



Naming, Tagging and Not Drowning: Conventions for a Shared Account

Adoption · <https://docs.functionize.com/hc/en-us/articles/43802507417367-Naming-Tagging-and-Not-Drowning-Conventions-for-a-Shared-Account>

Studio accounts do not fail slowly because the product runs out of room. They fail slowly because nobody can find anything.

Here is what a real, actively used Studio account looks like after a few years: around 1,900 projects, 2,800 sessions, 1,260 orchestrations, 580 extensions, and a component library with names like `copy of copy of New Login`. Every one of those was created by somebody solving a real problem that day. None of them were named for the person who would need to find them two years later.

This page is the small set of conventions worth agreeing *before* the second team starts. All of them cost minutes. None of them can be retrofitted cheaply.



WHAT IS SHARED, AND WHAT IS NOT

Account-wide

Tags — the filter offers every tag anyone ever made.

Components — Add Component lists every one in the account.

Extensions — unless marked Private.

So these names are a shared vocabulary whether you treat them as one or not.

Scoped to a project

Tests — each one lives in exactly one project.

Execution presets — defined on a project, offered only to it.

Project boundaries are therefore the first thing to agree, because settings are inherited from here.

FOUR TAGS, AGREED ONCE

smoke

Must never break. Minutes, not tens of minutes. If this is red, nothing else matters.

core

The critical journeys. This is the set that gates a release.

slow

Legitimately long-running. Lets everyone else exclude them without an argument.

quarantine

Known flaky, deliberately excluded, **with an owner**.

Forty tags nobody uses is worse than four everybody does, because the filter becomes noise. Do not add a tag per team, per sprint or per person — **Created By** and **Project** already answer those.

NAME AFTER WHAT IT PROVES

Ages well

- *Guest checkout completes with a saved card*
- *Checkout smoke*
- *storefront — guest sign-in* (a component)

Becomes a lie

- *Test 14*
- *Nightly 2am* — the first time it moves
- *login* — in an account-wide library



What is shared, and what is not

This is the part most teams get wrong, because three of the most useful things in Studio are account-wide rather than project-scoped.

Scoped to a project

Tests (a test lives in exactly one project) and **execution presets** (defined on a project, offered only to that project).

Shared across the whole account	Tags — the Tags filter offers every tag anyone has ever created. Components — the Add Component list shows every component in the account, regardless of which project built it. Extensions — unless marked Private, they are available to the whole team.
--	---

So a tag is a shared vocabulary whether you treat it as one or not, and a component named `login` is a name your whole organization can only use once meaningfully.

Names: say what it proves, not when it runs

One rule covers most of it. **Name things after what they are for, not after a circumstance that will change.**

Tests	<i>Guest checkout completes with a saved card</i> beats <i>Test 14</i> and beats <i>Checkout</i> . The name should survive being read by somebody who was not in the room.
Orchestrations	<i>Checkout smoke</i> survives a schedule change. <i>Nightly 2am</i> becomes a lie the first time somebody moves it.
Projects	The application, not the person or the sprint. <i>Storefront</i> , not <i>Priya's tests</i> .
Components	Because these are account-wide, qualify them: <i>storefront</i> — <i>guest sign-in</i> , not <i>login</i> . A bare <i>login</i> in a shared library is a name collision waiting to happen.
Sessions	Generated titles are good, not perfect. Rename the ones that solved something hard, so the next person finds them.

A prefix turns a search box into a folder. Search on every list screen matches the title, so `Storefront -` on the front of forty tests makes them one keystroke away. This is worth more than any folder structure, because it works across screens.

Tags: three or four, agreed once

Tags are the highest-leverage thing in the product and the easiest to ruin. A scheme with forty tags that nobody uses is worse than four that everybody does, because the filter becomes noise.

A starting scheme that has to be actively argued against:

<code>smoke</code>	Must never break. Minutes, not tens of minutes. If this is red, nothing else matters.
<code>core</code>	The critical journeys. This is the set that gates a release.
<code>slow</code>	Legitimately long-running. Lets everyone else exclude them without arguing about it.
<code>quarantine</code>	Known flaky, deliberately excluded, with an owner. The alternative is a suite everyone has learned to ignore.

Add an application prefix only if you genuinely have several applications: `storefront:smoke` . Do not add a tag per team, per sprint or per person — those are questions the **Created By** and **Project** filters already answer.

Tags are editable straight from the Tests screen, so there is no excuse for tagging being a separate task. See *The Tests Screen*.

Ownership: every scheduled thing has a name against it

The single question that predicts whether a suite is still being read in six months is *who reads a red result, and by when?*

Make it answerable in the product rather than in somebody's memory:

- **Put the owner in the email alert, not in a wiki.** An orchestration's **Email Alerts** field takes a shared address, and **Email Note** carries one line of your own. Use it to say what the suite gates and who acts. See *Run Order, Re-runs and Alerts*.
- **Send to an alias, never a person.** A personal address becomes an unread mailbox the week they change team.
- **Audit quarterly.** Filter the orchestrations list to *Scheduled*, sort by *Created* oldest first, and read the top. That is the list of jobs running on a timer for reasons nobody remembers.

A convention for retiring things

Nobody deletes anything in a shared account, because deleting somebody else's work feels rude and deleting your own feels premature. The result is accumulation.

Two cheap habits fix most of it:

- **Sort the Tests screen by Last Run, ascending, once a month.** The bottom of that list is tests nobody runs. Either they matter, in which case they belong in an orchestration, or they do not, in which case say so.
- **Filter by Created By when somebody changes team.** Ownership is the thing that silently lapses, and this is the only screen that makes it visible.

Neither requires deleting anything. Both make it obvious what a decision would be.

The one-page version

Prefix names by application. Name after what a thing proves. Four tags, agreed. Qualify component names, because they are account-wide. Every schedule has an alias in its alert and a line saying what it gates. Review Scheduled and Last Run once a quarter.

Where to go next

- [Scaling From One Team to Many](#) — the wider organizational version of this, including project boundaries.
- [The Tests Screen](#) — where tagging and the ownership filters actually live.
- [Managing Orchestrations: Filters, Bulk Actions and Run History](#) — the quarterly audit, in one filter.
- [Keeping a Suite Green: Flake, Tiers and Who Owns Red](#) — what [quarantine](#) is for, and how to get tests back out of it.