



Start Here

Studio Basics · <https://docs.functionize.com/hc/en-us/articles/43778581058327-Start-Here>

Everything we have written about Functionize Studio, on one page. Pick the box that matches what you are trying to do.

New to Studio? Work down the left column: **Studio Basics** to understand it, then **Exercises** to actually build something.

Studio Basics

What Studio is, how to build a test, how to read a run, and how to diagnose a failure.

- [What Is Functionize Studio?](#)
- [Getting Started](#)
- [Starting a Session: The Studio Home Screen](#)
- [Your First Test in Functionize Studio: A Visual Walkthrough](#)
- [Understanding a Test Run and Its Results in Functionize Studio](#)
- [Operating the Steps Pane: Instructions, Actions and Components](#)
- [and 4 more...](#)

What's New

What has shipped in Studio, newest first. Updated every Friday, so it is worth a look even if you already know your way around.

- [What's New in Functionize Studio](#)

Onboarding

New to Studio? Start here. Your first 30 minutes, getting it into your application, what to automate first, what to measure, and how credits work.

- [A Map of Studio: Every Screen and What It Is For](#)
- [Your First 30 Minutes with Functionize Studio](#)
- [Before You Point Studio at Your Own Application](#)
- [Choosing What to Automate First](#)
- [Your First Two Weeks: An Onboarding Plan](#)
- [What to Measure: QE Metrics and KPIs with Studio](#)
- [and 1 more...](#)

Adoption

Taking Studio across an enterprise: how the pieces fit, bringing your team in, journeys that cross several web applications, and scaling from one team to many.

- [How It All Ties Together: Studio Across an Enterprise](#)
- [Inviting Your Team and Working Together in Studio](#)
- [Naming, Tagging and Not Drowning: Conventions for a Shared Account](#)
- [Testing End-to-End Across Several Web Applications](#)
- [Scaling From One Team to Many](#)

Exercises

Seven self-paced exercises against our demo store, each with a printable lab handout. No sign-in needed.

- [Start Here: The Functionize Studio Exercise Series](#)
- [Exercise 1 - Create Your First Test From a Prompt](#)
- [Exercise 2 - Write Assertions That Prove Something](#)
- [Exercise 3 - Read a Test Run and Its Evidence](#)
- [Exercise 4 - Change a Test by Asking](#)
- [Exercise 5 - Break a Test on Purpose, Then Diagnose It](#)
- [and 6 more...](#)

Best Practices

How to use Studio well: picking the right candidates, keeping the testing agent independent of the building agent, writing prompts that last, and stopping a good suite from rotting.

- [What Makes a Good Automation Candidate](#)
- [Writing Prompts That Produce Durable Tests](#)
- [Reviewing AI-Generated Tests: What a Human Still Has to Check](#)
- [The Independent Tester Principle: Never Build and Test in the Same Session](#)
- [Setting Up an Adversarial Test Loop With Claude and the Functionize MCP Server](#)
- [Keeping a Suite Green: Flake, Tiers and Who Owns Red](#)

Templates

Test strategy documents you can download, edit and put your own name on — Master, Integration, Regression and User Acceptance, in Word and PDF, plus a set for an SAP S/4HANA transformation.

- [Test Strategy Templates You Can Use Today](#)
- [Master Test Strategy Template](#)
- [Integration Test Strategy Template](#)
- [Regression Test Strategy Template](#)
- [User Acceptance Test Strategy Template](#)
- [Testing an SAP S/4HANA Transformation](#)
- [and 4 more...](#)

Value Calculators

Spreadsheets that turn a testing decision into a number a finance reviewer will accept — in Excel and Google Sheets, with a tear sheet on how to run each one.

- [Value Calculators: Building the Business Case](#)
- [Manual to Automated: What the Change Is Worth](#)
- [Baseline and Improve: Valuing a Change in Tooling](#)

Features

How individual capabilities work — self-healing, workflows, extensions, test data and reporting.

- [Projects: Boundaries, Defaults and What Inherits What](#)
- [The Tests Screen: Finding, Tagging and Fixing Tests at Scale](#)
- [Test Data: Execution Presets, Variables and Secrets](#)
- [Orchestrations: Running a Suite as One Job](#)
- [Scheduling an Orchestration: The Six Cadences and What Each One Costs](#)
- [Run Order, Re-runs and Alerts: An Orchestration's Advanced Settings](#)
- [and 6 more...](#)

Settings

Every setting, what it does, and which level to change it at.

- [Settings: Where Everything Lives and What to Change First](#)
- [Timeouts, Access and Browser Behavior: Getting Studio Into Your Application](#)

Skills

Ready-made skills for Salesforce, SAP, Workday, ServiceNow and more — plus how to build one for your own application.

- [Why Context Matters, and How to Give Studio More of It](#)
- [Why Skills Matter — Giving Your AI Agent the Context It Needs](#)
- [Functionize Skills: What Is Available and How to Install Them](#)
- [How to Build a Skill in Claude for Your System Under Test](#)
- [Complimentary Skill Building for Your System Under Test](#)

Model Context Protocol (MCP)

Connect Claude Code, Cursor, VS Code, Gemini CLI, ChatGPT and others to Functionize.

- [Getting Started with the Functionize MCP Server](#)
- [Connect Claude Code to the Functionize MCP Server](#)
- [Connect Claude Desktop to the Functionize MCP Server](#)
- [Connect Cursor to the Functionize MCP Server](#)
- [Connect VS Code and GitHub Copilot to the Functionize MCP Server](#)
- [Connect Gemini CLI to the Functionize MCP Server](#)
- [and 4 more...](#)

Non-Public Application Testing

Testing applications that are not reachable from the public internet.

- [Functionize Quickconnect \(ACS Quickconnect\)](#)
- [ACS Quickconnect Data-flow Diagram](#)

Glossary

The vocabulary a team needs before it can argue productively about testing, AI and what "done" means.

- [AI Testing and AI-DLC: A Glossary](#)
- [The AI-Driven Development Lifecycle: Terms](#)
- [Agentic Systems: Terms](#)
- [Evaluating and Testing AI: Terms](#)
- [Quality Engineering and AI Assurance: Terms](#)

FAQ

The questions people actually ask in their first weeks, answered, with links to the detail.

- [Frequently Asked Questions](#)

If you only read three things

1. [Your First Test in Functionize Studio](#) — the whole loop, in pictures.
2. [Diagnosing a Failed Test](#) — the one skill that separates people who trust their suite from people who ignore it.
3. [How to Build a Skill](#) — the biggest single lever on the quality of generated tests.

Working through it as a team

Each of the seven exercises has a printable lab handout attached to it — the tasks, the exact prompts, a checkpoint and stretch challenges. Download them and hand them out when you are onboarding people.

This index is generated from the live documentation, so it stays in step as articles are added. 91 articles across 15 areas.