



Exercise 8 - Tag a Suite and Give It a Schedule

Exercises · <https://docs.functionize.com/hc/en-us/articles/43802499884695-Exercise-8-Tag-a-Suite-and-Give-It-a-Schedule>

Lab handout — Lab 8

A printable worksheet for this exercise: the tasks, the exact prompts, a checkpoint and stretch challenges. Hand it round when you are onboarding a team.

[Download the PDF \(64 KB\)](#)

Exercise 6 built an orchestration by ticking tests from a list. That works the day you build it and quietly stops being right a month later, when three new tests should have been in it and nobody noticed.

This exercise fixes that. You will tag your tests, use the tag to find them, build a suite from what you find, and configure the settings that decide what happens when something goes wrong. About fifteen minutes.

You need at least two tests in a project. If you have done Exercises 1 and 2, you have them.

WHAT YOU WILL DO

- 1 **Tag from the Tests screen**
Add **smoke** to two tests without opening either one.
- 2 **Find them by tag**
Clear the project filter and set **Tags = smoke**. That is your suite, assembled by what the tests *are*.
- 3 **Build the orchestration**
Title, Projects, Tests, Schedule — but do not press Create yet.
- 4 **Open Advanced settings**
Four decisions. Switch run order to **Sequential** and watch **Halt after failure** appear; switch back and watch it go.
- 5 **Run it and read the history**
Total Runs, Avg. Time, Tests, and one row per run showing who triggered it.
- 6 **Clone it for the other data set**
Clone, rename, then **Execution Presets** — two suites over the same tests, different data, nothing duplicated.

THE TWO RE-RUN CHECKBOXES, IN STUDIO'S OWN WORDS

Incomplete tests	"Tests interrupted before finishing (timeout, crash)." Tick this one.
Failed tests	"Tests that ran to completion but returned a failure." Leave it clear — re-running asks the same question and hopes for a different answer.

Tag a suite, schedule it, and clone it for a second data set

 www.functionize.com

Step 1 — Tag from the Tests screen

Open **Tests** in the left sidebar. Use the **Project** filter to narrow to your project.

Find the **Tags** cell on one of your rows and click **Add tag**. Type `smoke`. Do the same on a second test.

You never opened a test. That is the point: tagging has to be a two-second job or it does not happen, and this is where it is a two-second job.

Tags are account-wide

The Tags filter offers every tag anyone in your organization has ever created, not just yours. So `smoke` means whatever your whole account has collectively decided it means. Agree a small set early — see *Naming, Tagging and Not Drowning*.

Step 2 — Find them by tag

Clear the Project filter. Now set **Tags** to `smoke`.

You are looking at your suite — assembled by what the tests *are* rather than by which ones you remembered. Add **Test Status = Failed** and you have the question that actually matters on a Monday morning: *is anything that gates a release broken?*

Step 3 — Build the orchestration

Open **Orchestrations** and choose **New Orchestration**.

1. **Title.** *Guest smoke*. Name it after what it proves.
2. **Projects.** Pick yours. The field takes more than one, but one is right here.
3. **Tests.** Add the two you tagged. They appear grouped under their project.
4. **Schedule.** Leave it on **On Demand**.

Do not click Create yet.

Step 4 — Open Advanced settings

Below the four fields are two collapsed sections. Ignore **Admin runtime settings** — it is for Functionize administrators. Open **Advanced settings**.

Four decisions live here, and the point of this step is to make them deliberately rather than by leaving the defaults alone.

Run order	Leave it on Parallel . Read the two descriptions and notice that picking Sequential makes a Halt after failure checkbox appear that did not exist a moment ago. Switch back to Parallel and watch it vanish again.
Re-run → Incomplete tests	Tick it. Hover the information icon: " <i>Tests interrupted before finishing (timeout, crash).</i> " Those almost never mean your application is broken.
Re-run → Failed tests	Leave it clear. Hover it too: " <i>Tests that ran to completion but returned a failure.</i> " Re-running those asks the same question again and hopes for a different answer.
Email Alerts	Put a real address in, and write something useful in Email Note — not "nightly regression" but what the suite gates and who acts if it is red.

Now click **Create**. A toast confirms it and offers **Run now**.

Step 5 — Run it and read the history

Run the orchestration. When it finishes, open it. You get three numbers across the top — **Total Runs**, **Avg. Time** and **Tests** — and two tabs.

- **Run History** has one row per run: result, duration, how long ago, and who triggered it. A scheduled run and a hand-started one are told apart here.

- **Tests** lists what is currently in the suite.

Avg. Time is the number people underuse. It moves slowly, and then one week it does not — and that is usually the application telling you something before any test goes red.

Step 6 — Clone it for the other data set

From the orchestration's row menu, choose **Clone**. Rename the copy.

Now open the clone's row menu and choose **Execution Presets**. Studio shows one line per project in the suite and states the rule: *"Choose which execution preset each project uses when this orchestration runs."*

If your project has more than one preset, pick a different one and save. You now have two suites over exactly the same tests, proving the same flows against different data, with nothing duplicated. If the dropdown only offers **Default**, that is fine — the project has no presets yet, and building one is a project-settings job described in [Execution Presets: Running One Suite Against Different Data](#).

Checkpoint

You should be able to say yes to all five:

- Both your tests carry the `smoke` tag, and you added them without opening either test.
- Filtering Tests by that tag shows exactly those tests.
- Your orchestration exists, has run once, and its history shows who triggered it.
- You can explain why **Halt after failure** is not on screen when Parallel is selected.
- You can say, in one sentence, the difference between an *incomplete* test and a *failed* one.

Stretch

- Set the clone's schedule to **Daily** and watch a time field and a day-of-week picker appear. Tick a single day — that is how you get a weekly job out of a product with no Weekly option. Then set it back to **On Demand** so it does not run unattended.
- On the orchestrations list, filter **Schedule** to *Scheduled* and sort by **Created**, oldest first. That is the quarterly audit in one filter.
- Add a `quarantine` tag to a test you do not trust, and take it out of the suite. Decide who owns getting it back in.

Where to go next

- [Run Order, Re-runs and Alerts: An Orchestration's Advanced Settings](#) — the full argument behind each of the four decisions in step 4.
- [Scheduling an Orchestration: The Six Cadences and What Each One Costs](#) — before you put anything on a timer.
- [The Tests Screen: Finding, Tagging and Fixing Tests at Scale](#) — the rest of what that screen does.
- [Keeping a Suite Green: Flake, Tiers and Who Owns Red](#) — what happens after the suite exists.