

Launch a Run

Experimenter

Turn a study into a live run: pick a group, model set-up, and nudge, pass the overlap gate, and go live.

Screens

Launch-run launch-run.htmlCohorts cohorts.htmlModel-configurations model-configurations.html

Writes

one run experiment_runwho's enrolled run_enrollment

Key fields

the nudge nudged_idthe group cohort_idthe model set-up model_config_idlocked definitions RESTRICTkeep candidates? keep_candidatesstatus state

1 · Given / When / Then

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

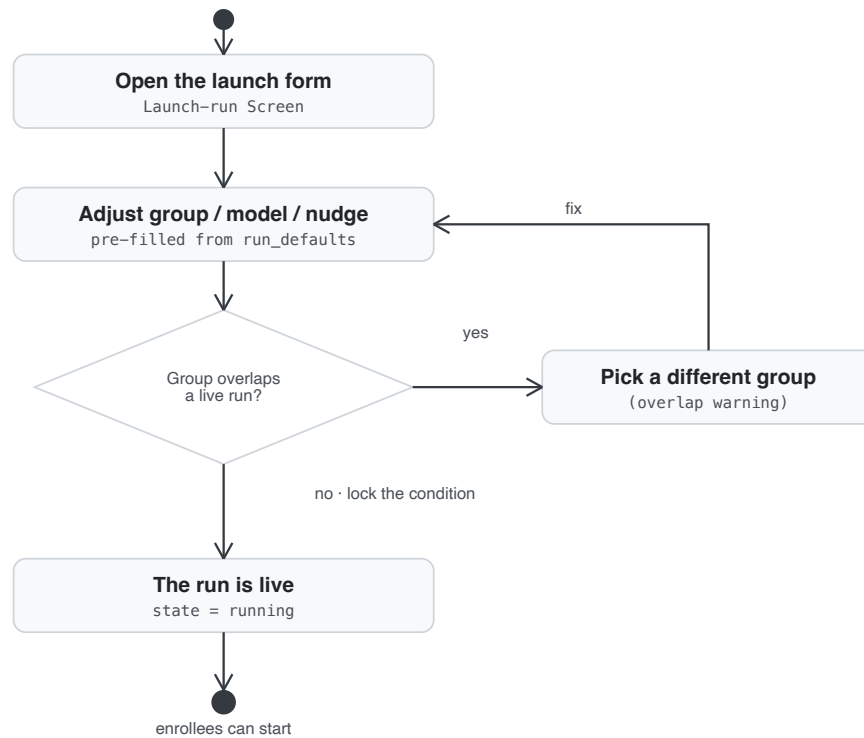
GIVEN	A study (<code>experiment</code>) with <code>run_defaults</code> , a reusable model set-up (<code>model_config</code>), and a cohort (<code>cohort</code>) all exist.
WHEN	On the Launch-run Screen the form arrives pre-filled from <code>experiment.run_defaults</code> (model set-up, nudge, and whether to keep candidates). The experimenter picks the target group, adjusts anything, optionally names the run and attaches context files, then clicks Launch.
THEN	The server first runs the disjoint-cohort check, which confirms none of the chosen people are already in a live run. If it passes, one run (<code>experiment_run</code>) is written with <code>state = running</code> , <code>started_at = now()</code> , <code>launched_by</code> set to the experimenter, and foreign keys to the nudge, model set-up, and cohort it uses (<code>nudge_id</code> , <code>model_config_id</code> , <code>cohort_id</code>). Those definitions, and the parent <code>experiment</code> , are held immutable while the run references them (<code>RESTRICT</code>), so later edits to the study or the model set-up can never change what this run used. Enrolled enrollees get <code>run_enrollment</code> rows. <code>reads</code> <code>run_defaults</code> , <code>cohort_member</code> , <code>experiment_run</code> <code>writes</code> <code>experiment_run</code> , <code>run_enrollment</code>

Variations and exceptions	
IF...	THEN...
Someone in the chosen group is enrolled in another run that is <code>running</code> or <code>paused</code>	The launch is refused with an overlap warning. End or wait for that run, or pick a non-overlapping group.
<code>run_defaults</code> names a model set-up that has since been deleted	The form asks for one. The default was never a foreign key, so nothing else is affected.
A context file is still extracting or embedding, or has failed	It is not attached to the run; only <code>ready</code> documents are.
The name is left blank	One is generated (study, group, model set-up, date-time) and frozen.

