

[COMPLETE USER GUIDE](#)**Test any website in plain English**

Everything you need to know about RoboQA — from your first test to CI/CD pipelines and bug integrations.

[Open RoboQA →](#)[↓ Download as Word doc](#)**RoboQA**[RoboQA Agent](#)[Templates](#)[Bulk Test](#)**Test any website in plain English**

No code. No setup. Results in 60 seconds.

Website URL**What do you want to test?**

Auto — engine picks Quick or Deep based on your prompt

TRY AN EXAMPLE

RoboQA Agent — Test any website in plain English. No code. No setup. Results in 60 seconds.

CONTENTS

- | | |
|---------------------------------|------------------------------------|
| 01 What is RoboQA? | 09 Notifications |
| 02 Getting Started | 10 CI/CD Integration |
| 03 RoboQA Agent | 11 Third-Party Integrations |
| 04 Understanding Results | 12 Dashboard |
| 05 Test Templates | 13 Real-World Use Cases |
| 06 Bulk URL Testing | 14 FAQs & Troubleshooting |
| 07 Test Suites | 08 Saved Credentials |

1

What is RoboQA?

RoboQA is an AI-powered website testing platform that lets you verify any website's functionality using plain English — no code, no setup, no Selenium scripts required. You describe what you want to test, and RoboQA's AI agent navigates your site, performs the actions, and reports back with pass/fail results, bug details, and screenshots in under 60 seconds.

Whether you're a developer checking your latest deployment, a QA engineer running regression suites, or a product manager verifying a

client site, RoboQA gives you instant, reliable results without writing a single line of test code.

Key capabilities

- **Natural language testing** — describe flows like "verify login works and user can access dashboard"
 - **Bulk URL testing** — run the same test across up to 10 sites simultaneously
 - **Pre-built templates** — ready-made test cases for login, checkout, forms, accessibility, and more
 - **Test Suites** — group related tests and run them all in one click
 - **Saved Credentials** — securely store login details for automated auth testing
 - **Scheduled tests** — run tests automatically on a daily, weekly, or custom schedule
 - **CI/CD integration** — trigger tests from GitHub Actions, GitLab CI, or any pipeline via API
 - **Bug tracking** — bugs are detected, categorised by severity, and logged automatically
 - **Notifications** — get alerted via Slack or email when tests fail
 - **Integrations** — auto-create tickets in GitHub Issues or Jira when bugs are found
-

Getting Started

Creating an account

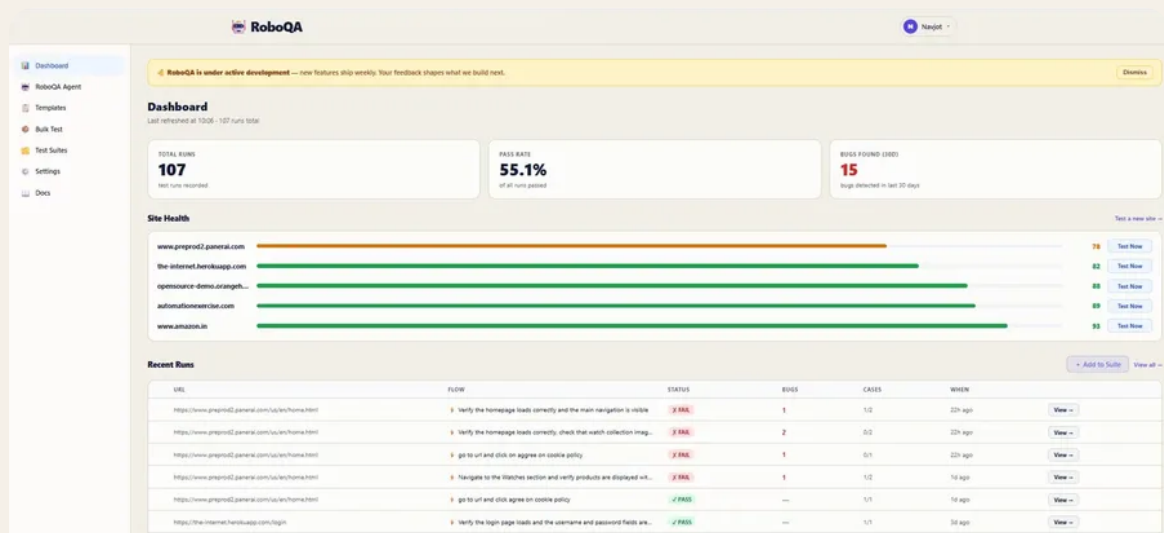
Visit roboqa.tech and click **Sign In / Sign Up** in the top-right corner. During the soft launch period, all new accounts start on the Free plan — no credit card required.

