

## Testim Automate

“Testim’s simple user interface makes it easy for our developers to add tests without a steep learning curve. We grew coverage by 30% in three months and found that very little maintenance is required so we can keep increasing coverage.”  
Totango

### The company

Testim is a software vendor that was established in 2014. It’s headquartered in California, and has an additional office in Israel. It boasts more than 50 employees as well as \$20M+ in funding and hundreds of customers. As of writing, its main product is Automate, an AI-driven test automation platform.

### What is it?

Testim Automate is a web-based functional test automation platform that uses AI to build highly resilient and automated UI tests. It enables you to accelerate test creation, minimise test maintenance, and integrate your test processes into your CI pipeline. It also has features to address scalability issues that typically arise as the number of tests and/or team members grows.

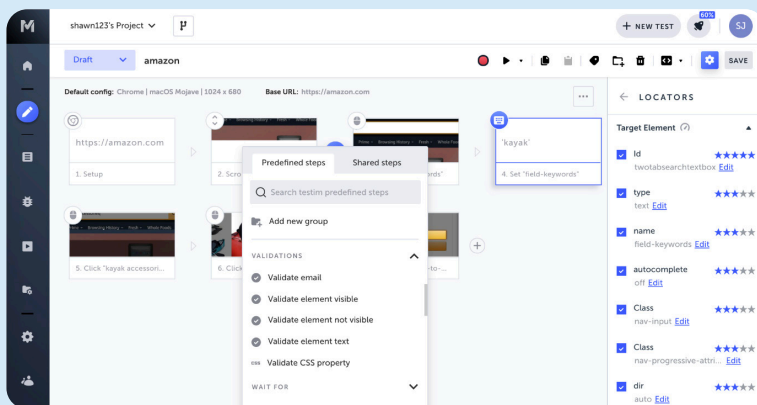
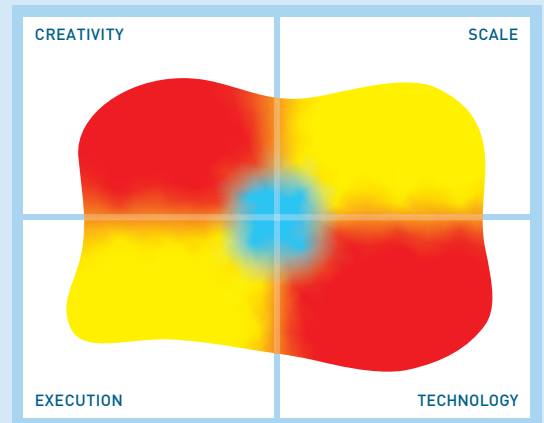


Figure 1 – Editing a test in Testim Automate

Automate exclusively supports the testing of web applications, including the mobile web. It integrates with a wide variety of CI tools as well as collaboration-oriented products, like Slack, Jira and Trello, and test management products, such as TestRail. Tests can run in parallel on Testim’s grid or on any Selenium-compatible grid, and you can even export Automate tests as Selenium-compatible code to avoid lock-in.

### What does it do?

Automate creates your tests as you use your application by using a Google Chrome extension to



The image in this Mutability Quadrant is derived from 13 high level metrics, the more the image covers a section the better. Execution metrics relate to the company, Technology to the product, Creativity to both technical and business innovation and Scale covers the potential business and market impact.

capture your actions as a series of test steps. These can then be configured into an automated test using Automate’s low-code visual editor (see [Figure 1](#)). This could include grouping steps together (forming a reusable test component), or adding validation, loops, parameterisation, and so on. You can further augment your tests with steps that leverage custom JavaScript and/or API calls, and to this end the platform comes with a full-fledged JavaScript editor, as shown in [Figure 2](#).

Automate distinguishes itself by automatically creating ‘smart locators’ for every element used in your test. These use AI to intelligently locate said elements at runtime for your tests to operate on, regardless of any stylistic or structural changes that have occurred in the meantime. While executing the test, each element’s properties and surrounding context are examined to find the matching element. Every part of any given smart locator, including its minimum confidence level, is fully transparent and can be adjusted if necessary. In effect, smart locators allow your tests to self-heal in the face of superficial changes, increasing resiliency and substantially reducing the need for maintenance.

