TestTale

Quick Help

Short, step-by-step guides for every part of TestTale: what each feature is for and how to use it. The same guides are in the app under **Help** in the sidebar.

1. Getting started
2. Write automated tests from a ticket, pull request or failed build
3. Review a run and open a pull request
4. Ask the assistant
5. Keep test cases in the library
6. Run a test plan by hand
7. Check a story before it's built
8. Run tests on a schedule and spot flaky tests
9. Use your own AI model
10. Connect GitHub, Jira and other tools
11. Set up a project
12. Invite your team
13. Plans, trial and billing
14. See who did what

1. Getting started

Where: Home

Set up your workspace once, then use the Home page as your daily starting point.

1. Open Home. The “Get set up” checklist shows what's left to do; each step is a link that takes you to the right place.
2. Add an AI key (Settings → AI models). Until you do, a demo agent gives sample answers so you can look around.
3. Connect GitHub so TestTale can read your code and open pull requests. Connecting Jira is optional but lets it read stories and file bugs.
4. Point the project at your own repository (Settings → Projects).
5. Start your first run from the box at the top of Home.

TIPS

- Below the checklist, “Needs attention” lists what's waiting for you: runs to review, approvals, failing schedules, flaky tests and more.
- You can hide the checklist with the × button once you're done with it.

2. Write automated tests from a ticket, pull request or failed build

Where: Home

Paste a Jira ticket, a GitHub pull request link, a failed CI run link or a plain description, and TestTale writes and runs the tests for you.

1. On Home, type or paste into “What should we test?”: a Jira key like PROJ-123, a pull request link, a failed GitHub Actions run link, or a sentence describing the feature.
2. Optionally pick the test framework and the AI model under the box.
3. Click “Generate tests”. TestTale reads the ticket or code changes, writes the tests, runs them and fixes its own mistakes.
4. When it's done, the run says “In review” and the tests are ready for you to check.

TIPS

- Add extra instructions after the key, e.g. “PROJ-123 include security edge cases”.
- If a test fails because the app itself is broken (not the test), TestTale flags it as a product bug instead of hiding the failure.

3. Review a run and open a pull request

Where: Runs

Check the tests TestTale wrote, ask for changes in plain words, then send them to your repository.

1. Open a run from Runs (or from “Waiting for your review” on Home).
2. Read the scenarios on the left and the test files in the editor. You can edit any file directly.
3. To change something, type it in the box (for example “add a test for an expired password”) and click “Send to agent”.
4. Click “Run tests” to run the suite again; failures are fixed automatically where possible.
5. When you're happy, click “Create PR”. Nothing goes to your repository until you do.

TIPS

- “Save to library” copies the run's scenarios into the Test library so manual testers can use them too.
- Each change to a file is kept as a version, so you can go back.

4. Ask the assistant

Where: Assistant

Chat with an AI that knows your workspace: runs, test cases, plans, reviews, regression results, Jira and your code.

1. Open Assistant and type a question, for example “Which tests are flaky?” or “What failed in the last run?”. Press Enter to send.
2. The line “Looked at: …” under an answer shows where the information came from.
3. Ask it to do things too: “Add test cases for password reset” or “File a bug for PROJ-123”.
4. Anything that changes something appears as a card. It only happens when you click “Approve”; click “Dismiss” to cancel.

TIPS

- Your conversations are private to you. Start a fresh one with “New chat”.
- Pick a different AI model for a conversation with the model menu under the message box.

5. Keep test cases in the library

Where: Test library

One place for all your test cases, written by people or saved from automated runs.

1. Open Test library. Use “New folder” to group cases (for example by feature).
2. Click “New test case”, give it a title, then add steps: what to do and what should happen.
3. Set the priority, add tags and link Jira issues if you like, then click “Create test case”. It gets a number like TC-12.
4. Use the search box and filters to find cases by text, number, tag, priority or Jira key.

TIPS

- Select several cases to move them to a folder or delete them together.
- Export puts the current list in a spreadsheet (CSV) file.

6. Run a test plan by hand

Where: Test plans

Group test cases into a plan, then work through them and record what passed and failed.

1. Open Test plans, click “New test plan” and name it (for example “Release 2.4”).
2. Click “Add test cases” and tick the cases you want.
3. Click “Start test run”. TestTale shows one case at a time with its steps.
4. Do the steps in your app, then mark the case Passed, Failed, Blocked or Skipped. Add notes if something looked wrong.
5. For a failure, “File bug in Jira” creates a bug with the steps and your notes. Click “Complete run” when you're finished.

