

Building a Current Context Skill for Claude

How to give Claude persistent awareness of your work across every conversation

The Problem

Claude is powerful, but stateless. Every new conversation starts from zero — no memory of your active projects, upcoming deadlines, or open loops. If you do serious work inside Claude Projects (focused workspaces for specific clients or initiatives), the problem compounds: context built in one Project doesn't carry into another.

The result is a lot of re-explaining. Claude is capable of being a genuine thought partner for prioritization and planning, but only if it knows where you stand.

The Solution: A Personal Context Skill

A Skill is an installable instruction set for Claude — a Markdown file that teaches Claude how to do something specific and consistently. Skills are account-level, meaning they're available in every conversation, including inside Projects.

We built a skill called

This skill:

- Fetches a structured "Current Context" page from Notion via MCP — no file uploads required
- Reads your active projects, upcoming commitments, open loops, and personal bandwidth
- Uses that context to inform planning, prioritization, and pinch-point conversations
- Offers to walk you through updates and write them back to Notion when the page goes stale

What's in the Context Page

The Notion page has four fixed sections Claude reads predictably:

Section	What it contains
Active Projects	Name, status, next action, deadline
Upcoming Commitments	Dates and what's required of you
Open Loops / Waiting On	Stalled items, who they're waiting on, since when

Section	What it contains
Personal Context	Energy, bandwidth, health, current focus priorities

The structure is intentionally rigid — Claude reads it the same way every time, which makes it fast and reliable.

How It Works in Practice

- You maintain one Notion page: 🕒 Dan's Current Context
- The skill is installed in your Claude account once
- Whenever you're in any Claude conversation — including inside a Project — and you say something like "help me think through my week" or "where do things stand across my work", Claude fetches the page automatically
- Claude surfaces conflicts, overloads, stale open loops, and priority recommendations based on what it finds
- If the page is more than 7 days old, Claude flags it and offers to walk you through a refresh, writing the updates back to Notion directly

No uploads. No copy-paste. No re-explaining your situation from scratch.

Why Notion (Not a File)

The first version of this skill used a Markdown file you'd upload to each conversation. We scrapped it immediately — the upload friction would have killed the habit.

Notion works because:

- Claude has a Notion MCP connection at the account level, available everywhere
- You're already living in Notion, so updating one page is low-friction
- Claude can write back to the page, not just read from it
- The page URL is hardcoded in the skill, so Claude always knows exactly where to look

How to Build This Yourself

Step 1: Create your context page in Notion

Create a page with these sections as headers, each containing a table or freeform notes:

- Last Updated
- Active Projects
- Upcoming Commitments
- Open Loops / Waiting On
- Personal Context

Note the page ID from the URL (the UUID at the end).

Step 2: Write the SKILL.md

A skill is a Markdown file with a YAML frontmatter block at the top. The description field is the triggering mechanism — Claude reads it and decides whether to use the skill. Make it specific about when to trigger, not just what it does.

Step 3: Package and install

The .skill file is a zip of your skill folder. Create it and install it via Claude's skill settings.

Step 4: Connect Notion

In Claude settings, connect your Notion account as an MCP integration. Claude will be able to fetch and update pages on your behalf.

Key Design Principles

Principle	Why it matters
Frictionless > comprehensive	A context system you actually maintain beats a perfect one you abandon.
Structured > freeform	Fixed sections mean Claude reads your context the same way every time.
Pull, don't push	Claude fetches context when relevant — the skill description controls when it triggers.
Two-way	Claude doesn't just read your context — it can update it. That keeps the page from going stale.
Account-level	Skills and MCP connections are global. Build it once, use it everywhere including inside Projects.

The Broader Pattern

This approach — a skill that fetches a structured Notion page — is reusable for many problems:

- A client context skill that pulls a CRM-style Notion page before drafting emails or proposals
- A weekly review skill that fetches your review template and walks you through it
- A content calendar skill that checks what's scheduled before Claude helps you draft something new

The pattern: structured Notion page + Notion MCP + skill with a hardcoded page ID = persistent, zero-friction context for Claude.

Built during a live skill-writing session on Claude.ai · April 2026