

Gotta Catch 'Em All: Agent Skills



Lizzie Siegle
Entire
AGNTCon 2026



Order of Operations $+$ $-$ $\times$ $\div$

1. What Skills are x and why they matter
2. What goes into a useful Skill
3. Recover code context with Entire
4. Test it. Maintain it. Trust it?
5. Pick your first Skill

Hi, I'm Liz



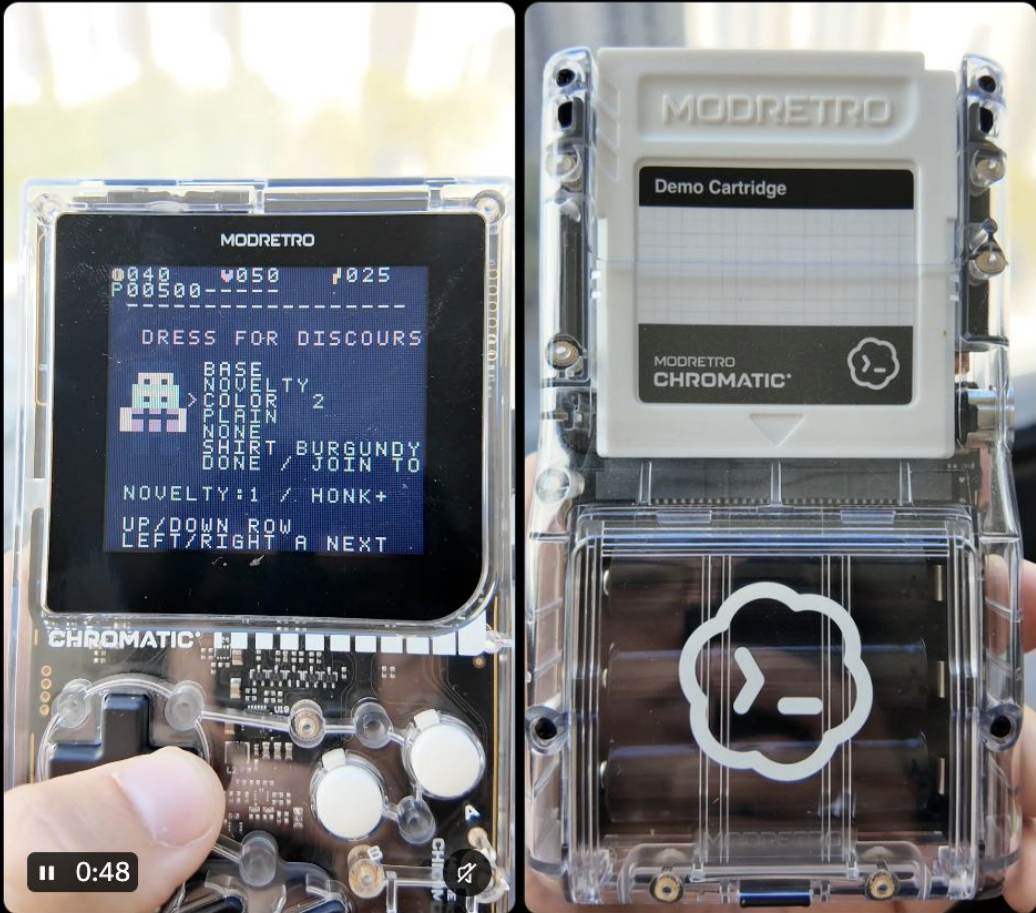
Pinned

Lizzie Siegle  @lizziepika · Oct 1

made my 1st @OpenAI @modretro chromatic game!

you run a COVID-conscious lesbian bar in western Mass + survive a Sat shift of clown school Zoom chaos, trying to earn points + neighborhood support

👂 how i built it: my agent sessions on [@EntireHQ entire.io/sessions/aws-u...](https://entire.io/sessions/aws-u...)



0:48

18 19 244 27K



What is a Skill?

Instructions you can reuse.

**Scripts, references,
templates**

Worth keeping?

