Turborepo Monorepo task runner

turbo.json task graph → hash inputs → **Local cache** .turbo/cache → miss → **Remote cache** shared → miss → **Run task** store outputs

CONCEPTS

Turborepo hashes a task's inputs; an unchanged hash replays the cached outputs and logs instead of running.

Tasks live in turbo.json; package.json scripts of the same name are what actually runs.

RUN TASKS

`turbo run <task>`
Run a task in all packages

`turbo run <task> <task>`
Run several tasks

`turbo run <pkg>#<task>`
Run one task in one package

`turbo run <task> --force`
Ignore cache, re-run everything

`turbo run <task> --dry`
Show what would run

`turbo run <task> --dry=json`
Dry run as JSON

`turbo run <task> --only`
Skip dependency tasks

`turbo run <task> --continue`
Keep going after a failure

RUN TASKS (cont.)

`turbo run <task> --concurrency <n>`
Cap parallel tasks (number or %)

`turbo run <task> --ui <stream|tui>`
Pick terminal UI

`turbo run <task> --output-logs <mode>`
Set log verbosity

`turbo watch <task>`
Re-run tasks on file changes

FILTER

`--filter=<pkg>`
Select a package by name

`-F <pkg>`
Short form of --filter

`--filter="./apps/*"`
Select by directory glob

`--filter=<pkg>...`
Package and its dependencies

`--filter=...<pkg>`
Package and its dependents

`--filter="[main]"`
Packages changed since a git ref

`--filter="...[HEAD^1]"`
Changed packages plus dependents

FILTER (cont.)

`--filter=!<pkg>`
Exclude a package

`--filter=<a> --filter=<b>`
Union of selections

`--affected`
Changed vs main, plus dependents

`turbo ls`
List packages in the repo

CACHE

`turbo login`
Log in to the remote cache

`turbo link`
Link repo to a remote cache

`turbo unlink`
Disconnect from remote cache

`turbo logout`
Log out of the provider

`turbo link --scope <team>`
Link to a given team

`--cache=local:rw,remote:r`
Set cache read/write access

`--token <token> --team <slug>`
Remote cache credentials for CI

REPO TOOLS

`turbo prune <pkg>`
Create a partial monorepo in out/

`turbo prune <pkg> --docker`
Docker-layer friendly output

`turbo generate`
Run a custom generator

`turbo generate workspace`
Scaffold a new package or app

`turbo gen workspace --copy <name>`
Copy an existing workspace

`turbo boundaries`
Check package boundary rules

`npx create-turbo@latest`
Start a new monorepo

TURBO.JSON

`"tasks": { ... }`
Define runnable tasks

`"dependsOn": ["^build"]`
Run dependencies' build first

`"dependsOn": ["lint"]`
Run same-package task first

`"outputs": ["dist/**"]`
Files to cache after success

TURBO.JSON (cont.)

`"inputs": ["src/**"]`
Files that affect the hash

`"cache": false`
Never cache this task

`"persistent": true`
Long-running task such as dev

`"env": ["API_URL"]`
Env vars hashed into the task

`"globalEnv": ["CI"]`
Env vars hashed into every task

`"globalDependencies": [".env"]`
Files that invalidate all caches

`"extends": ["//"]`
Inherit root config in a package

Dependency syntax

Entry	Runs first
`build`	build in same package
`^build`	build in dependencies
`app#build`	build in package app

RECIPES

Remote cache in CI

```
export TURBO_TOKEN=<token>
export TURBO_TEAM=<team>
turbo run build
```