

Docs » Views and Templates Category » Content Project » MCP Plugin

MCP Plugin

Exposes the content as an MCP (Model Context Protocol) server for AI agents.

Anonymous · 21/08/2026

MCP plugin

Exposes the content of the website as an [MCP](#) (Model Context Protocol) server, so AI agents (Claude Code, Claude Desktop, Cursor, claude.ai connectors, etc.) can search and fetch it directly as tools during their own conversation — instead of relying on stale training data or scraping HTML.

This is a different consumption channel than the [API](#) plugin: the API plugin is a one-shot bulk export meant for an indexing pipeline (Qdrant, etc.); the MCP plugin is a live, callable interface for any MCP-capable agent.

Endpoint

A single route, `POST /api/mcp` (also accepts `DELETE` to end a session, and `OPTIONS` for CORS preflight). It deliberately has **no .json suffix**: unlike every other endpoint in this package, its response is not always JSON — it can also be `text/event-stream` (SSE) when the protocol needs to keep the connection open mid-call. The MCP protocol negotiates that per-request via the `Accept` header, not via the URL, and every real MCP client already expects a plain endpoint URL with no extension.

It speaks JSON-RPC 2.0 over the “Streamable HTTP” transport of the [official PHP MCP SDK](#) (`mcp/sdk`), synchronously — there is no event loop involved, it fits the same request/response model as every other controller in this package.

Configuration (`services.yaml`)

Enabled under `derafu.content.config.plugins.mcp`:

```
parameters:
  derafu.content.config:
    plugins:
      mcp:
        ask:
          enabled: true
```

Option	Type	Default	Description
<code>server_name</code>	string	<code>derafu-content</code>	Name announced to MCP clients during the <code>initialize</code> handshake.
<code>server_version</code>	string	<code>1.0.0</code>	Version announced to MCP clients during the <code>initialize</code> handshake.
<code>session_path</code>	string	a subdirectory of <code>sys_get_temp_dir()</code>	Directory where MCP sessions are persisted on disk between requests (a session is created on <code>initialize</code> and referenced by later calls via the <code>Mcp-Session-Id</code> header; since each request builds a new server instance, this cannot live in memory). Point it at something under the website's <code>var/</code> directory for a more durable/cleanable location.
<code>session_ttl</code>	int	<code>3600</code>	Time to live, in seconds, of a persisted session.
<code>ask.enabled</code>	bool	<code>false</code>	Whether the <code>ask</code> tool (LLM-backed conversational answers) is registered. See Tools below.

This plugin does not handle authentication nor rate limiting — that is not its responsibility. If the endpoint needs to be protected, do it at the HTTP stack level (e.g. with the middlewares of `derafu/http`), before this controller is reached.

The `mcp/sdk` dependency

`mcp/sdk` is a `require-dev` dependency (with a `suggest` entry) of `derafu/content`, not a hard `require`. If you enable this plugin, add it explicitly to **your own** `composer.json`:

```
{
  "require": {
    "mcp/sdk": "^0.7"
  }
}
```

If you don't enable the plugin, nothing needs it: hitting `/api/mcp` without it installed and without the plugin configured fails with a normal “plugin not found” error, exactly like hitting `/api/search.json` without the search plugin configured — it does not affect any other endpoint of the website.

Tools

Tool	Always registered?	Description
<code>search_content</code>	yes	Semantic search across the indexed content, delegating to the Search plugin's engine (Qdrant, or whatever is configured there). Accepts an optional source (<code>academy</code> , <code>blog</code> , <code>docs</code> , <code>faq</code> , <code>pages</code> , or <code>all</code>) to scope the search; omitting it or passing <code>all</code> searches every source at once.
<code>get_content</code>	yes	Fetch a single item by source and URI: full Markdown body plus metadata (title, tags, dates, authors).
<code>list_content</code>	yes	Browse/filter the items of a source (<code>academy</code> , <code>blog</code> , <code>docs</code> , <code>faq</code> , <code>pages</code>), optionally by tag, category or free-text search.
<code>list_tags</code>	yes	List the tags used in a source, with how many items use each one.
<code>ask</code>	only if <code>ask.enabled: true</code>	Ask a natural-language question and get a conversational answer from the LLM configured in the Search plugin, grounded on the indexed content.

`ask` is opt-in because, unlike the other four tools (which only ever read from the indexed content), it depends entirely on the quality of whatever LLM backend answers it — a bad or failed answer there makes the whole MCP server look unreliable, even though the rest of the tools never touch an LLM at all.

Content frontmatter

Not applicable — this plugin has no content items of its own, it reuses whichever content plugins (Academy, Blog, Docs, FAQ, Pages) are already enabled.

Last updated on 21/08/2026