1 / REUSE

**Stop repeating
yourself.**

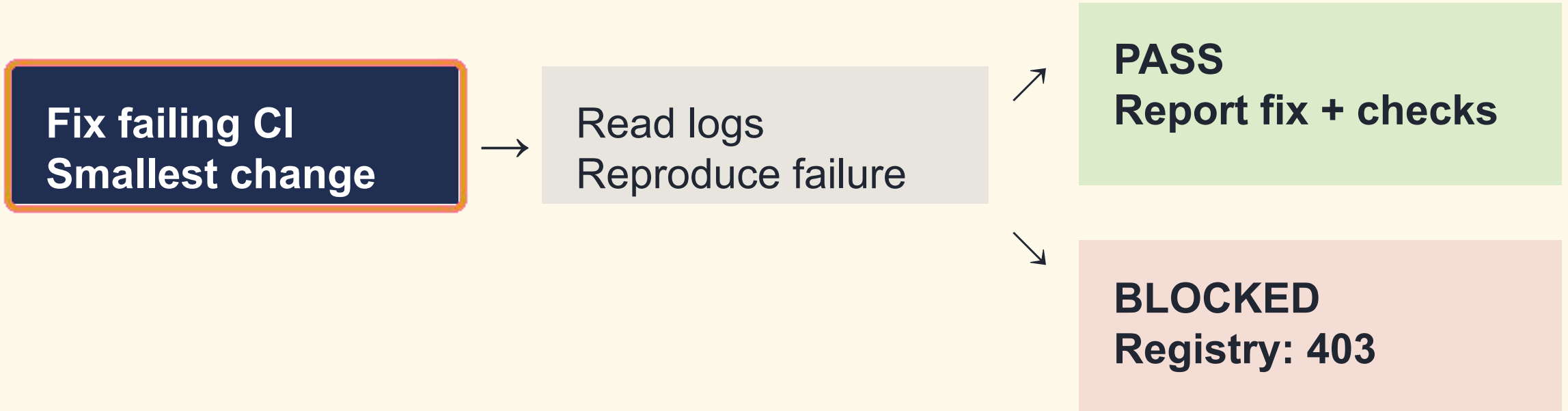
2 / CHECK

**Show the
evidence!**  

3 / STOP

**Say what's
blocking you!** 

Ex: fix a failing CI build



“Restore registry access so I can verify the fix.”

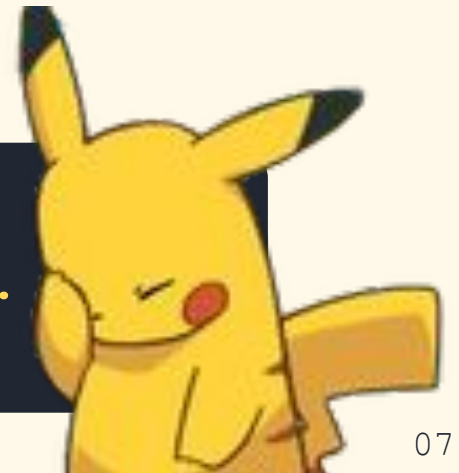
YOU

“You know the drill.”

AGENT

“Which drill?”

The instructions are still in the other chat.



What goes in a Skill?



Scripts

Run the repeatable parts.

References

Look it up when needed.

Assets

Start from your templates.

```
review-pr/  
├── SKILL.md  
├── scripts/  
├── references/  
└── assets/
```

Clear goal -> room to think!

