Search Plugin

Search engine integration, with an optional LLM-based conversational answer.

Anonymous · 21/08/2026

Search plugin

Provides a search page and API that proxy to an **external** search engine — this plugin does not index anything itself, it queries whatever semantic/full-text search service you point it at (in practice, a service backed by Qdrant, fed by the API plugin's export). It optionally also proxies to an LLM to answer questions conversationally, grounded on that same indexed content.

Routes

Route	Path	Description
`search`	`GET /search`	Search page (HTML).
`search_api`	`GET /api/search.json?q=...`	Search results as JSON.
`search_llm_query`	`GET /api/search/llm.json?q=...`	Conversational answer from the LLM (see LLM backend below).

Configuration (`services.yaml`)

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

```
parameters:
    derafu.content.config:
        plugins:
            search:
              url:
'https://search.example.com/api/search?collection=%s&base_url=%s&text=%s'
              collection: 'my-site'
              base_url: 'https://example.com'
              llm_url: 'https://api.openai.com'
              llm_model: 'gpt-4o-mini'
              llm_api_key: '%env(OPENAI_API_KEY)%'
```

Option	Type	Default	Description
`url`	string	*(required)*	`sprintf()` template for the search engine's URL. See URL template below.

Option	Type	Default	Description
`collection`	string	none	Collection/index identifier, URL-encoded and injected into `url`.
`base_url`	string	none	Base URL of the website, URL-encoded and injected into `url` (useful when the search backend serves more than one site).
`llm_url`	string	none	Base URL of the LLM backend. Leave unset to disable the `ask` tool (MCP) and make `search_llm_query` fail.
`llm_model`	string	none	Model name sent to the LLM backend. **Required if `llm_url` is set** — there is no generic default, since no single model name makes sense across every provider. `llm()` throws immediately with a clear message if `llm_url` is configured without it, instead of silently sending an empty model name to your backend.
`llm_api_key`	string	none	API key sent as `Authorization: Bearer <key>`.
`llm_completions_path`	string	`/v1/chat/completions`	Path of the chat completions endpoint, appended to `llm_url`. See LLM backend.

URL template

`url` is a `sprintf()` template consumed in this order — trailing placeholders can be omitted:

1. `%s` → collection (only if both `collection` and `base_url` are set).
2. `%s` → base URL (only if both `collection` and `base_url` are set).
3. `%s` → the URL-encoded search query (always present, always last).

If `collection` or `base_url` is not configured, `url` is expected to have a single `%s` for the query only.

LLM backend

The `llm_*` options configure a client for **any backend exposing an OpenAI-compatible chat completions endpoint** — the `{model, messages}` request / `choices[0].message.content` response shape that has become a de facto standard. With no code changes, that covers:

- OpenAI itself (`llm_url: https://api.openai.com`, default `llm_completions_path`).
- OpenRouter (`llm_url: https://openrouter.ai/api`, default `llm_completions_path`, `llm_model` like `anthropic/claude-sonnet-4.5`) — itself proxies Claude, Gemini, Llama, etc. through the same shape.

- Anthropic's own OpenAI-compatible endpoint (`llm_url: https://api.anthropic.com`, default `llm_completions_path`).
- Self-hosted Open WebUI — needs `llm_completions_path: /api/chat/completions` instead of the default, since it does not follow the standard path.
- Groq, Together AI, Ollama, Azure OpenAI (`Authorization: Bearer` variant), and most other providers.

A backend that does not speak this protocol at all (a provider's native, non-compatible API) is not supported today, but the client is behind an interface (`Derafu\Content\Plugin\Search\Contract\LlmClientInterface`) precisely so an alternative implementation can be swapped in later without touching `SearchController` or the `ask` MCP tool.

Failures from either the search engine or the LLM backend (unreachable host, non-200 response, unexpected response shape) surface as `SearchUpstreamException`, mapped to **502 Bad Gateway** — not a generic 500 — since the request to this website was fine, it's whatever it depends on that failed. The error message includes whatever detail the upstream returned (`error`/`error.message`/`detail`, depending on the provider).

Content frontmatter

Not applicable — this plugin has no content items of its own; `searchable` (see generic frontmatter) is read by whoever builds the external search index, not by this plugin.

Last updated on 21/08/2026