

DevCLI Troubleshooting Notes – 2025-10-19

This document captures today's work on DevNotes AI CLI + Docker/Ollama integration:

- achievements
- issues
- fixes
- plan for tomorrow

Session Summary

- Built/installed Dev CLI (container & host variants).
- Verified Ollama API reachable (version 0.12.6).
- Confirmed model replies inside container (warm-up via: `echo "hello" | ollama run mistral:latest`).
- Host tools installed: `devask_h`, `devteach_h`, `devnote_h`.
- Git push verified: `docs/ai_sessions/session_*.md` created and pushed.

Unresolved/Observed Issues

- 1) Container devask/devteach: sometimes no visible output.
 - Non-stream body occasionally empty; streaming printer previously too strict.
- 2) Host tools initial run: no output.
 - Parser expected only 'message.content'; some builds stream 'response'.
- 3) Docker healthcheck flakiness.
 - Resolved by removing healthcheck and probing manually.

Effective Fixes Applied Today

A) Compose simplification: removed healthcheck dependency.

B) Container CLI: prefer streaming; warm model explicitly before use.

C) Host CLI: robust streaming parser accepts both keys:

```
msg = (d.get("message", {}).get("content") or d.get("response") or "")
```

D) Repo flow: devnote_h saves markdown and auto-commits/pushes.

Known-Good Commands

Prove API

```
curl -s http://localhost:11434/api/version && echo
```

Warm model in container

```
docker compose exec -T ollama sh -lc 'echo "hello" | ollama run mistral:latest'
```

Host tools (streaming; guaranteed visible output)

```
export OLLAMA_MODEL=mistral:latest
```

```
devask_h "Say one short sentence."
```

```
devteach_h "Explain POSIX log rotation in 3 steps."
```

```
DEVREPO="$HOME/Projects/devnotes" devnote_h "Summarize today's setup in 5 bullets."
```

Raw streaming probe

```
printf '%s' '{"model":"mistral:latest","messages":[{"role":"user","content":"say hi"}], "stream":true}' | curl -sN http://localhost:11434/api/chat -H 'Content-Type: application/json'
```

Plan for Tomorrow

- 1) Port host-side dual-key streaming parser to container devask/devteach.
- 2) Optional: add `_wait_ollama` readiness helper (up to 60s retry).
- 3) Unify tools: single command with persona flag, e.g. `devask_h --teacher "..."`.
- 4) Add smoke-test script to validate API, stream, and repo write.
- 5) Keep default model lightweight (mistral:latest or qwen2.5:3b-instruct).

Appendix: Paths

- Repo: `~/Projects/devnotes`
- AI sessions: `docs/ai_sessions/session_*.md`
- Host tools: `/usr/local/bin/devask_h`, `devteach_h`, `devnote_h`
- Persona: `~/.devnotesctl/Operator_Teacher.txt`
- Docker project: `~/devnotes-docker`