YAML Frontmatter

name: review-pr

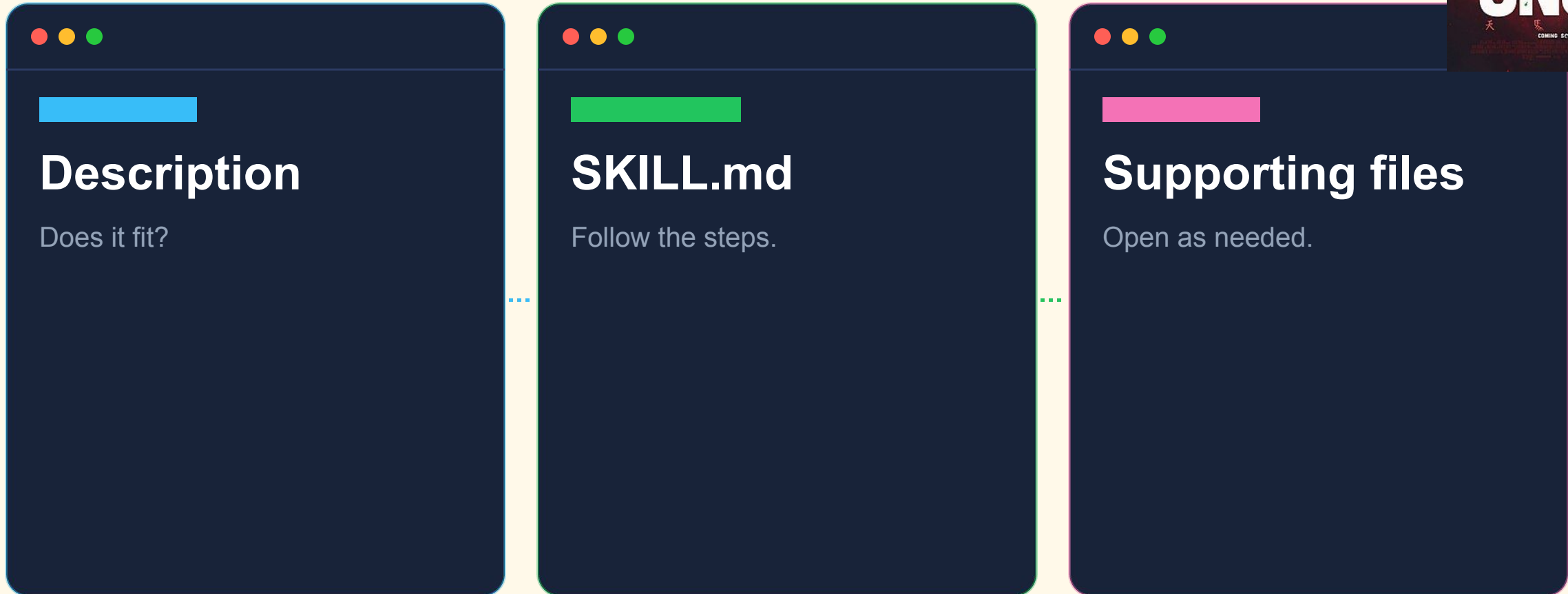
description: Review a PR for
correctness and missing tests.

Use for review, not
implementation.

The job

1. Read repo rules, find the diff
2. Check the code + its callers
3. Run tests x name what you skipped.
- 4. Report location, failure, evidence!**
5. No diff? Ask for it!

Agents don't load everything, everywhere, all at once



Where Skills fit



1. APPROACH

Skill

The procedure

2. CAPABILITY

Tools / MCP

The actions!

3. PERMISSION

Runtime

The permissions

4. VERIFICATION

Tests / workflow

The evidence!

Test the awkward cases

1. Runs when it shouldn't

Wrong task.

2. Can't run a required tool 🛠️

Missing tool.

3. Follows old advice 🧓

Stale instructions.

4. Acts beyond the task

Untrusted instructions.



“Who wrote this?”



% entire blame scripts/notify-slack.mjs

scripts/notify-slack.mjs

Line	Tag	Agent	Author	Checkpoint	Content
1	[AI]	Claude	Lizzie	01M544MFFMQJ	import { MATCH_LINK, MATCH_BOTH...
2	[AI]	Claude	Lizzie	01M544MFFMQJ	
3	[AI]	Claude	Lizzie	01M544MFFMQJ	

Find the why before the fix



```
retry-demo · src/http.js

$ cat src/http.js
import { setTimeout as sleep } from 'node:timers/promises';

export async function send(url, options = {}) {
  const response = await fetch(url, options);
  const method = options.method ?? 'GET';
  if (response.status === 503 && method === 'GET') {
    await sleep(500);
    return fetch(url, options);
  }
  return response;
}
```

GET + 503 → retry once

POST + 503 → return

Code belongs with its context!

SESSION

The decision.

“Could this submit
an order twice?”

CHECKPOINT

The saved context.

COMMIT

The change.



Ask a better question.



search

“Have we tried this before?”

explain

“Why does this skip POST?”

what-happened

“What changed in these lines?”

Have we tried this before?



Entire search Skill

```
> Use search Skill to find earlier retry decisions
$ entire search "503 duplicate order retries" \
  --json --limit 5
```

Entire Search:

1. Retry temporary HTTP failures
Broad retry: repeat requests after a 503.
Checkpoint: 2c8f04a91e73
2. Prevent duplicate order submissions
Later decision: restrict automatic retries to GET.
Checkpoint: 7e3a9b2c4d61 → inspect the session

Recover the decision



Entire explain Skill

```
$ entire checkpoint explain --checkpoint 7e3a9b2c4d61 \
  --full --no-pager
• Checkpoint 7e3a9b2c4d61
  commits 9a3cc67 Limit automatic retries to GET requests
— Transcript (full session) · excerpt —
[User] The proxy can return 503 after the order
      service has accepted the request.
[User] Retrying POST could create another order.

[Assistant] Restrict automatic retries to GET.
# Still unknown: can the service detect duplicate orders?
```