Account requirements

- Full name
- Email address
- Password — minimum 10 characters, must include uppercase, lowercase, number, and symbol

After signing up, check your inbox for a verification email. Click the link to activate your account, then sign in to access your dashboard.

Navigating the app



- **Dashboard** — total runs, pass rate, bugs found, and site health scores
 - **RoboQA Agent** — the main testing interface for individual tests
 - **Templates** — pre-built test prompts organised by category
 - **Bulk Test** — run the same test across up to 10 URLs simultaneously
 - **Test Suites** — manage and run grouped collections of test cases
 - **Settings** — credentials, notifications, integrations, schedules, and CI/CD setup
 - **Docs** — in-app documentation and CI/CD reference
-

3 RoboQA Agent

The RoboQA Agent is the heart of the platform. Enter a URL and describe what you want to test in plain English — the AI handles everything else.

How to run a test

- 1 Enter your **Website URL** — must include https://
- 2 Describe **what** in plain English, e.g. "verify the login flow works
to test and the dashboard loads correctly"
- 3 Choose **scan** : Auto (recommended), Quick, Standard, or
your **mode** Deep
- 4 Select your **viewport** : Desktop, Tablet, Mobile, or Mobile Large
- 5 Click **Run Test** — results appear in approximately 60 seconds

Scan speed settings

- **Auto** — RoboQA picks Quick or Deep based on your prompt complexity (recommended)
- **Quick** — faster scan, ideal for simple checks like page load and visible elements
- **Standard** — balanced depth and speed for most everyday tests
- **Deep** — thorough multi-step flows; best for checkout, forms, and complex auth flows

Viewport options

- **Desktop** — standard 1280px wide browser window
- **Tablet** — simulates tablet screen dimensions
- **Mobile** — simulates smartphone viewport (~375px)
- **Mobile Large** — simulates larger phone screen (~414px)

Example prompts to try

Login flow: "verify the login page loads and users can sign in with valid credentials"

Checkout: "add a product to cart and complete the checkout process"

Contact form: "fill in and submit the contact form and confirm the success message appears"

Performance: "verify the page loads within acceptable time with no errors"

Accessibility: "check the page for WCAG accessibility violations"

4

Understanding Your Results

After a test run completes, the results page gives you a full breakdown of what happened, what passed or failed, and what bugs were detected.

RoboQA

Dashboard

RoboQA Agent

RUN SUMMARY

Score **60/100**
Pass rate **100%**
Steps **1**
Bugs found **0**
Duration **—**

Run Result

single_url PASS https://the-internet.herokuapp.com/login

Run Summary

STATUS	STEPS	DURATION
PASS	1	—
✓ PASSED 1	✗ FAILED 0	BUGS 0 C:0 H:0 M:0 L:0
WEBSITE SCORE HOW IS THIS CALCULATED? — 60/100 TREND  35		
Show run ID		

Website Score

DOMAIN the-internet.herokuapp.com	60 / 100 SCORE
FLOW RELIABILITY 85	BUG DENSITY 100
UI ISSUES 0	SUCCESS RATE 1
FORM RELIABILITY 20	AUTOMATION SUCCESS N/A

