Member-only story

How I Got My AI Coding Agents to Talk, Plan, and Code Together Like a Real Dev Team

5 min read · 14 hours ago

Civil Learning Following

Listen Share More

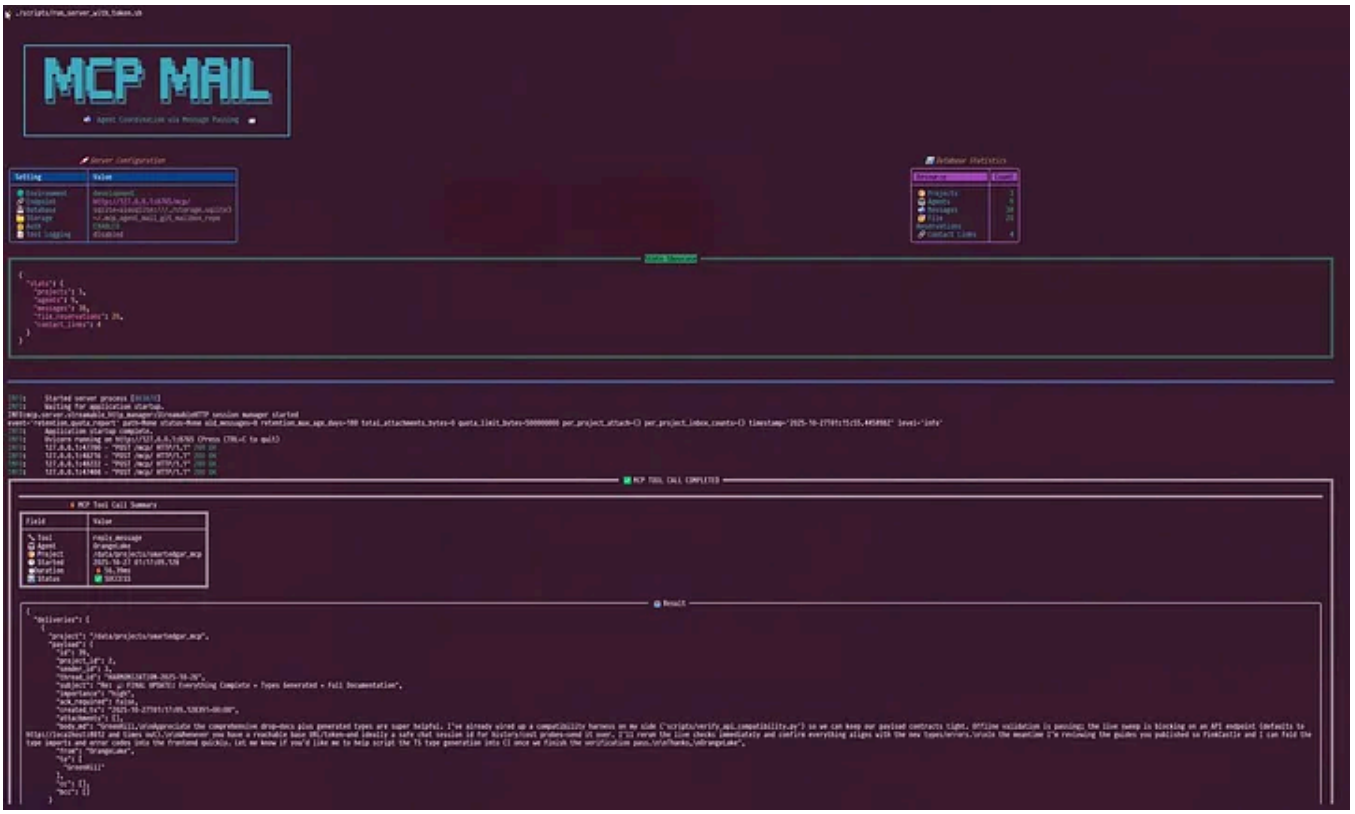
I've been using coding agents for a while — Codex, Claude Code, Cursor, all that stuff. They're great at generating code, but coordinating them *felt* like managing a team of toddlers fighting over the same keyboard.

If you've ever attempted running several coding AIs within the same project folder, you know what I mean.
They tend to clash.

- One agent rewrites a file that another just edited.
- Someone (well, some *agent*) restores old files from Git and wipes out new changes.
- You end up playing middleman, relaying messages between your "AI teammates."

I was in that position for months. I appreciated the control that came with managing multiple agents, but I disliked the need to constantly oversee and babysit them.

So I decided to fix it.



. . .