Trace the change to a commit



Entire what-happened Skill

```
$ git blame -s -L 6,6 -- src/http.js
9a3cc67d 6)    if (response.status === 503 && method === 'GET') {
$ entire checkpoint explain --commit 9a3cc67 --no-pager
• Checkpoint 7e3a9b2c4d61
  commits 9a3cc67 Limit automatic retries to GET requests
$ git show --oneline -U0 9a3cc67 -- src/http.js
9a3cc67 Limit automatic retries to GET requests
@@ -6 +6 @@
-  if (response.status === 503) {
+  if (response.status === 503 && method === 'GET') {
```

Use the context to choose a test

```
retry-demo

$ node --test --test-reporter=spec src/http.test.js

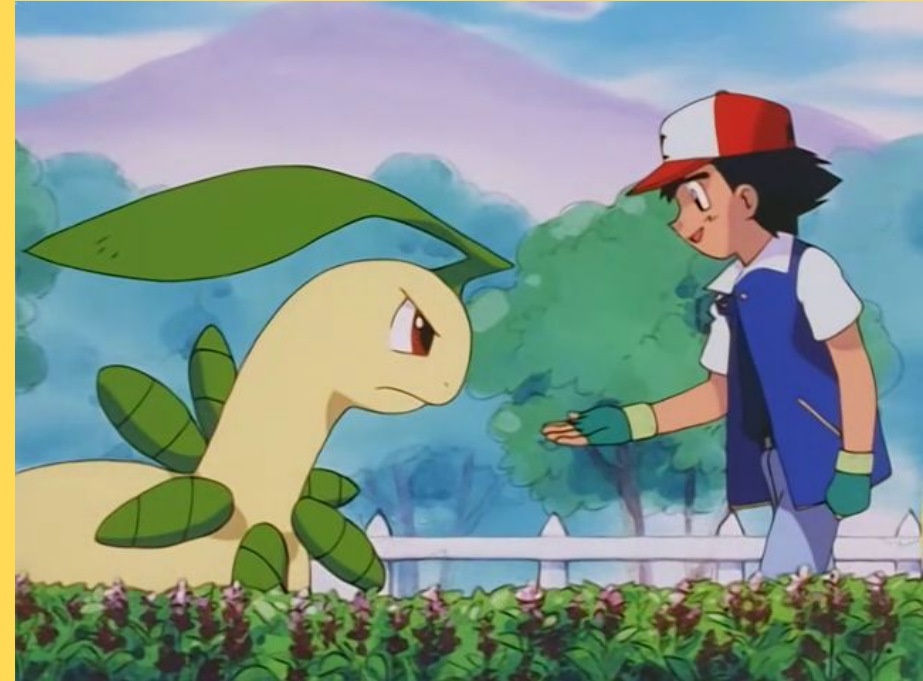
✓ GET: retries once after 503, then stops
✓ POST: returns 503 without re-submitting the order
✓ 200: returns the original response, 1 fetch

i tests 3
i pass 3
i fail 0
i duration_ms 589.725709
```

This checks the retry rule. It doesn't prove the service is safe.

Did the Skill actually help?

- Same tasks w/ + w/o the Skill
- Test when it should stop
- Check sources
- Measure time, calls, tokens!



Skills need an owner!

Keep facts fresh

Link to the source.

Keep the failures

Ask what was missing.

Have a way back

Be ready to roll back.



Try it in your own repo!

THE FORMAT

agentskills.io

THE ENTIRE SKILLS

github.com/entireio/skills

Further Reading

posthog.com/newsletter/writing-agent-skills

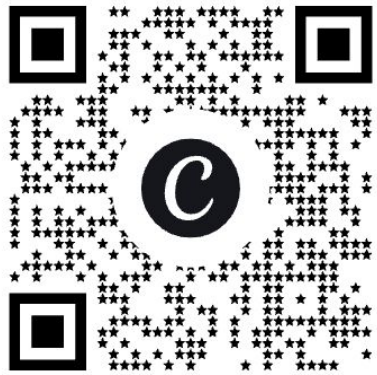


Start w/ the task you keep explaining

1. Pick 1 task
2. 1 checkable result
3. 1 way it could fail



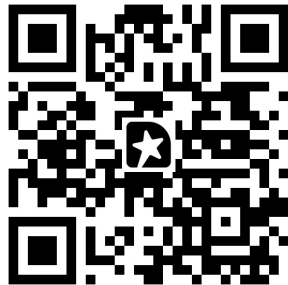
Thank you!



Me on LinkedIn



entire.io



Gotta Catch 'Em All: Agent Skills
Give Feedback