Run Result page — summary, website score, and detailed sub-metrics

Run Summary panel

- **Status** — PASS or FAIL for the overall run
- **Steps** — total number of test steps executed
- **Duration** — time taken to complete the run
- **Passed / Failed** — count of individual steps
- **Bugs** — total bugs found, broken down by Critical (C), High (H), Medium (M), Low (L)

Website Score (out of 100)

FLOW RELIABILITY

85

Consistency of test flow completion

BUG DENSITY

100

Inverse of bugs found per run

UI ISSUES

0

Visual / layout problems detected

SUCCESS RATE

1

% of test cases that passed

FORM RELIABILITY

20

Form submission consistency

AUTOMATION SUCCESS

N/A

Whether automated steps succeeded

Test Steps

● **Step 1: Positive — Valid login credentials authenticate successfully**

PASS

Confirmed cleanly.

Before

After

No screenshot available

No screenshot available

[Open screenshot](#)

[Open screenshot](#)

What happened:

'scenario' ran — steps passed.

Expected:

'scenario' should pass all checks and produce a verifiable response.

Teach RoboQA this passing step for future runs on this site.

[Save step](#)

► Expanded details

Teach RoboQA all 1 passing step for future runs on this site.

[Save all 1 passed step](#)

Page Health

1 page health warning found after the last scenario:

- **Broken images (1)** detected on initial page load — images not visible within 2-3 seconds affect real users on slow connections [low]

Test Steps with pass/fail detail, Page Health warnings

Test Steps

Each step in your test is shown with its result, before/after screenshots (where available), what happened, and what was expected. For passing steps, click **Save step** to teach RoboQA what correct behaviour looks like for that site — improving future accuracy.

Page Health

Below the test steps, the Page Health section flags background issues detected during the run — broken images, slow assets, missing alt text, etc. — that aren't directly related to your test prompt.

Actions

[Run Another Test](#)[Copy Share Link](#)[Open Public Report](#)[Retest After Fix](#)[Open Bug Dashboard](#)[Download CSV](#)[Download PDF](#)

Scheduled Testing

[Daily](#)[Save Schedule](#)

AI Suggestions

Async Loading

Add explicit loading-state waits and optimize heavy assets on pages exceeding threshold.

Test Steps

Step 1: Positive — Valid login credentials authenticate successfully

[PASS](#)

Confirmed cleanly.

Before

After

No screenshot available

No screenshot available

[Open screenshot](#)[Open screenshot](#)

What happened:

'scenario' ran — steps passed.

Expected:

'scenario' should pass all checks and produce a verifiable response.

Teach RoboQA this passing step for future runs on this site.

[Save step](#)

Actions panel, AI Suggestions, and Scheduled Testing from the results page

Actions from the results page

- **Run Another Test** — start a new test immediately
- **Copy Share Link** — get a shareable URL for this run's results
- **Open Public Report** — view a clean, client-friendly version of the report
- **Retest After Fix** — re-run the exact same test after making changes
- **Open Bug Dashboard** — jump to the bug tracker for this domain
- **Download CSV** — export all results data as a spreadsheet
- **Download PDF** — save the full report as a PDF file

Scheduled Testing: From any results page, set up recurring tests for that URL. Choose Daily, Weekly, or another interval and click Save Schedule. RoboQA re-runs the test automatically and notifies you if anything changes.

5 Test Templates

The Templates page gives you a library of pre-built test prompts organised by category. Select a template and it's instantly ready to use — no prompt writing required.

Test Templates

Pick a pre-built template and run it instantly — no prompt writing needed.

