

# find-skills

## *Installation Guide*

*The meta-skill that teaches your agent to find every other skill*

Part of vercel-labs/skills • npx skills

# Contents

---

|                                                |          |
|------------------------------------------------|----------|
| <b>Contents</b>                                | <b>2</b> |
| <b>1. What find-skills Does</b>                | <b>3</b> |
| <b>2. Why It Is Super Important</b>            | <b>3</b> |
| It stops your agent from reinventing the wheel | 3        |
| It builds in quality control                   | 3        |
| It is the gateway to the whole ecosystem       | 3        |
| <b>3. Prerequisites</b>                        | <b>4</b> |
| <b>4. Installing find-skills</b>               | <b>4</b> |
| Recommended: global install                    | 4        |
| Alternative: project-only install              | 4        |
| Target a specific agent                        | 4        |
| Non-interactive (CI / scripted)                | 4        |
| Flag reference                                 | 4        |
| <b>5. Verify the Installation</b>              | <b>4</b> |
| <b>6. Putting It to Work</b>                   | <b>5</b> |
| <b>7. Troubleshooting</b>                      | <b>5</b> |
| <b>Further Resources</b>                       | <b>5</b> |

## 1. What find-skills Does

**find-skills** is a *meta-skill* — a skill whose only job is to help your agent discover and install other skills. Once it is in place, whenever you ask your agent something like “*how do I do X,*” “*is there a skill for X,*” or you simply describe a task, the agent knows to search the open skills ecosystem before trying to solve the problem from scratch.

Concretely, the skill teaches the agent to:

- Recognize when a request maps to an existing, installable skill.
- Check the `skills.sh` leaderboard first for battle-tested, popular options.
- Run `npx skills find [query]` with sensible keywords when the leaderboard doesn’t cover the need.
- Vet candidates by install count, source reputation, and GitHub stars before recommending anything.
- Install the chosen skill for you with `npx skills add <package>`.

## 2. Why It Is Super Important

Most coding agents don’t know what they don’t know. The ecosystem has hundreds of skills covering React performance, PR reviews, deployments, testing, accessibility audits, changelog generation, and far more — but an agent will never use any of them if it doesn’t realize they exist. **find-skills** closes that gap. It turns “I’ll write something from scratch” into “let me check whether a vetted skill already solves this.”

### It stops your agent from reinventing the wheel

Without it, your agent hand-rolls solutions to problems that the community has already solved, tested, and tuned. A request like “make my React app faster” produces ad-hoc advice instead of pulling in a skill with 40+ Vercel-Engineering performance rules. The meta-skill routes you to the proven artifact.

### It builds in quality control

Discovery without judgment is dangerous — anyone can publish a skill. **find-skills** bakes in guardrails so the agent doesn’t install low-quality or untrustworthy packages:

| Signal            | Rule of Thumb                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------|
| Install count     | Prefer skills with <b>1K+ installs</b> ; be cautious below 100.                                      |
| Source reputation | Official sources ( <i>vercel-labs</i> , <i>anthropics</i> , <i>microsoft</i> ) over unknown authors. |
| GitHub stars      | Treat a skill from a repo with under <b>100 stars</b> with skepticism.                               |

### It is the gateway to the whole ecosystem

Think of **find-skills** as **the package manager’s search bar, living inside your agent**. Install this one skill and your agent gains an evergreen path to every other skill — including ones

published after today. That is why it is often the very first skill people install: it makes all the others discoverable.

**In short:** find-skills is high-leverage because it is multiplicative. Every other skill becomes reachable, and reaches you with quality checks already applied.

### 3. Prerequisites

- Node.js with `npx` available (bundled with npm). Nothing needs to be globally installed.
- A supported coding agent — e.g. Claude Code, Cursor, Codex, OpenCode. The CLI auto-detects what you have.
- Write access to wherever you're installing (your project folder, or your home directory for a global install).

### 4. Installing find-skills

#### Recommended: global install

Because find-skills is useful in every project, install it globally so every agent on your machine can reach it:

```
npx skills add vercel-labs/skills --skill find-skills -g
```

**Heads up:** The find-skills skill lives in the `vercel-labs/skills` CLI repo itself — not in the `vercel-labs/agent-skills` content repo. Point the `--skill` flag at `vercel-labs/skills` as shown above.

#### Alternative: project-only install

To scope it to the current project (so it's committed and shared with your team), drop the `-g` flag:

```
npx skills add vercel-labs/skills --skill find-skills
```

#### Target a specific agent

If the CLI doesn't auto-detect your agent, or you want to be explicit, name it with `-a`:

```
npx skills add vercel-labs/skills --skill find-skills -g -a claude-code
```

#### Non-interactive (CI / scripted)

Add `-y` to skip every confirmation prompt:

```
npx skills add vercel-labs/skills --skill find-skills -g -a claude-code -y
```

#### Flag reference

| Flag                                  | Description                                       |
|---------------------------------------|---------------------------------------------------|
| <code>--skill find-skills</code>      | Selects just this one skill from the repo         |
| <code>-g, --global</code>             | Installs to your user directory (all projects)    |
| <code>-a, --agent &lt;name&gt;</code> | Targets a specific agent, e.g. <i>claude-code</i> |

| Flag      | Description                |
|-----------|----------------------------|
| -y, --yes | Skips confirmation prompts |

## 5. Verify the Installation

Confirm the skill landed where you expect:

```
npx skills list          # all installed skills
npx skills ls -g         # global skills only
```

You should see `find-skills` in the output. For Claude Code, a global install lands in `~/.claude/skills/`; a project install lands in `.claude/skills/`. Other agents use their own paths.

## 6. Putting It to Work

Once installed, you don't invoke `find-skills` directly — your agent activates it automatically when a request looks like it could be served by a skill. Just talk to your agent naturally:

- “How do I make my React app faster?” → agent searches `react performance`.
- “Can you help me with PR reviews?” → agent searches `pr review`.
- “I need to create a changelog.” → agent searches `changelog`.

You can also drive the underlying commands yourself at any time:

```
npx skills find          # interactive, fzf-style search
npx skills find react performance
npx skills find react --owner vercel # scope to an owner
npx skills add vercel-labs/agent-skills@react-best-practices
```

## 7. Troubleshooting

| Symptom                           | Fix                                                                                                           |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------|
| “No skills found” when installing | Double-check the source is <code>vercel-labs/skills</code> and the flag is <code>--skill find-skills</code> . |
| Skill not activating in the agent | Confirm it installed to that agent's skills path; restart the agent session if needed.                        |
| Permission errors                 | Use the project scope if you can't write to <code>~</code> , or fix directory permissions.                    |
| Agent not detected                | Name it explicitly with <code>-a &lt;agent&gt;</code> during install.                                         |

## Further Resources

- Skills directory & leaderboard: [skills.sh](#)
- CLI repository: [github.com/vercel-labs/skills](https://github.com/vercel-labs/skills)
- Official skill collection: [github.com/vercel-labs/agent-skills](https://github.com/vercel-labs/agent-skills)

Licensed MIT.