TIPS

- Editing a test case later doesn't change past runs; they keep the version you tested.
- The plans list shows the result of each plan's latest run.

7. Check a story before it's built

Where: Story reviews

Let the AI read a user story and point out vague wording, missing cases and the tests you'll need.

1. Open Story reviews. Enter a Jira key, or switch to “Paste a story” and paste the text.
2. Click “Review story”. After a few seconds you get a testability score from 1 to 5.
3. Read the open questions, the cases the story doesn't cover, and the suggested acceptance criteria.
4. Tick the test ideas you want and click “Add … to the library”; they're added as draft test cases.
5. For a Jira story, “Post to Jira” shows the comment first and only posts it when you confirm.

TIPS

- A low score isn't bad news: it tells you which questions to ask before anyone starts coding.

8. Run tests on a schedule and spot flaky tests

Where: Regression

Run your existing automated tests every hour, day or week, and see which tests are unreliable.

1. Open Regression and click “New schedule”.
2. Name it, choose which test files or folders to run (leave it empty to run the project's tests folder), and pick how often and when.
3. Click “Create schedule”. It runs on its own from then on; “Run now” runs it immediately.
4. Watch the pass rate chart and the lists: “Flaky tests” (pass one time, fail the next), “Failing now” and “Slowest tests”.
5. Click a bar or a run to see each test's result and error message.

TIPS

- Hourly schedules need the Business plan; Team includes daily and weekly.
- Untick “On” to pause a schedule without deleting it.

9. Use your own AI model

Where: Settings → AI models

TestTale uses your company's AI account (OpenAI, Anthropic, Google or a compatible service), so you control the cost and the data.

1. Open Settings → AI models and click “Add key”.
2. Choose the provider and paste the API key from that provider's website. For other services, also enter their address (base URL).
3. Click “Check & save”. TestTale checks the key and lists the models it can use.
4. Choose the default model (and optionally a faster one for small jobs), then click “Save defaults”.

TIPS

- Keys are stored encrypted and are never shown again in full.
- You can still pick another model for a single run or chat.

10. Connect GitHub, Jira and other tools

Where: Settings → Connections

Connections let TestTale read tickets and code, and (when you approve) open pull requests or file bugs.

1. Open Settings → Connections.
2. Use the one-click GitHub and Jira buttons and sign in, if your administrator has set them up.
3. Or click “Add server” under MCP servers: pick GitHub, Atlassian or “Other server”, paste the address, and either a token or “Save & sign in”.
4. Each server's tools are listed as read or write. Read tools can be used automatically; write tools always wait for a person to approve.

TIPS

- Without connections, TestTale uses demo tickets and a demo repository, which is fine for trying it out.

11. Set up a project

Where: Settings → Projects

A project tells TestTale which repository to test, where the tests live and how to run them.

1. Open Settings → Projects and click “New project” (or “Edit” on the demo project).
2. Enter the repository owner and name, the main branch and the folder your tests live in.
3. Pick the test framework your team uses and the address of the app to test (for example your staging site).
4. Add any passwords or settings the tests need as run secrets; they are stored encrypted. Click “Save”.

TIPS

- To run tests on your own GitHub Actions instead of TestTale's sandbox, use “Add workflow” and follow the steps shown.

12. Invite your team

Where: Settings → Members

Add teammates to your workspace and choose what they can do.

1. Open Settings → Members.
2. Enter a teammate's email, choose a role and click “Send invite”. They get an email with a link to join.
3. Members can use every feature; admins can also change settings and members; the owner can do everything, including billing.

TIPS

- Each member is a paid seat; the bill adjusts automatically when people join or leave.

13. Plans, trial and billing

Where: Settings → Billing & usage

Every workspace starts with a 14-day free trial; after that, choose a plan to keep working.

1. Open Settings → Billing & usage to see your plan, days left in the trial and this month's usage.
2. Pick monthly or yearly (yearly is cheaper), then click “Choose Team” or “Choose Business” and pay in the secure checkout.
3. “Manage billing” opens the billing portal for invoices, payment details or cancelling.

TIPS

- If the trial ends or the subscription stops, the workspace becomes read-only: you can still view and export everything, but not start new work.
- Team: daily schedules, up to 10 projects. Business: hourly schedules, unlimited projects, longer audit history.

14. See who did what

Where: Settings → Audit log

The audit log records important actions, such as invites, connections, approvals and pull requests.

1. Open Settings → Audit log (admins and owners only).
2. Each line shows who did what, and when.

Still stuck? Use the support email on the Help page in the app.