Sample: dispatcher questions · michalgorski.com: articles, admin panel, assistant

2026.10.10

This is a sample artifact from an AI-driven delivery process, attached to the article "AI SDLC: cheaper code & new risks to manage". It was generated by the dispatcher, an AI agent that plans the work and runs coding sub-agents. The original was in Polish; this is an English translation. The owner's answers are shown in the grey boxes, translated. Internal paths and server details are removed.

How it works: the dispatcher does not wait for answers. It starts implementing its recommendations right away and corrects course when the owner sends the filled-in document back.

Questions for features A (articles section), C (admin panel) and B (the assistant learns from articles). Each has a recommendation; work starts from the recommendation, and your answer can change it.

Type your answer in the grey box: a letter, your own text, or paste a picture. Empty box on a question = I take the recommendation. Empty box on a decision D = it stays.

A. Articles section on the site

Q1. What is the section called, what is its address, and which sentence sits under the index heading?

DEC-301 left the name, address and languages to your decision. The name goes into navigation only when the section is switched on publicly. The sentence under the heading is new copy on the site.

- **A. Articles / Artykuły (recommended)**: /articles and /pl/artykuly. Under the heading: "How software delivery changes when AI writes part of the code, written for the people accountable for the result."
- **B. Writing / Teksty**: /writing and /pl/teksty, same sentence as in A.
- **C. Your own name or sentence**: Type the EN/PL name and the sentence under the heading.

Q1 · Your answer (a letter, your own text, or paste a picture)
We are writing the article now, and we will upload it through the article upload interface in the admin panel.

Q2. What about numbers in an article that have no claim ID in claims.json?

Site rule 1.4: every number has a range, a unit and a source. A claim ID in claims.json needs a pull request and a deploy, but articles are meant to go in through the panel with no deploy.

- **A. Warning plus your confirmation in the panel (recommended)**: Validation lists the numbers with their sentences; Publish works only after you tick "numbers checked". Claim IDs for this session's numbers come in a separate pull request; publishing does not wait for them.
- **B. Block without a claim ID**: Every number needs an entry in claims.json before publishing. Every article with numbers needs a pull request and a deploy.
- **C. No number checks**: Fastest, departs from rule 1.4.

Q2 · Your answer (a letter, your own text, or paste a picture)

B. The assistant learns from articles

Q3. Do facts from an article go through your review in the panel before the assistant uses them?

Today the assistant knows only its knowledge base (about 70 facts) and your corrections. After an article is published, the server can extract facts from it with one model call (about 0.02 USD per article).

- **A. Review in the panel (recommended)**: The panel shows candidate facts; you approve, edit or reject each one. Only approved facts reach the assistant, with a summary of the article.
- **B. Automatic**: All extracted facts go to the assistant at once. Faster; risk of a fact you do not want in answers.
- **C. The whole article in the prompt**: No extraction and no review. Simplest, most expensive (see Q5); the model may combine sentences into new claims.

Q3 · Your answer (a letter, your own text, or paste a picture)

B. Send me the extracted facts by email. If I do not like one of them, I want to be able to delete or edit it. I also want my assistant to reason over the whole article, not only the facts. Note a feature for later: when there are many articles we will need RAG. Not now, but let's keep it in the plan.

Q4. How does the assistant treat numbers from an article with no claim ID?

Today nothing checks numbers in the assistant's answers except the prompt rules. Numbers in the knowledge base come from the site specification.

- **A. Only inside approved facts (recommended)**: A number may appear in an answer if it stands in a fact you approved, with the article as its source.
- **B. No numbers outside claims.json**: Facts with numbers that have no claim ID are dropped at extraction.
- **C. No limits**: Any number from the article may appear in an answer.

Q4 · Your answer (a letter, your own text, or paste a picture)

C

Q5. How much article knowledge do we add to the assistant under the 1 USD daily cap?

