Curate the Model Catalog

Admin The one list of AI models the install may use: add, describe, and retire.

Screen Model catalog `model-catalog.html`

Writes the catalog entry `model_catalog`

Key fields who supplies it `provider` · its API id `model` · readable name `display_name` · selectable? `enabled` · strength tags `aptitudes` · routing sentence `expertise` · published numbers `benchmarks` · how much it can read `context_window` · free-text `notes`

1 · Given / When / Then

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

GIVEN	The signed-in person is an **admin**. API keys for the providers live in the server's configuration, never in the database.
WHEN	On the **Model-catalog Screen** they add a model: its `provider` and exact API `model` id, a human `display_name`, what it is good at — short tags in `aptitudes` for filtering the picker, and one plain sentence in `expertise` describing its strengths in prose — any published `benchmarks` worth recording, its per-token prices, its `context_window` — how many tokens it can accept in one call — and free-text `notes`. Setting `enabled = true` makes it selectable.
THEN	A `model_catalog` row exists and now appears when an experimenter builds a model set-up. The catalog is the single list of models the whole install may use. Nobody types a raw model id anywhere else; a `model_config_model` always points at a catalog entry. **writes** `model_catalog`

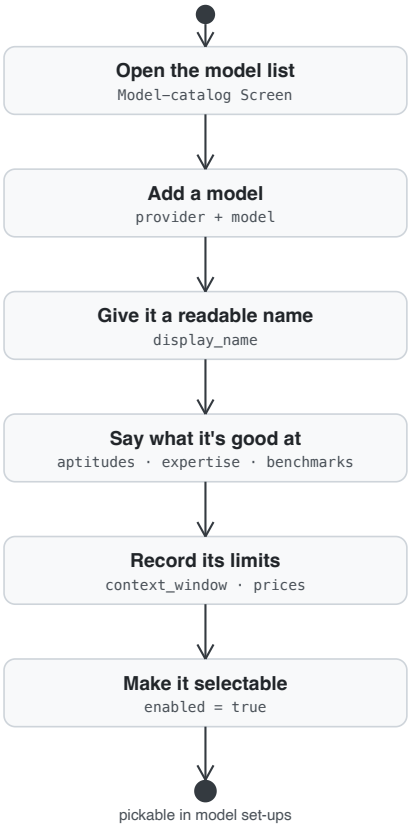
Variations and exceptions

IF…	THEN…
A model is no longer wanted	Set `enabled = false`: it stops appearing in new set-ups. Deleting is refused while any `model_config_model` references it, so existing set-ups keep working and finished runs stay reproducible.
The `provider` and `model` pair already exists	Refused: the pair is unique.
The API model id is mistyped	Nothing catches it at entry; the failure lands in `chat_round.error` the first time a run calls the provider.
`context_window` is left blank	Refused: nothing else in the system can discover it, and a wrong figure is not recoverable from a provider's reply.
`context_window` is overstated for one model	Every set-up containing it is over-filled: the history budget is the **smallest** window among a set-up's members, so one wrong figure silently mis-sizes every model it is paired with.

Acceptance test — *Given* an admin, *When* they add and enable a model, *Then* it appears in the model-set-up picker; *and* deleting a model that a set-up references is refused while `enabled = false` succeeds and hides it from new set-ups.

2 · The Journey, Screen by Screen

One screen; enabling is what makes it usable.



3 · In Plain English

Every AI model the install is allowed to use lives in one list. Experimenters never type a model name by hand; they pick from here.

That indirection is the point. It lets an admin control exactly which models are in play, and it lets the name shown in the interface stay friendly while the exact API identifier stays precise underneath.

You can record what each model is good at in two ways. Short tags and published benchmark scores are there for a person deciding what to try. One plain sentence describing the model's strengths does more work than that: when a study routes each question to whichever model suits it, that sentence is what the question is matched against, so it is worth writing carefully.

Two numbers are recorded because nothing else can discover them: what the provider charges per token, and how much text the model can read at once. The second one matters more than it sounds. When several models are compared, they all have to be given exactly the same conversation to work from, so the amount of history anyone gets is limited by the least capacious model in the group. Overstating one model's capacity therefore quietly affects every model it is used alongside.

When a model becomes obsolete you retire it rather than deleting it. Old studies referenced it, and those studies still have to make sense years later.

Fits the schema cleanly. `model_catalog` is referenced by `model_config_model.catalog_id`, which is why retirement is a flag rather than a delete. Of the descriptive columns only `expertise` is read by the system — the mixture-of-experts router matches each prompt against that one sentence, so a vague or missing one makes routing fall back on the study's judge; `aptitudes` and `benchmarks` are documentation for the person choosing.