All

Authentication

E-commerce

Forms

Navigation

Mobile

Performance

Accessibility

API



AUTHENTICATION

Login Flow

Verify users can log in and access their account

login auth credentials

Use Template →



AUTHENTICATION

Sign Up Flow

Test new user registration end to end

signup register email verification

Use Template →



E-COMMERCE

Checkout Flow

Test the full purchase flow from cart to confirmation

checkout payment ecommerce cart

Use Template →



NAVIGATION

Search Functionality

Verify search returns relevant results

search navigation

Use Template →



FORMS

Contact Form

Test contact or inquiry form submission

form contact submit

Use Template →



AUTHENTICATION

Password Reset

Test forgot password email flow

password reset email

Use Template →



MOBILE

Mobile Navigation

Test hamburger menu and mobile navigation

mobile navigation hamburger

Use Template →



PERFORMANCE

Page Load & Performance

Check page loads fast with no errors

performance load speed

Use Template →



ACCESSIBILITY

Accessibility Audit

Check for WCAG accessibility violations

accessibility wcag a11y

Use Template →

Test Templates library — searchable and filtered by category

Available template categories

- **Authentication** — Login Flow, Sign Up Flow, Password Reset
- **E-commerce** — Checkout Flow
- **Forms** — Contact Form
- **Navigation** — Search Functionality
- **Mobile** — Mobile Navigation (hamburger menu testing)
- **Performance** — Page Load & Performance
- **Accessibility** — Accessibility Audit (WCAG checks)
- **API** — API endpoint testing

Using a template

- 1 Browse or search for the template you need
- 2 Click **Use Template** → to load the prompt into the Agent
- 3 Enter your URL and click **Run Test**

Templates can be filtered by category using the tag pills at the top of the page, or searched by keyword.

6

Bulk URL Testing

Bulk Test lets you run the same test prompt across up to 10 different URLs simultaneously — perfect for agency workflows, multi-region deployments, or competitive benchmarking.

RoboQA
RoboQA Agent
Templates
Bulk Test

Bulk URL Test

Run the same test prompt across multiple URLs simultaneously.

💡 Test your homepage across 5 languages

💡 Check all your client sites in one click

💡 Compare 10 competitor sites for the same feature

URLs to test2/10 URLs

https://example1.com

https://example2.com

+ Add URL

What should we test on each URL?

Example: verify the login flow works and the dashboard loads correctly

Viewport

Desktop

Run All

Tests run in parallel — may take 2-5 min per URL

Bulk URL Test — add up to 10 URLs, tests run in parallel

How to run a Bulk Test

- 1 Navigate to **Bulk Test** in the sidebar
- 2 Add URLs using the input fields (up to 10). Use **+ Add URL** to add more rows
- 3 Write your test prompt in "What should we test on each URL?"
- 4 Choose Desktop, Mobile, or Tablet viewport
- 5 Click **Run All** — tests run in parallel (2-5 minutes per URL)

Suggested use cases

- Test your homepage across 5 language variants in one click
- Verify all your client sites work after a platform update
- Compare 10 competitor sites for the same feature or flow
- Regression-test staging, QA, and production environments simultaneously

7 Test Suites

Test Suites let you group related test cases together and run them all with a single click. Think of them as saved test plans that you can revisit, expand, and automate.

Test Suites

Group test cases into suites and run them all in one click.

[+ New Suite](#)

Core Engine Tests

10 cases

May 26, 2026

[▶ Run All](#)[Rename](#)[Delete](#)

test

1 case

May 24, 2026

[▶ Run All](#)[Rename](#)[Delete](#)

test

NAME	URL	PROMPT	VIEWPORT
press the Enter key and verify the result shows which key was pressed	https://the-internet.herokuapp.com/key_presses	press the Enter key and verify the result shows which key was pressed	desktop ×

[+ Add Test Case](#)

Test Suites — grouped test cases with Run All capability

Creating a Suite

- 1 Go to **Test Suites** in the sidebar and click **+ New Suite**
- 2 Give your suite a name (e.g. "Core Engine Tests", "Client Site Checks")
- 3 Click **+ Add Test Case** to add individual tests — each with a name, URL, prompt, and viewport
- 4 Click **Run All** to execute every case in the suite

Tip: From any results page, click **+ Add to Suite** to save that test run directly to an existing suite. This is the fastest way to build up a suite from tests you've already run manually.

