

Rules & Repo Config

A cheat sheet. Write team context once — the AI picks it up on every session.

01 Always Apply

Activates: every chat session, unconditionally.

Use for: team-specific standards — TypeScript, bun, vitest.

```
description: "Project conventions"
alwaysApply: true
```

- Use TypeScript for code
- Use bun for package management
- Use vitest for testing

Note: every team mate consumes context on every chat. Keep short.

02 Apply Intelligently

Activates: when Agent reads the description and judges it relevant.

Use for: domain-specific guidance that only matters in some tasks.

```
description: "Standards for React components and Tailwind styling"
alwaysApply: false
```

- Tailwind CSS, no inline styles
- PascalCase filenames, named exports

Note: without a description, the rule silently does nothing.

03 File-Scoped (glob)

Activates: when files matching the glob are part of the conversation.

Use for: language- or directory-specific guidance. Narrowest scope.

```
description: "API endpoint validation"
alwaysApply: false
globs: ["src/api/**/*.ts"]
```

- Use zod for request/response validation
- Export inferred TypeScript types

Note: globs match on @-mention or unrelated tools.

04 Manual (@-mention)

Activates: when this matching the glob are part of the conversation, you pull in by name.

Use for: step-by-step workflows and runbooks you pull in on demand.

```
description: "Template for a new Express service"
```

1. Copy @express-service-template.ts
2. RESTful routes, no NPL-style
3. Include error-handling middleware

Note: invoke with @rule-name in Agent chat. Think: saved playbook.

AGENTS.md vs .cursor/rules/

A two-question flow to pick the right home.

Need to work across multiple AI tools? → **AGENTS.md**

Plain Markdown at repo root, tool-agnostic (Cursor, Claude Code, Codex, Copilot).

Need conditional activation? → **.cursor/rules/**

.mdc files with frontmatter. Glob-scoped, agent-decided, or manual triggers.

Want both? Use them together — they merge.

Best practices

Start simple, earn your rules.

When to add one: Agent makes the same mistake twice — write the correction as a rule, not before.

Keep them: under 500 lines, composable, with concrete examples. Reference @filename.ts rather than copying code that will go stale.

Commit them: both AGENTS.md and .cursor/rules/ go in git. The whole team gets them on clone.

Hierarchy: team rules (Enterprise) > project rules (.cursor/rules/) > user rules (your prefs). Higher tier wins on conflict.

Rules apply to Agent chat only. They don't affect Inline Edit (Cmd+K) or Cursor Tab. Create rules fast with `/create-rule <your instruction>` in Agent chat — it drafts the .mdc file for you.