Turborepo Monorepo task runner

CONCEPTS

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

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

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

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

`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 (cont.)

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

`--affected`
Changed vs main, plus dependents

CACHE

`turbo login`
Log in to the remote cache

`turbo link`
Link repo to a remote cache

`--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 workspace`
Scaffold a new package or app

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

`"cache": false`
Never cache this task

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

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