Today the prompt is about 11 thousand tokens: about 0.008 USD per answer with cache and 0.022 USD without, so 45 to 125 answers a day within the cap. Facts from one article are about 800 tokens (+5% cost per answer); the full text is about 3,500 tokens (+25%).

- **A. Facts, up to 6 thousand tokens (recommended)**: Only approved facts and summaries, up to about 6 thousand tokens in total (about 6 articles; above that the panel asks you to choose). Daily cap of 1 USD unchanged.
- **B. Full text, cap unchanged**: Full articles in the prompt; fewer answers a day within 1 USD.
- **C. Raise the daily cap**: For example to 2 USD; type the amount. This is spending above the limit, so it is only your decision.

Q5 · Your answer (a letter, your own text, or paste a picture)

C, raise it to 2 USD

Decisions already taken (being implemented now)

D1. Article package format

Status: in progress (package P0)

I chose: Markdown with YAML front matter, one file per language, images alongside or in a .zip: Fields: slug, lang, title, summary, date, updated, translationOf (the PL file points to EN), cover, coverAlt, assistant. Locally: a template, an article-package skill and a check command with the same validation as the panel.

Alternatives:

- **B. One file with both languages**: Fewer files, harder to fix the PL version on its own.
- **C. A form in the panel instead of a file**: No local Claude Code needed, worse for long texts.

D1 · Keep it? Empty = keep. Or an alternative letter or your own text

D2. Section visibility

Status: in progress (package P1)

I chose: Without ?articles=on the pages return 404: With the flag they work normally, with noindex; not in navigation, sitemap or llms.txt. Switching on publicly (navigation, sitemap, indexing) is one configuration change, on your word.

Alternatives:

- **B. Pages visible without the flag, noindex only**: Simpler, but the address can be found and shared.

D2 · Keep it? Empty = keep. Or an alternative letter or your own text

Put this somewhere so I remember to switch it on.

D3. Admin panel

Status: in progress (package P2)

I chose: A new Articles tab in the admin panel: Drag in files or a .zip, instant validation, draft preview, Publish goes live, Unpublish returns to draft. The last 5 versions are kept. Upload in chunks, because the server accepts at most 64 KB per request; that security limit stays.

Alternatives:

- **B. Raise the request limit to 10 MB**: Simpler upload, weaker server protection.

D3 · Keep it? Empty = keep. Or an alternative letter or your own text

B, but with a 5 MB limit

D4. Article data on the server

Status: in progress (package P0)

I chose: Files in the assistant's existing data directory: It is the only directory the service may write to, so no server configuration change and no manual step. Survives every release. The source copy stays with you locally.

Alternatives:

- **B. A separate data directory**: Needs a change to the service configuration on the server (your decision and your step).

D4 · Keep it? Empty = keep. Or an alternative letter or your own text

D5. The assistant and the hidden section

Status: package P3, after answers to Q3 to Q5

I chose: Facts after your approval; a link to the article only after the public switch-on: The assistant can answer from the article right after review, but does not show a link to the hidden section.

Alternatives:

- **B. The assistant learns only after the public switch-on**: No demo before launch.
- **C. A link with the ?articles=on flag**: Reveals the hidden section to anyone who asks.

D5 · Keep it? Empty = keep. Or an alternative letter or your own text

D6. The article text

Status: rule

I chose: I do not publish it: Code goes to production after green CI. You upload and publish the article yourself, after your yes. I can upload it as a draft for preview once the article is ready.

D6 · Keep it? Empty = keep. Or an alternative letter or your own text

D7. Selected writing on the About page

Status: unchanged

I chose: Stays off until the section is switched on publicly: When switched on, the block can show the newest articles.

Alternatives:

- **B. Switch on now with flagged links**: Reveals the section.

D7 · Keep it? Empty = keep. Or an alternative letter or your own text

Access tasks (passwords, accounts, server): none. Everything fits within the dispatcher's permissions.