Bun JS runtime and toolkit

CONCEPTS

> Bun is one binary: runtime, package manager, test runner and bundler. It runs TypeScript and JSX directly, no build step.

> Put Bun flags right after bun: 'bun --watch run dev'. After the script name they go to the script.

> Dependency lifecycle scripts (postinstall) are blocked unless the package is in trustedDependencies.

INSTALL AND ADD

bun install
Install all dependencies

bun add *<pkg>*
Add a dependency

bun add *<pkg>*@*<version>*
Add a specific version or tag

bun add -d *<pkg>*
Add a dev dependency

bun add -g *<pkg>*
Install a CLI tool globally

bun install --frozen-lockfile
Fail if lockfile is out of date

bun ci
Same as install --frozen-lockfile

bun install --production
Skip devDependencies

REMOVE AND UPDATE

bun remove *<pkg>*
Remove a dependency

bun update
Update within package.json ranges

bun update --latest
Ignore ranges, take latest

bun update -i
Pick updates interactively

REMOVE AND UPDATE (cont.)

bun outdated
Show packages with newer versions

bun audit
Check for known vulnerabilities

RUN AND WATCH

bun *<file>*
Run a TS, JS or JSX file

bun run *<script>*
Run a package.json script

bun --watch *<file>*
Restart on file change

bunx *<pkg>*
Run a package binary without installing

TEST

bun test
Run all tests

bun test *<path>*
Run files matching a path filter

bun test -t '*<regex>***'**
Run tests by name pattern

bun test --watch
Re-run on file change

bun test --coverage
Print a coverage report

BUILD AND SCAFFOLD

bun init -y
Scaffold with default answers

bun build *<entry>* **--outdir** *<dir>*
Bundle to a directory

bun build *<entry>* **--target** *<env>*
Target browser, bun or node

WORKSPACES AND PACKAGES

bun run --filter '*<pattern>***'** *<script>*
Run a script in matching workspaces