

How 72s does Ai



The Concept

Keep the Ai engine interchangeable

Who knows whats good in a year

Setup an Obsidian data-store

humanly readable .md files

Let Ai document everything there

We can reuse that later in other places



The Situation

- Working in a team
- Wanting to improve together

Aim

- Make install reproducible
- Maintain separate users
- Use the same knowledge base about company, products and methods



The Idea

Sync the Obsidian DataStore via Nextcloud

The Problem

Claude and Claude-code set up in /home

—> no Sync



The Solution

Put all the setup in the NextCloud sync folder

Let claude write a setup.sh including

- versioning
- version adoption sign-off



The setup.sh

1. **Ensure all symlinks** — runs first so `.claude/` exists before any config is written into it:

- `.claude` → `claude-config` (relative) inside the workspace root and `filemaker/fm-lab/`
- `~/ai-workspace/.claude` → `_AI_Workspace/claude-config`
- `~/fm-lab/.claude` → `filemaker/fm-lab/claude-config`
- `fm-lab/node_modules`, `rest-api/node_modules`, `packages/shared/node_modules`, `apps/web/node_modules` → corresponding `~/fm-lab/*` directories (only wired if the target already exists, i.e. `npm install` has been run)

2. **Render settings.local.json** from the template, substituting `{{WORKSPACE}}`, `{{HOME}}`, and `{{HOME_PATH}}` for the current machine's path



3. **Hardlink** `~/.claude/settings.json` to `_setup/claude-user-settings.json` (symlink fallback for cross-filesystem)
4. **Make hook scripts executable** (`chmod +x .claude/hooks/*.sh`)
5. **Install daily cron job** at 08:00 to re-run `setup.sh --quiet` and catch drift
6. **Write version sign-off** — `_setup/signoffs/$USER.md` records the applied version
7. **Update setup stamp** — writes the current epoch timestamp to `~/.claude/.workspace-setup-stamp`, marking when setup last ran successfully



What Ai does for 72s



Fix-Starter

- Marketing
- Support - Documentation - HelpTexts
- Coding
 1. Laravel
 2. Javascript
 3. n8n
 4. Ansible

AND





Magic by Marcel Moré (Beta-Test Programm)

FM-LAB

“A **DuckDB**-based tool for analyzing **FileMaker SaXML exports**. Converts the XML structure of a FileMaker solution into a queryable DuckDB database — covering all object types and their dependencies — for fast cross-reference analysis.”



[How-It-Works](#)



Now you can either:

<http://localhost:5173/>

—> Demo



Or:

You can tell Claude code, that there is a DuckDB datastore for FileMaker and you can now

- research and document processes
- plan changes
- plan test scenarios
- write out checklists
- quality control

—> Demo



Staus Quo

Research, suggestions and documentation ...

Local DuckDB for every user

Outlook

‘FMUpgradeTool —update’ (once this is reliable)

Who want’s to copy and paste?

Centralized DuckDB for the whole office



What is your take on
all of this?

