



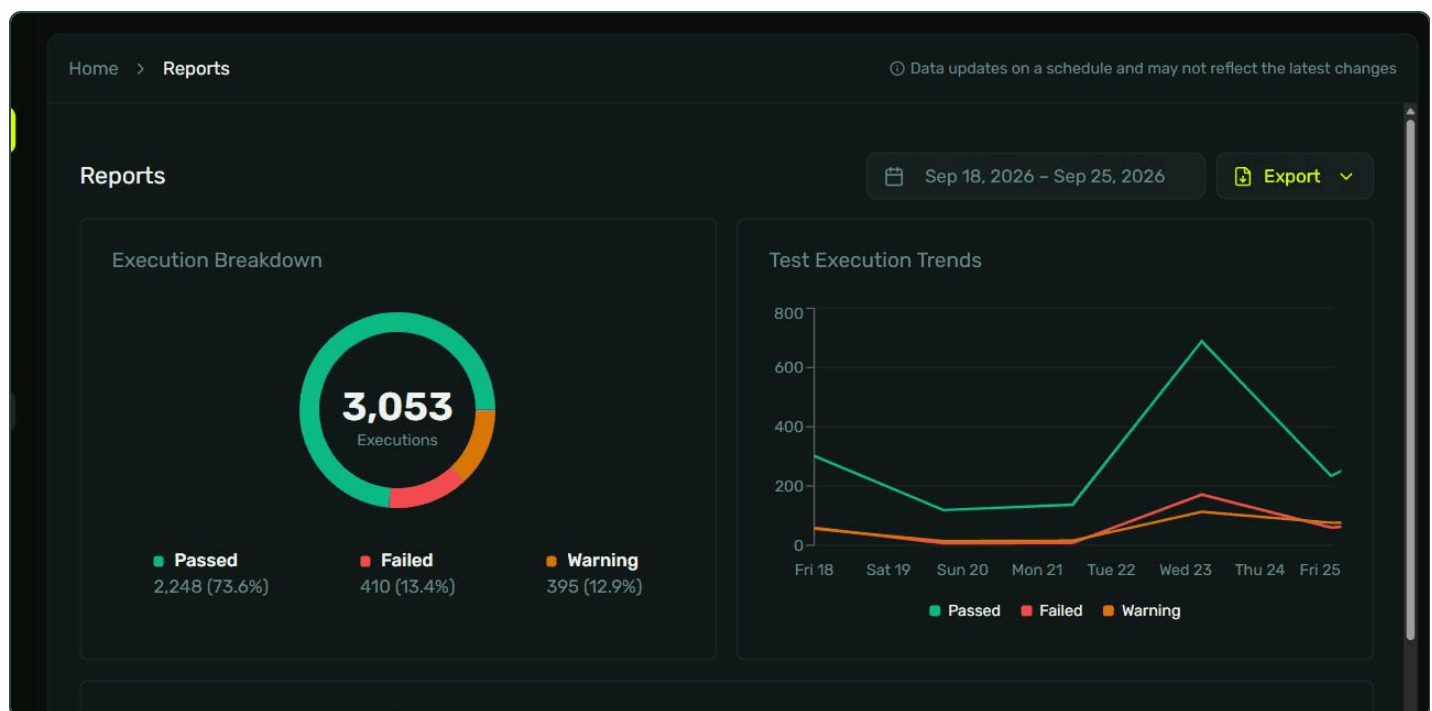
Reports: Pass Rates, Trends and What They Actually Tell You

Features · <https://docs.functionize.com/hc/en-us/articles/43767870492951-Reports-Pass-Rates-Trends-and-What-They-Actually-Tell-You>

Reports answers one question across everything you run: is testing getting better or worse? It is the only screen in Studio that looks across projects rather than at a single test.

What is on the page

Open **Reports** in the left sidebar.



Three things, plus a date range and an export:

- **Execution Breakdown** — total executions for the period, split into **Passed**, **Failed** and **Warning**, with counts and percentages.
- **Test Execution Trends** — the same three series plotted per day, so you can see whether a bad day was a blip or the start of something.
- **Pass Rate Breakdown by Project** — a table underneath, listing every project with its tests, passed, failed, warning and pass rate, plus an overall figure.

Warning is not Failed

The third category is the one people skip past. A **Warning** is not a pass and it is not a failure — it is a run that completed with something worth looking at.

If your warning count is comparable to your failure count, you have a body of results nobody is triaging. Warnings are where slow degradation hides: they do not break the build, so they accumulate quietly until a warning becomes a failure at the worst moment.

The date range and export

The range is set from the picker at the top right and is carried in the page URL, so a particular window can be bookmarked or pasted to a colleague. **Export** takes the data out for a spreadsheet or a deck.

A week is usually the right window for a stand-up, and a month for a trend. A single day tells you almost nothing, because one large orchestration can dominate the totals.

The caveat printed on the page

At the top of Reports, Studio tells you plainly:

Data updates on a schedule and may not reflect the latest changes.

Take that seriously. Reports is an analytics view, not a live console. If you have just run a suite and want to know how it went, open the orchestration or the test — the per-run evidence is immediate and complete. Reports is for the shape of things over days and weeks.

This also means Reports is the wrong place to verify a fix. Repair a test, re-run it, and read the run itself, as described in [Understanding a Test Run and Its Results](#).

Reading the pass rate honestly

A pass rate is only as meaningful as the tests behind it, and it is easy to flatter.

A project at 100% may have one trivial test. The table shows the test count beside the rate for exactly this reason — read them together.

A high pass rate with few assertions proves little. Tests that click through without checking anything pass reliably and tell you nothing; that is the subject of [Exercise 2](#).

A falling pass rate is not automatically bad. It often means someone added genuinely demanding tests. Look at what changed before treating it as a regression.

What to do with it

1. Once a week, look at the trend rather than the number. Direction beats level.

2. Sort the project table by pass rate and look at the bottom. Those projects either have real problems or abandoned tests — both worth acting on.
3. Check the warning share. If it is growing, someone needs to triage.
4. When a project looks wrong, stop using Reports and go to the failing tests themselves. See [Diagnosing a Failed Test in Functionize Studio](#).

A suite nobody reads manufactures confidence. Reports is the cheapest way to notice that is happening.

Related

- [Orchestrations: Running a Suite as One Job](#) — what produces most of these numbers
- [Projects: Boundaries, Defaults and What Inherits What](#) — why the per-project split looks as it does
- [What to Measure: QE Metrics and KPIs with Studio](#) — which of these numbers is worth reporting