Acceptance test — *Given* a study, model set-up and non-overlapping group, *When* the experimenter launches, *Then* exactly one `experiment_run` exists with `state = running` and foreign keys to its nudge, model set-up, and cohort, and one `run_enrollment` row per enrolled enrollee.

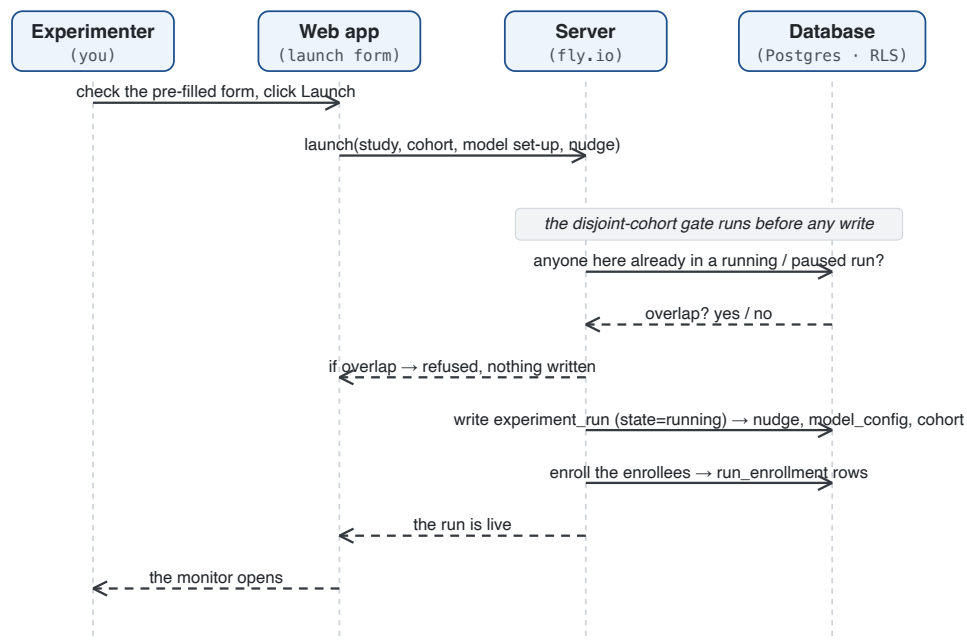
2 · The Journey, Screen by Screen

The overlap gate is the only thing between you and a live run.



3 · Behind the Scenes

The overlap gate and the write, step by step.



A dashed line is a reply. The gate query runs before any write; a live overlap means nothing is written at all.

4 · In Plain English

Launching is the moment a reusable study becomes one concrete, running thing: this group of people, this model set-up, this nudge, right now.

The form arrives already filled in from the study's defaults, so most launches are just “check and go.” The one thing that can stop you is the overlap rule. If any of those people are already inside another live study, ChatMaestro refuses, because their answers would be shaped by two conditions at once.

When it does go, the run is pinned to the exact definitions it uses — the study, the nudge, the model set-up, and the group. Those are locked for as long as the run refers to them. That is what makes the results trustworthy months later: you cannot rewrite the study or delete a model set-up out from under a run, so it always knows exactly what it used.

Fits the schema cleanly. The gate is a query over `run_enrollment × experiment_run.state`; the condition is fixed by the `RESTRICT` rules on the run's foreign keys, no copy needed. What happens next to the run is the state machine in *Run a nudge-comparison series*.