

Newsletter Automation & Workflow

How newsletters move from the Chronicle into Listmonk, and the plan to automate drafting in n8n.

- [Newsletter Workflow: BookStack to Listmonk \(Draft Automation\)](#)

Newsletter Workflow: BookStack to Listmonk (Draft Automation)

Newsletter Workflow: BookStack ? Listmonk (draft automation)

What we're automating

Publishing a newsletter is a two-app chore: the content lives in the Chronicle book of BookStack (book id 24), and the send happens in Listmonk. Right now each issue is moved by hand. This page records the process we followed for the September 2026 issue and the plan to automate the "draft it into Listmonk" half of it in n8n.

Why drafts — not scheduled sends

When a campaign is created in Listmonk with no `send_at` date, it is stored as a **draft** and stays there until a human schedules or sends it. That is exactly the right automation boundary:

- The hard part to automate is the boilerplate: pulling the page, cleaning the content, converting it to email HTML, and creating the campaign with the right list, template, subject, and tags.
- The human step that shouldn't be automated blindly is deciding when it goes out (and reviewing the draft first). So the automation stops at "campaign created as a draft," and a person schedules/sends it in the Listmonk UI.
- This also sidesteps a known bug: the deployed Listmonk MCP server's `schedule_campaign` tool posts `send_at` to the `/status` endpoint, which ignores it, so every schedule fails with "Campaign needs a date to be scheduled." Drafts need no date, so the automation never touches that.

The manual process (what the September 2026 issue looked like)

1. Locate the issue in the Chronicle: book id 24, chapter "2026", page "September 2026 Newsletter" (page id 220 on 2026-09-07).
2. Read the page's raw content. BookStack's WYSIWYG editor stores it as markdown with `<span data-preserver-spaces="true">` wrappers around nearly every block and inline run.
3. Strip the span tags to recover clean markdown, then convert to the richtext HTML Listmonk wants. Verification: check for balanced `**/`, no leftover `<tags>`, links intact, heading levels preserved.
4. Create the campaign via the Listmonk MCP server with:
 - name = page name ("September 2026 Newsletter")
 - subject = a punchy topic phrase, matching the house style (e.g. "What's Really in Your Teacup? Part II: Beyond Organic")
 - lists = [3] (Newsletter Welcome)
 - template_id = 1 (Default campaign template)
 - content_type = "richtext", body = the converted HTML
 - tags = a consistent scheme (safe set: `newsletter`, a topic tag, a date tag such as `september-2026`)
5. Confirm status is `draft`, with the full body intact via `get_campaign`.

Note on finished campaigns: Listmonk refuses to update campaigns whose status is `finished` or `running` ("Cannot update a running or a finished campaign"). So tags and content must be set while the campaign is a draft. Once something has gone out you can no longer edit it — tag at creation.

MCP routes and clients used

All use the same `Authorization: Basic <base64>` header from `~/.hermes/config.yaml` (the `mcphub` entry). Two front-ends exist; verify with `tools/list` first because they shift:

- **BookStack** — `mcpserver.tea-tavern.com/servers/bookstack/sse` (38 tools, `get_/create_/update_*` naming)
- **Listmonk** — `mcpserver.tea-tavern.com/servers/listmonk/sse` (26 tools; `create_campaign`, `get_campaign`, `update_campaign`, etc.)
- Client: `~/.hermes/scripts/mcp_call_url.py "<url>" "$AUTH" <tool> '<json>'`

The plan: automate the drafting in n8n

Goal: when a new page is created in the Chronicle, n8n drafts it into Listmonk. The flow (built from the n8n MCP server's capabilities):

1. **Trigger** — BookStack webhook on `page_create` → n8n **Webhook** node. (Fallback if webhooks aren't wired: Schedule + HTTP polling of `/api/pages`, de-duped.)
2. **Filter** — If node: only proceed when `book_id == 24` (Chronicle).

3. **Fetch** — HTTP GET `https://bookstack.tea-tavern.com/api/pages/{id}` with `Authorization: Token <bookstack-api-token>`.
4. **Convert** — Code node: strip the `data-preserver-spaces` spans, then markdown→HTML to the richtext body Listmonk expects.
5. **Draft** — HTTP POST `https://listmonk.tea-tavern.com/api/campaigns` with `Authorization: Token <listmonk-api-token>` and the campaign body above, no `send_at`, so it's created as a draft.

Credentials needed in n8n's store: a BookStack API token and a Listmonk API token. Open decisions: how the subject line is derived (the page name is "September 2026 Newsletter", not the punchy subject), and whether tags should be read from the page's BookStack tags instead of hard-coded.

Tag scheme (house convention)

- Monthly newsletters: `newsletter`, `<topic>`, `<yyyy-mm>`
- Topical/education: topic as one hyphenated tag (caffeine, water, tea-history, is-organic-tea)
- Event eBlasts: `newsletter`, `event`, `<event>-<yyyy>`
- Welcome / opt-in flows: `welcome`, `onboarding`