Managing Suites

- **Rename** — update the suite name at any time
 - **Delete** — remove the suite and all its cases
 - Add or remove individual test cases within a suite at any time
 - Each suite shows its case count and date of last modification
-

8 Saved Credentials

RoboQA can automatically log into your sites during test runs using securely stored credentials. When a login form is detected on a matching domain, the saved username and password are filled in automatically — no manual input needed.

Settings

Manage your credentials, preferences, and integrations.

ACCOUNT

[Saved Credentials](#)

[Billing & Plan](#)

DEVELOPER

[CI/CD Integration](#)

PREFERENCES

[Notifications](#)

[Integrations](#)

[Network](#)

[Schedules](#)

[Rules](#)

Saved Credentials

Credentials are AES-encrypted and stored securely. When a test detects a login form on a matching domain, the saved username and password are auto-filled — no manual input needed.

DOMAIN	LABEL	USERNAME	BASIC AUTH	AUTH	ACTIONS
the-internet.herokuapp.com	Test Login	tomsmith	—	—	Edit Delete

Add Credential

DOMAIN *

example.com

USERNAME / EMAIL *

user@example.com

LABEL (OPTIONAL)

e.g. Admin account

PASSWORD *

Enter password

☐ This site uses HTTP Basic Auth (browser-level login dialog)

OTP Secret (optional)

e.g. JBSWY3DPEHPK3PXP

TOTP secret key for 2FA-protected accounts. When provided, the engine generates and submits OTP codes automatically after login.

Auth Type (optional)

Standard (none) ▼

Automatically inject this auth header on every request during test runs.

[Save Credential](#)

Settings — Saved Credentials (AES-encrypted at rest)

Adding a credential

- 1 Go to **Settings → Saved Credentials**
- 2 Fill in the **Domain** (e.g. example.com), **Username/Email** , and **Password**
- 3 Optionally add a **Label** to identify the credential (e.g. "Admin account")
- 4 Check **HTTP Basic Auth** if the site uses a browser-level login dialog
- 5 For 2FA-protected sites, **TOTP** to enable automatic OTP
enter your **Secret** generation
- 6 Set **Auth Type** if the site uses a specific auth header (otherwise leave as Standard)
- 7 Click **Save Credential**

All credentials are AES-encrypted at rest. Passwords are never shown in plaintext after saving. RoboQA matches credentials to the domain automatically on every test run.

9

Notifications

Stay informed about test results without checking the dashboard manually. RoboQA can notify you via Slack or email whenever a test run finishes.

Settings

Manage your credentials, preferences, and integrations.

ACCOUNT

🔑 Saved Credentials

💰 Billing & Plan

DEVELOPER

⚙️ CI/CD Integration

PREFERENCES

🔔 Notifications

🔗 Integrations

🌐 Network

📅 Schedules

📋 Rules

Notifications

Get notified via Slack or email when a test run finishes.

SLACK WEBHOOK URL

Create an Incoming Webhook in your Slack workspace and paste the URL here.

NOTIFICATION EMAIL

Notify on

☐ Failures only

☒ All runs (pass and fail)

Save Notification Settings

Settings — Notification configuration (Slack webhook + email)

Setting up Slack notifications

- 1 In your Slack workspace, create an **Incoming Webhook** (Apps → Incoming Webhooks)
- 2 Copy the Webhook URL (starts with `https://hooks.slack.com/services/...`)
- 3 Paste it into **Settings → Notifications → Slack Webhook URL**
- 4 Click **Save Notification Settings**

Email notifications

Enter your email address in the Notification Email field. Choose whether to be notified on **Failures only** (to reduce noise) or **All runs** (pass and fail).

Notifications are sent immediately after each test run completes. If you've enabled scheduled tests, you'll receive an alert after each scheduled run — useful for catching overnight regressions.

CI/CD

Integration

RoboQA integrates directly with your development pipeline. Trigger tests automatically on every push, on a schedule, or from any CI/CD system that can make an HTTP request.

