Workflows: Scheduled Jobs Written in Plain Language

Features · https://docs.functionize.com/hc/en-us/articles/43767914800279-Workflows-Scheduled-Jobs-Written-in-Plain-Language

A workflow is a recurring job described in plain language and run on a schedule. Where a test proves your application behaves correctly, a workflow just *does something* in a browser and reports back — checking a competitor's price, pulling a figure off a dashboard, confirming a third-party service is up.

Workflows, tests and orchestrations

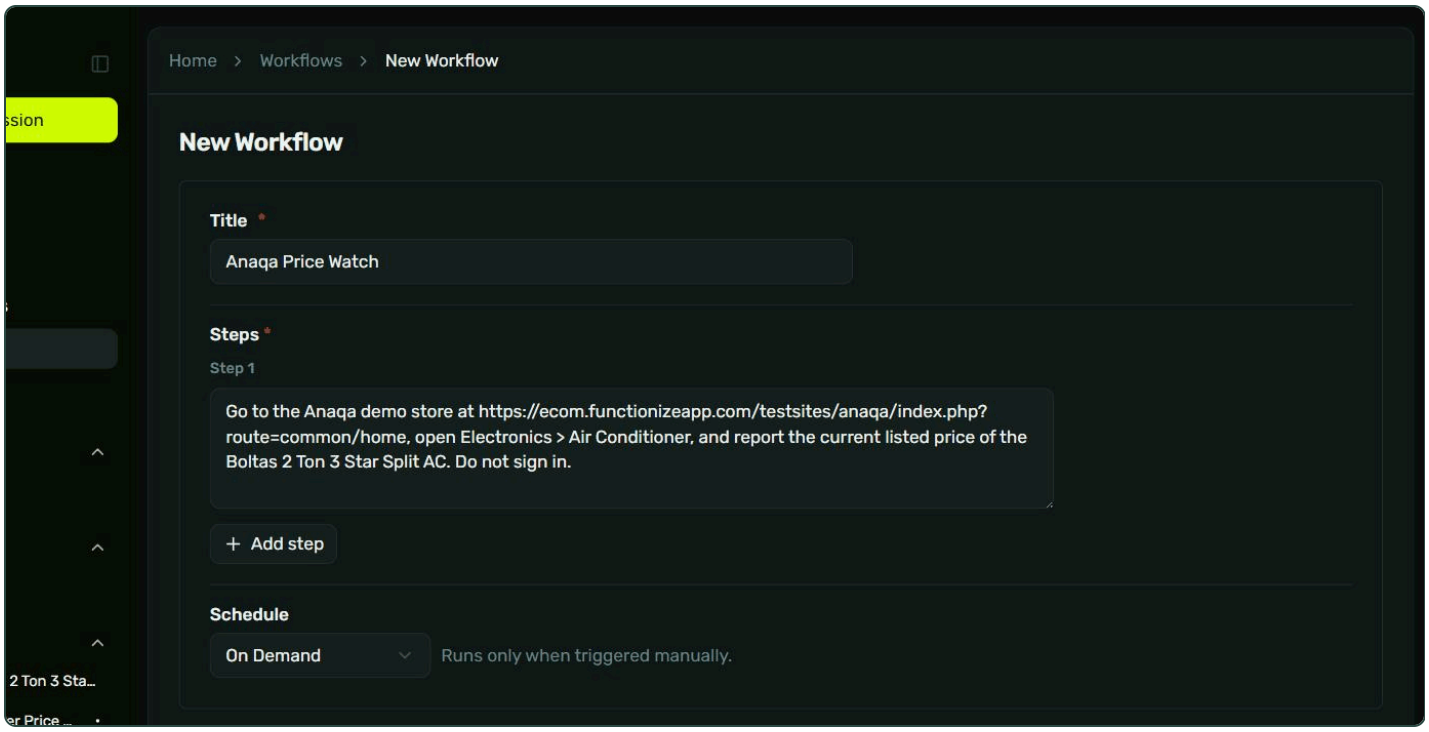
These three are easy to confuse, so it is worth being precise:

Test	Belongs to a project. Has steps and assertions. Passes or fails. This is the thing you build in Your First Test.
Orchestration	A named group of existing tests, run together as one job. Covered in Understanding a Test Run and Its Results.
Workflow	A standalone, scheduled task written as prose. Not tied to a project, and not pass/fail in the same sense — it produces a result.

If you find yourself asking someone to "just check that page every morning", that is a workflow.

Creating one

Open **Workflows** in the left sidebar and choose **New Workflow**.



Three things to fill in:

- **Title.** What the job is for.
- **Steps.** One or more free-text boxes — "*Describe what this workflow should do…*". Use **Add step** to break a longer job into ordered stages rather than writing one enormous paragraph. Each step carries three small controls on its right: **move up**, **move down** and **remove**, so the order is something you can change afterward rather than something you have to get right first time.
- **Schedule. On Demand**, **Hourly**, **Every 4 Hours**, **Every 12 Hours**, **Daily** or **Monthly**. On Demand runs only when you trigger it.

The scheduler here is the same one orchestrations use, and it has the same hidden depth: choosing **Daily** adds a time field and a day-of-week picker, which is how you get a weekly or weekdays-only job out of a product with no Weekly option. Scheduling an Orchestration covers the six cadences and what each one costs in full, and every word of it applies to a workflow.

Click **Create workflow**. The workflow gets its own page showing total runs, the schedule, who created it, the prompt, and a run history — with **Run**, **Edit** and **Delete** along the top.

What happens when it runs

Clicking **Run** starts an **agent session** and a background agent, exactly as creating a test does — you are taken to the session and can watch it narrate its way through the job. The run then appears in the workflow's history.

This shared machinery is why the advice in Exercise 3 — Read a Test Run and Its Evidence applies here too: read what the agent says it is doing, because that is where it tells you about anything ambiguous.

Writing a workflow prompt well

Say what you want reported, not just what to look at. "Open the pricing page" gives you nothing back. "Open the pricing page and report the listed monthly price for the Team plan" gives you an answer.

Be explicit about signing in. If the job does not need an account, say so — "do not sign in" keeps the agent on the public path and avoids it hunting for a login.

One job per workflow. If the title needs an "and", it is probably two workflows. They are cheap.

Be specific about the destination. Give the full URL. The agent checks reachability before it starts.

A caution worth reading

In testing this feature, a prompt asking the agent to "report the current listed price" led it to start building a *test* for that page before it noticed and corrected itself — it said so plainly in the session and redirected to reporting the figure.

That is the product being honest rather than the product being broken, but it tells you something useful: workflow prompts phrased like test descriptions can be read as test descriptions. If you want a report, ask for a report. Words like *verify* and *check that* lean toward test creation; *report*, *tell me* and *summarize* lean toward the answer you actually want.

Schedules cost runtime

The same warning as orchestrations applies, and more sharply, because a workflow is easy to set and forget. An hourly workflow runs 730 times a month. Start On Demand, confirm the output is what you want, then schedule it — and decide who reads the result before you do.

Cleaning up

A workflow's own page has a **Delete** button, so unlike some objects in Studio they are straightforward to remove. Delete the experiments you make while learning; a workflow list full of abandoned trials is how scheduled jobs end up running that nobody remembers creating.

Related

- Orchestrations: Running a Suite as One Job — for scheduled work that is running tests
- Projects: Boundaries, Defaults and What Inherits What — where the defaults come from