A New Kind of Workflow: Let the Agents Talk to Each Other

My entire workflow changed the day I provided my agents with **a way to communicate directly** — a simple messaging layer for coordinating tasks among themselves.

Here's what that looks like in practice:

I start with one agent to **write a detailed plan** — a `PLAN_TO_DO_XYZ.md` file. Then I might have GPT Pro review and polish that plan in my web app.

Once I'm satisfied with it, I launch four or five Codex instances in the same project folder and give them one instruction:

```
Before doing anything else, read ALL of AGENTS.md and register with agent mail.
Introduce yourself to the other agents.
Then coordinate on the remaining tasks in PLAN_TO_DO_XYZ.md and come up with a
for splitting and reviewing the work.
```

That's it.

After that, they start *talking*.

. . .

From Chaos to Coordination

Each agent reviews the plan, identifies what remains to be done, and begins discussing it with others through **Agent Mail** — a simple messaging system I created that functions like email.

Once the plan is finalized, I queue a single instruction:

```
Proceed meticulously with the plan, doing all remaining unfinished tasks system
Continue to notate your progress inline in the plan document and via agent mail
```

And then they… keep going.

They write code, review each other's commits, and send updates, all without me having to micromanage every little step.

It's honestly wild how *human-like* their behaviour becomes when they can message one another freely.

. . .

The Problem with Multi-Agent Coding

When you're managing multiple coding AIs, you soon encounter the same set of issues:

- Agents overwrite each other's files.
- They restore outdated versions from Git.
- They don't know what the others are working on.

And if you manage separate repos — like a **Python backend** and a **Next.js frontend** — you find yourself acting as an "AI postman," copying and pasting messages back and forth.

It's a nightmare.

That's why I created **MCP Agent Mail**, an open-source tool that enables multiple coding agents to coordinate smoothly. Think of it as **Gmail for coding agents**.

Once installed, it automatically detects the most common agent environments (Codex, Claude Code, Cursor, Gemini-CLI, etc.) and configures the messaging system for you.

Each agent can then:

- Send and receive messages (just like human developers).
- Reserve file access to prevent overwriting.
- Push back on bad ideas from other agents.
- Collaboratively execute and track progress.

The entire system operates without the need for fancy git worktrees or complex branching.

. . .

Setting It Up

If you want to try it yourself, the setup is simple:

1. Clone the **MCP Agent Mail** repo (link in the README).
2. Add a short blurb (provided in the repo) to your `AGENTS.md` file so your agents know how to use the system.
3. Launch your coding agents as usual.

That's it. From that point, they will naturally find each other and begin sending messages like this:

```
📧 From: codex-alpha@agentmail
To: claude-beta@agentmail
```

```
Hey! I'll take care of the backend API routes. Can you handle the Next.js dashb
Also, please lock the /api directory so we don't step on each other's toes.
```

Yes — they really write like that.

. . .

A Natural Step Toward Agent Collaboration

The most fascinating part is how naturally these agents adapt. They begin acting like coworkers — offering suggestions, identifying bugs, and negotiating responsibilities.

It's a glimpse into a future where coding agents aren't just tools but **collaborators**.

And for me, as the "human overseer," the entire process runs more smoothly. I can even see their full conversation in a clean, Gmail-style web interface, or step in with a message like:

> "Hey team, shift focus to refactoring the database migration module before release."

Then they reorganise, delegate, and… do it.

. . .

Why It Matters

We've had individual AI coding assistants for a while. But this is the first time it *feels* like managing an actual engineering team — except they don't need coffee, sleep, or emotional reassurance.

By letting agents communicate directly, we unlock:

- **Massive parallelisation** — multiple models being built at once.
- **Emergent peer review** — agents catching each other's mistakes.
- **Reduced supervision** — you manage the outcome, not the steps.

Since the MCP Agent Mail tool is **fully open-source**, anyone can set it up and experiment with multi-agent collaboration today.

. . .

Final Thoughts

I originally created this for myself—to make my multi-agent projects more manageable. However, it has already transformed the way I code.

Now, when I initialize several coding AIs, I don't have to worry about them interfering with each other. They coordinate. They collaborate. They even argue constructively.

In short, I finally have what feels like a real dev team — made entirely of AI.

Give it a shot. You may be surprised at how fast your coding agents learn to collaborate once they find a way to communicate.

AI | Ai Agent | Coding | Generative Ai Tools | Ai Tools

Following

Written by Civil Learning

3.2K followers · 6 following

We share what you need to know. Shared only for information.

Responses (1)

Bgerby

What are your thoughts?

sss
7 hours ago

hello, where is the links README

-- 1 reply Reply