
part two • the workshop

hack your own tools: hands on.

set up opencode, wire it to openrouter, ship your own website today

lenz gschwendtner • lnz.me

do this first

three signups, that's all you need



opencode.ai/download

the agent — nothing to sign up for, just
install



openrouter.ai

one key, every model — free tier is
enough to start



github.com/signup

your work, versioned and backed up —
and free hosting

scan all three now — you'll need each of them in the next hour

step one · install

get opencode running

```
# mac / linux  
curl -fsSL https://opencode.ai/install | bash
```

windows: scoop install opencode · or npm install -g opencode-ai · or brew install anomalyco/tap/opencode

then make a home for your work

```
mkdir my-site && cd my-site  
git init  
opencode
```

that's it. no account, no dashboard, no golden token.

an agent in your terminal, waiting for instructions.

step one • thirty seconds later

the five commands that matter

`/connect` — hook up a model provider

`/models` — pick which brain to use

`/init` — let the agent read your project, writes AGENTS.md

`tab` — switch between build mode and plan mode

`esc` — interrupt it when it goes off the rails

plan first, build second. read what it wrote before you commit.

step two • models

wire it to openrouter

1. openrouter.ai → settings → keys → create api key (starts with sk-or-v1-...)
2. in opencode: `/connect` → search "openrouter" → paste the key
3. `/models` → pick one — kimi k2, deepseek, gemini flash are cheap or free

one key, hundreds of models, one bill.

keys live in `~/.local/share/opencode/auth.json` — never in your repo.

bored of your model? `/models` and switch. that's the whole point.

step two • a word on ownership

openrouter got bought — have a backup

openrouter sold to stripe. maybe fine, maybe not your thing.

the point of this setup: you're not locked in.

requesty.ai does the same job, based in europe.

european privacy rules, hundreds of models, openai-compatible.

same prompt, different name: tell opencode to switch to requesty, paste your key, done.



requesty.ai

scan it, sign up

step two • even easier

let the agent configure itself

no config files. just say what you want, in plain words.

```
"set yourself up to use openrouter as your provider.  
my api key is sk-or-v1-... — put it in the right place  
and make kimi k2 your default model."
```

the agent writes its own config, restarts with the new model,
and confirms it worked. if something's wrong, say so — it fixes it.

that's the trick with agents: you describe, they wire it up.

step three • build something real

your personal website, written by an agent

```
"build me a personal website. one page, dark background,  
big type, no frameworks – just html and css.  
about me, what i do, links to my stuff.  
make it feel like a zine, not a linkedin profile."
```

then iterate like you're talking to a person:

"more contrast." "make the headline huge." "add a page for my field notes."

watch the files appear. open index.html in your browser.

plain html and css. no build step, no framework, nothing to break.

step four • ship it

live on github pages, free forever

```
"commit everything, create a public github repo  
called my-site, push it, and turn on github pages.  
then give me the link."
```

the agent handles git, the repo, and the pages setup.

a minute later it's live at `you.github.io/my-site`

every change from now on: "commit and push" — that's the whole pipeline.

your site, your repo, your domain. no platform in between.

stretch goal

netlify — same repo, fancier plumbing

- sign up with your github account — no new password to forget
- "add new site" → import your repo → deploy
- every push rebuilds; pull requests get preview urls
- free tier covers a personal site with room to spare

github pages is plenty. netlify is for when you outgrow it —
forms, redirects, serverless functions, analytics without trackers.

and because the repo is yours, switching hosts is one push away.

try these

prompts to steal

01 "read this folder and tell me what's here, then propose a structure for my notes."

02 "make my site load faster and work on a phone. show me what you changed and why."

03 "write a field note about what i built today. blunt, lowercase, no buzzwords. commit it."

04 "explain this error like i'm smart but new, then fix it."

the pattern: describe the outcome, let the agent figure out the how

you just shipped your own corner of **the web.**

an agent, a model key, a repo, a live site.

no golden tokens were harmed.

lenz gschwendtner

ecosystem builder · serial entrepreneur · punk

lnz.me



lnz.me