A second major AI capability is the ability to automatically find steps that have been duplicated over multiple tests, then group them into reusable test components and refactor existing tests to accommodate them. This includes steps that have

Automation	★★★★★	Implementation	★★★★★
Collaboration	★★★★★	Integration (Third Party)	★★★★★
Ease of use	★★★★★	Integration (Vendor Platform)	★★★★★
Functionality	★★★★★	Maintenance	★★★★★

“ Before Testim, new tests in a large application were taking one to two days to write and make stable. The same tests in Testim take between 20 and 30 minutes. ”  
Outbrain

the same structure but use different data, in which case they will be automatically parameterised. This capability also helps prevent the creation of duplicate steps during test creation, by detecting patterns and suggesting existing groups to auto-complete the test case.

Automate also provides simple and effective root cause analysis: when a test fails, the platform will present screenshots for each test step up until the point where it failed, alongside the relevant console log, network activity, and ‘baseline’ screenshots from a previous passing test. Additionally, failed tests are grouped (and can be analysed) by failure type, and errors can be presented in aggregate to

save time triaging and prioritising work. Testim also suggests the failure cause and can learn from previously tagged failures to help identify recurring issues.

In addition to all of the above, Automate comes with a significant TestOps

- Grouped test steps enable reuse, and automated group discovery both reuses and refactors, reducing unnecessary duplication and driving collaboration.
- Smart locators make your tests far more robust, and can quite dramatically reduce time spent on maintenance.

Moreover, when your tests fail, you will be able to trust that it is because something has actually gone wrong with your system, not with your tests themselves. This should make implementing testing into your CI build processes far more palatable and efficient, by making it much less likely that your build will fail due to a need for test maintenance.

There are also other, perhaps less prominent, benefits. For instance, Testim caters well to both technical and non-technical users. It also emphasises TestOps, and with good reason: TestOps helps your testing to scale up (and shift left) in order to keep pace with ever-accelerating development pipelines, and thereby prevent your tests from becoming a bottleneck to your development.

The only aspect we care to criticise is the product’s relatively small amount of high-level design automation, usually exemplified by model-based testing or integration with the same. That said, the keyword here is ‘relative’: Testim already provides a number of the benefits of model-based design, including enabling improved test resilience, step reuse, and collaboration via its grouping and smart locator features, and it does so without the significant onboarding difficulties that tend to come with more fundamentally model-based approaches. Even so, we would be pleased to see this capability enter the picture in some fashion, perhaps through partnership with existing solutions.

### The Bottom Line

Testim Automate uses AI to help you build automated, robust, reliable, CI-ready, and ultimately trustworthy tests. It creates these tests quickly and with superb ease of use, while helping you manage and scale up your testing operation. If you want to revamp and accelerate your testing process, it is certainly a product worthy of your attention.

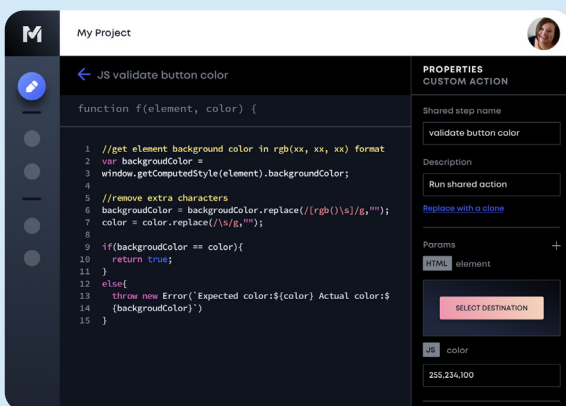


Figure 2 – Creating a JavaScript step in Testim Automate

capability. This includes full source control for your test assets, including branching, merging, pull requests, and conflict resolution; test management, via a centralised, searchable test library; and test insights, such as failure type and reporting trends. Notably, tests can be equipped with a test owner, and always come with a test status, which can be draft, active, quarantine, or ‘evaluating’. The latter is the most interesting, as it means the test will be deployed as part of your CI pipeline, but that its failure will not interrupt the build. As its name suggests, this is useful for evaluating tests before deploying them in full.

### Why should you care?

Testim Automate provides several notable benefits, over and above what is typical for a test automation product:

- Tests are easy and fast to create.

[FOR FURTHER INFORMATION AND RESEARCH CLICK HERE](#)