

Inspect Embeddings

Admin Check what is registered for embedding, whether it is covered, and whether the queue is healthy.

Screen Embeddings embeddings-setup.html

Reads what gets embedded embedding_ingest_registry the vectors embedding the queue embedding_job

Key fields which table source_schema · source_table what text to embed text_template

how to split it chunking_policy what to carry along filter_columns switched on? is_active

job health status · attempts · error

1 · Given / When / Then

Human wording first; the exact table/field in `mono`.

GIVEN	The signed-in person is an admin . Embedding is registry-driven: a row in <code>embedding_ingest_registry</code> declares that a given table should be embedded, and how.
WHEN	On the Embeddings Screen they review the registry: which tables are registered, the <code>text_template</code> that composes the text, the <code>chunking_policy</code> , the <code>filter_columns</code> carried onto each vector, and whether the entry <code>is_active</code> . They then check coverage and the health of the <code>embedding_job</code> queue.
THEN	Reading changes nothing. Re-queueing a stuck job sets its <code>status</code> back to <code>pending</code> ; the worker picks it up, replaces that source row's <code>embedding</code> rows, and finishes at <code>done</code> . reads <code>embedding_ingest_registry</code> , <code>embedding</code> , <code>embedding_job</code> writes <code>embedding_job.status</code> when re-queued

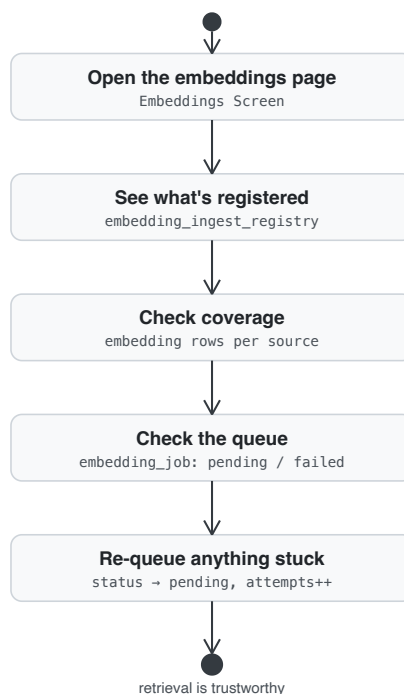
Variations and exceptions

IF...	THEN...
A registry entry is set to <code>is_active = false</code>	New work stops; the rule and the vectors already built stay.
Source rows have no <code>embedding</code> rows	Retrieval silently misses them — the coverage view exists to make that visible.
A job keeps failing	It shows a rising <code>attempts</code> and a stored <code>error</code> instead of retrying forever in silence; re-queueing sets it back to <code>pending</code> .
The <code>text_template</code> or <code>chunking_policy</code> is changed	Existing vectors were built under the earlier rule and match the new one only after a re-embed.

Acceptance test — *Given* a failed embedding job, *When* the admin re-queues it, *Then* its `status` returns to `pending` and on success reaches `done` with that source row's `embedding` rows replaced, not duplicated.

2 · The Journey, Screen by Screen

Registry → coverage → queue health.



3 · In Plain English

Retrieval only works if the material was actually turned into vectors. This screen exists so that this is checkable rather than assumed.

What gets embedded is not hard-coded. A registry says: this table, composed into text this way, split this way, carrying these extra fields for filtering. Turn an entry off and new work stops without destroying what is there.

Two failure modes are worth watching: gaps, meaning source rows nothing was ever built for, and stuck jobs. Both are shown rather than hidden, and a stuck job records how often it tried and what the error was, so you are diagnosing rather than guessing.

If in doubt you can always rebuild. The vectors are derived from the source text, so re-embedding costs time and nothing else.

Fits the schema cleanly. The `embedding_job` state machine is drawn in *Ingest a document*; this scenario is the operational view over it. `embedding` is derived data: losing it costs compute, never information.