



What's New in Functionize Studio

What's New · <https://docs.functionize.com/hc/en-us/articles/43804951871511-What-s-New-in-Functionize-Studio>

Everything below has shipped and is available to you now. This page is updated every Friday, newest first, so you can see what changed without reading a release note that assumes you already know the product.

Each entry says what it is, and then the part that actually matters — what it lets you stop doing. If something here is new to you and you want the full walkthrough, the links go to the article that covers it properly.

Where the engineering effort is going. Functionize Studio is where the product work happens. The entries below are a running record of that, and the cadence is deliberate — small, useful changes, every week, rather than a release every six months that nobody has time to absorb.

September 2026

A month of work on the Studio editor, on what the agent can build for you, and on getting Studio into applications that resist automation.

Reorder instructions and actions by dragging them

Drag and drop now works at both levels of a test. Move a whole instruction, or move a single action inside one, without rebuilding the sequence around it.

This is the change most likely to save you time on a test you are actively shaping. Getting the order right used to mean describing the order you wanted; now you can just put it there.

See [Operating the Steps Pane](#).

Collapse what you are not working on

Instruction rows, action cards, and loop and conditional blocks all collapse now, with a badge on hover to expand them again. Collapsed rows keep a status indicator, so you can fold a long test down to its shape and still see which parts are healthy.

Anything carrying a warning expands on its own, so folding a test away never hides the part you needed to look at.

Components can be parameterized by asking

Build a component once, then ask the agent to parameterize it — a login flow that takes a user, a search flow that takes a term. Previously this was something you set up by hand after the component existed.

Combined with edits propagating to every test that uses the component, this is what makes a component library worth keeping rather than a set of copies that drift apart.

See [Naming, Tagging and Not Drowning](#).

Studio recognizes bot protection and tells you what to do about it

When a site refuses automated traffic, Studio now names that as the reason rather than leaving you to work it out, and offers Quickconnect where Quickconnect is the answer. It also explains directly why an address on your own machine cannot be reached from a hosted browser.

These two account for a large share of the time people lose in their first week against a real application, and both now explain themselves at the moment they happen.

See [Before You Point Studio at Your Own Application](#).

The agent can resize the browser, and read through paginated tables

Two additions to what the agent can build. It can **resize the browser**, so a test can prove a layout works at a phone width as well as a desktop one. And it can **search a paginated table for a named value**, walking the pages itself instead of needing a step per page.

The second one matters more than it sounds. Any journey that ends in "find my order in this list" was awkward to describe before.

Orchestrations can span projects from the command line

Orchestration updates now work across projects from the CLI, and an orchestration can report its current configuration back to you. If you drive Studio from your own AI tool over MCP, that means you can ask what an orchestration is set to and change it without opening the interface.

See [Managing Orchestrations](#) and [Getting Started with the Functionize MCP Server](#).

Smaller things worth knowing

Attachments up to 10 MB	The limit on files you attach to a session has gone up, with a longer upload window to match.
"Chat" is now "Session"	One word throughout Studio, matching what the knowledge base has always called it.
More in the tests API	A test now reports its frequency, its URL, its resource type and its last execution job identifier, so an integration can read them without a second call.
A first-run dashboard	New accounts land on something that teaches rather than a modal to dismiss, and accounts that already have work no longer see the introduction at all.

August 2026

What landed through the end of August.

Orchestrations can now be driven entirely by the agent

Orchestrations used to be something you built in the interface and the agent knew nothing about. That gap is closed. You can now **create, edit and schedule an orchestration by asking for it**, and — the more useful half — you can **ask about one**. Orchestration-level metrics and failure detail come back in the conversation, so "what failed in last night's regression run, and why" is a question rather than a report you go and assemble.

Previously the agent could only see and act at the level of a single test case. Now it works at the level of the suite, which is the level most people actually plan at.

See [Orchestrations: Running a Suite as One Job](#) and [Managing Orchestrations](#).

Components are fully supported, and an edit reaches every test that uses one

Components — the reusable flows you build once and drop into many tests, which you may know from other tools as page objects — are now fully supported by the agent. The important behavior is propagation: **edit a component once and the change reaches every test case that uses it**, rather than staying local to the test you happened to be looking at.

That is the difference between a component library that pays for itself and one that quietly rots. If your login flow changes, you change it once.

See [Naming, Tagging and Not Drowning](#) for how components, tags and projects are scoped, and [Operating the Steps Pane](#) for building one.

Chat history now persists per test case

Start a new session on a test and the earlier conversation is still there. Prior sessions are no longer lost, so the reasoning behind a test — why a step was phrased a particular way, what you already tried — stays attached to the test rather than to whoever happened to be in the room.

This one is small and it removes a real daily irritation. It also makes handover between two people on the same test a genuine possibility.

See [Sessions: The Conversations Behind Your Tests](#).

Custom reporting: build it, filter it, export it, and diagnose without leaving it

Reports are now yours to shape. Build a report, **filter it by column**, and **export it** for whoever needs it outside Studio. Failures can be **diagnosed inline** from the report itself, so the path from “this number looks wrong” to “here is the cause” no longer runs through three screens.

See [Reports: Pass Rates, Trends and What They Actually Tell You](#).

How this page works

Updated	Every Friday morning. A quiet week gets a short entry rather than a padded one.
What is listed	Capabilities that have shipped and that you can use today. Things still in testing are not listed here, because an entry you cannot act on is not news.
What is not listed	Internal work, infrastructure, and routine maintenance. If a change does not alter what you can do, it does not earn a line.
If something here is new to you	Follow the link on the entry. The knowledge base article is the full version; this page is only the notice that it changed.

Where to go next

If you are new, start with [Start Here](#) and work through the [Onboarding](#) path. If you are already running tests and want to get more out of the account, the [Adoption](#) section is the one written for you.