

pnpm Cheatsheet

By Dejan Panovski • Updated on Sep 11, 2026 •

Quick reference for pnpm commands covering installation, dependencies, lockfiles, scripts, workspace filtering, the package store, and Node.js runtimes

pnpm is a Node.js package manager that keeps package files in a shared content-addressable store and links them into each project. This cheatsheet covers installing pnpm and Node.js runtimes, adding and updating dependencies, running scripts, working across a monorepo with filters, and maintaining the store.

Install and Update pnpm

Set up the pnpm executable and keep it current.

<code>curl -fsSL https://get.pnpm.io/install.sh sh -</code>	Install with the standalone script
<code>wget -qO- https://get.pnpm.io/install.sh sh -</code>	Install without curl
<code>npm get-pnpm</code>	Install through npm
<code>pnpm self-update</code>	Update pnpm to the current release
<code>pnpm self-update 12.3.4</code>	Move to a specific pnpm version
<code>pnpm setup</code>	Create PNPM_HOME and update the shell profile
<code>pnpm --version</code>	Print the installed pnpm version

Node.js Runtimes

Install language runtimes with pnpm itself.

<code>pnpm runtime set node lts -g</code>	Install the Node.js LTS release
<code>pnpm runtime set node 22 -g</code>	Install a Node.js major version
<code>pnpm runtime set node latest -g</code>	Install the newest Node.js release
<code>pnpm runtime set node 22.23.2 -g</code>	Install an exact Node.js version
<code>pnpm runtime set deno 2 -g</code>	Install Deno
<code>pnpm runtime set bun latest -g</code>	Install Bun

Project Setup

Create and edit project metadata.

pnpm init	Create package.json
<code>pnpm init --init-package-manager</code>	Create package.json and pin pnpm
<code>pnpm init --bare</code>	Create package.json with required fields only
<code>pnpm import</code>	Build pnpm-lock.yaml from another lockfile
<code>pnpm pkg get name</code>	Print a package.json field
<code>pnpm pkg set scripts.start="node index.js"</code>	Add or update a field
<code>pnpm pkg delete scripts.old</code>	Remove a package.json field

Add Dependencies

Record packages in package.json and the lockfile.

pnpm add express	Add a runtime dependency
<code>pnpm add -D eslint</code>	Add a development dependency
<code>pnpm add -O sharp</code>	Add an optional dependency
<code>pnpm add -E express</code>	Add without a version range
<code>pnpm add express@5.1.0</code>	Add a specific version
<code>pnpm add -g http-server</code>	Add a global package
<code>pnpm add -w typescript</code>	Add to the workspace root

Install Dependencies

Reproduce a dependency tree from the lockfile.

<code>pnpm install</code>	Install everything package.json declares
pnpm install --frozen-lockfile	Install without changing the lockfile
<code>pnpm ci</code>	Clean install for CI pipelines
<code>pnpm install --prod</code>	Skip development dependencies
<code>pnpm install --offline</code>	Install from the store only
<code>pnpm install --ignore-scripts</code>	Install without lifecycle scripts
<code>pnpm fetch</code>	Fetch the lockfile contents into the store

Update and Inspect

Check versions and trace where packages come from.

<code>pnpm outdated</code>	Show packages with newer versions
<code>pnpm update</code>	Update within package.json ranges
<code>pnpm update express --latest</code>	Update one package and its range
<code>pnpm update -i</code>	Choose updates interactively
<code>pnpm list --depth 0</code>	List direct dependencies
<code>pnpm why express</code>	Explain why a package is installed
<code>pnpm licenses list</code>	List dependency licenses

Remove and Clean

Drop packages and reclaim project space.

<code>pnpm remove express</code>	Remove a dependency
<code>pnpm remove -g http-server</code>	Remove a global package
<code>pnpm prune</code>	Remove extraneous packages
<code>pnpm prune --prod</code>	Remove development dependencies
<code>pnpm dedupe</code>	Collapse duplicate versions in the lockfile
<code>pnpm dedupe --check</code>	Report duplicates without writing
<code>pnpm pm clean</code>	Remove node_modules from every workspace project

Scripts and Execution

Run package scripts and one-off tools.

<code>pnpm run build</code>	Run a named script
<code>pnpm start</code>	Run the start script
<code>pnpm test</code>	Run the test script
<code>pnpm run build -- --watch</code>	Pass arguments to a script
<code>pnpm exec eslint .</code>	Run a local package binary
<code>pnpm dlx create-vite@latest app</code>	Run a package without installing it
<code>pnpm pm <command></code>	Force a built-in command over a same-named script

Workspaces and Filtering

Target one package or a slice of a monorepo.

<code>pnpm -r run build</code>	Run a script in every workspace package
<code>pnpm --filter web-app add zod</code>	Add a dependency to one package
<code>pnpm --filter web-app run test</code>	Run a script in one package
<code>pnpm --filter "@scope/*" test</code>	Select packages by pattern
<code>pnpm --filter web-app... build</code>	Select a package and its dependencies
<code>pnpm --filter ...web-app build</code>	Select a package and its dependents
<code>pnpm --filter "...[origin/main]" build</code>	Select packages changed since a branch

Store and Cache

Maintain the shared content-addressable store.

<code>pnpm store path</code>	Print the store location
<code>pnpm store status</code>	Check the store for modified packages
<code>pnpm store prune</code>	Remove orphaned packages from the store
<code>pnpm store add express@5.1.0</code>	Add a package to the store
<code>pnpm cache path</code>	Print the metadata cache directory
<code>pnpm root -g</code>	Print the global node_modules path
<code>pnpm bin -g</code>	Print the global binary directory

Security and Diagnostics

Audit dependencies and check the installation.

<code>pnpm audit</code>	Scan dependencies for advisories
<code>pnpm audit --prod</code>	Audit production dependencies only
<code>pnpm audit --fix</code>	Add overrides for non-vulnerable versions
<code>pnpm doctor</code>	Run diagnostics on the pnpm environment
<code>pnpm sbom --sbom-format spdx</code>	Generate a software bill of materials
<code>pnpm approve-builds</code>	Approve dependency build scripts
<code>pnpm ignored-builds</code>	List packages with blocked build scripts

Related Guides

Use these references for Node.js package workflows.

pnpm Command	Full pnpm tutorial with examples
npm Command	The npm equivalents of these commands
Install Node.js on Ubuntu 26.04	Install Node.js and npm