <code>package.json</code> . <code>npm run start</code>
<code>npm start</code>	Runs the <code>start</code> script defined in <code>package.json</code> . Often used to start the application. <code>npm start</code>
<code>npm test</code>	Runs the <code>test</code> script defined in <code>package.json</code> . Used for running tests. <code>npm test</code>
<code>npm stop</code>	Runs the <code>stop</code> script defined in <code>package.json</code> . Used for stopping the application. <code>npm stop</code>
<code>npm restart</code>	Runs the <code>restart</code> script defined in <code>package.json</code> . Used for restarting the application. <code>npm restart</code>

Advanced NPM Usage

Publishing Packages

<code>npm login</code>	Logs you into the npm registry using your credentials. <code>npm login</code>
<code>npm publish</code>	Publishes the package in the current directory to the npm registry. Make sure to increment the version number in <code>package.json</code> before publishing. <code>npm publish</code>
<code>npm unpublish <package_name>@<version></code>	Removes a published package from the npm registry. Use with caution! <code>npm unpublish my-package@1.0.0</code>
<code>npm owner add <user> <package_name></code>	Adds a user as an owner of a package, allowing them to manage the package. <code>npm owner add username my-package</code>

Managing Cache

<code>npm cache clean --force</code>	Clears the npm cache. Use with caution, as it can sometimes resolve installation issues. <code>npm cache clean --force</code>
<code>npm cache verify</code>	Verifies the integrity of the cache and cleans up corrupted data. <code>npm cache verify</code>
<code>npm config get cache</code>	Gets the cache location. <code>npm config get cache</code>

NPM Configuration

<code>npm config list</code>	Displays the current npm configuration. <code>npm config list</code>
<code>npm config get <key></code>	Retrieves the value of a specific configuration key. <code>npm config get registry</code>
<code>npm config set <key> <value></code>	Sets a configuration key to a specific value. <code>npm config set registry https://registry.npmjs.org/</code>
<code>npm config delete <key></code>	Deletes a configuration key. <code>npm config delete registry</code>

NPM Versioning and Security

Semantic Versioning

NPM uses semantic versioning (semver) to manage package versions.

`MAJOR.MINOR.PATCH`

- MAJOR:** Incompatible API changes
- MINOR:** Add functionality in a backwards compatible manner
- PATCH:** Backwards compatible bug fixes

Examples of version ranges:

- `1.2.3` : Exact version
- `^1.2.3` : Compatible with 1.x.x, but not 2.0.0
- `~1.2.3` : Compatible with 1.2.x, but not 1.3.0
- `>` : Greater than
- `<` : Less than
- `>=` : Greater than or equal to
- `<=` : Less than or equal to
- `*` : Any version

Security Audits

<code>npm audit</code>	Analyzes your project's dependencies for security vulnerabilities and provides a report. <code>npm audit</code>
<code>npm audit fix</code>	Attempts to automatically fix the detected security vulnerabilities by updating to secure versions of the dependencies. <code>npm audit fix</code>
<code>npm audit fix --force</code>	Attempts to automatically fix the detected security vulnerabilities by updating to secure versions of the dependencies with force. <code>npm audit fix --force</code>

Bumping Versions

<code>npm version <new_version></code>	Sets the package version to a specific version number. <code>npm version 2.0.0</code>
<code>npm version major</code>	Increments the major version number. <code>npm version major</code>
<code>npm version minor</code>	Increments the minor version number. <code>npm version minor</code>
<code>npm version patch</code>	Increments the patch version number. <code>npm version patch</code>
<code>npm version prerelease</code>	Increments the pre-release version number. <code>npm version prerelease</code>