DOCS

CI/CD Integration

Settings

TOOLS

RoboQA Agent

Bulk Test

CI/CD Integration

Run RoboQA journey tests automatically from any pipeline, on every push, or via cron.

Trigger via API

POST to `/api/cicd/trigger` with your session token as the `X-API-Key` header. Your session token is visible in the browser's DevTools → Application → Cookies → `roboqa_session`.

```
# Minimal - runs a journey test and returns results synchronously
curl -X POST https://roboqa.tech/api/cicd/trigger \
-H "X-API-Key: YOUR_SESSION_TOKEN" \
-H "Content-Type: application/json" \
-d '{"url":"https://yourapp.com","prompt":"verify login works"}'

# With viewport and webhook
curl -X POST https://roboqa.tech/api/cicd/trigger \
-H "X-API-Key: YOUR_SESSION_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "url": "https://yourapp.com",
  "prompt": "login then add item to cart and checkout",
  "viewport_preset": "mobile",
  "webhook_url": "https://hooks.yourteam.com/roboqa"
}'
```

Request body fields:

FIELD	TYPE	DESCRIPTION
<code>url</code>	required	The URL to test (must include https://)
<code>prompt</code>	required	Natural language description of the journey to test
<code>viewport_preset</code>	optional	desktop (default), tablet, mobile, mobile_large
<code>webhook_url</code>	optional	URL to POST results to after test completes

CI/CD Integration — API reference and curl examples

Getting your API token

Your session token acts as your API key. Find it in your browser's DevTools → Application → Cookies → `roboqa_session`. Pass it as the `X-API-Key` header.

Trigger a test via API

```
POST https://roboqa.tech/api/cicd/trigger
```

Required fields:

`url` – The URL to test (must include https://)
`prompt` – Natural language description of the journey to test

Optional fields:

`viewport_preset` – desktop (default), tablet, mobile, mobile_large
`webhook_url` – URL to POST results to after test completes

```
curl -X POST https://roboqa.tech/api/cicd/trigger -H "X-API-Key: YOUR_SESSION_TOKEN"
```

GitHub Actions integration

Add a RoboQA step to your `.github/workflows/` directory. Store your session token as a repository secret named `ROBOQA_API_KEY`. The workflow will fail the CI build if RoboQA returns a FAIL status — preventing broken code from being merged.

Webhook payload

When a `webhook_url` is provided, RoboQA POSTs a JSON payload to that URL within 10 seconds of the test finishing. Use the `job_id` to link directly to the results page:

```
https://roboqa.tech/tool/results?run_uuid=<job_id>&run_type=journey
```

Third-Party Integrations

RoboQA connects to external services to extend its automation capabilities and streamline your bug management workflow.

Integrations
Connect third-party services to extend RoboQA's automation capabilities.

2captcha CAPTCHA Solving
When a CAPTCHA is detected during a test run (reCAPTCHA v2, hCaptcha, Cloudflare Turnstile), RoboQA will automatically solve it using 2captcha. Leave blank to disable.

2CAPTCHA API KEY
.....

Save API Key Clear

GitHub Issues
Automatically create GitHub issues for failing tests. Requires a Personal Access Token with `repo` scope.

PERSONAL ACCESS TOKEN
ghp_.....

REPOSITORY (OWNER/REPO)
myorg/my-repo

Save GitHub Settings

Settings — Integrations (2captcha CAPTCHA solving and GitHub Issues auto-ticketing)

2captcha — Automatic CAPTCHA solving

When a CAPTCHA is detected during a test run (reCAPTCHA v2, hCaptcha, Cloudflare Turnstile), RoboQA can automatically solve it using the 2captcha service. Paste your 2captcha API key in **Settings** → **Integrations**. Leave blank to disable.

GitHub Issues — Auto-create bug tickets

Automatically create a GitHub issue for every failing test. Requires a Personal Access Token with the `repo` scope. Configure your token and repository (in `owner/repo` format) in Settings → Integrations.

Jira Cloud — Auto-create bug tickets

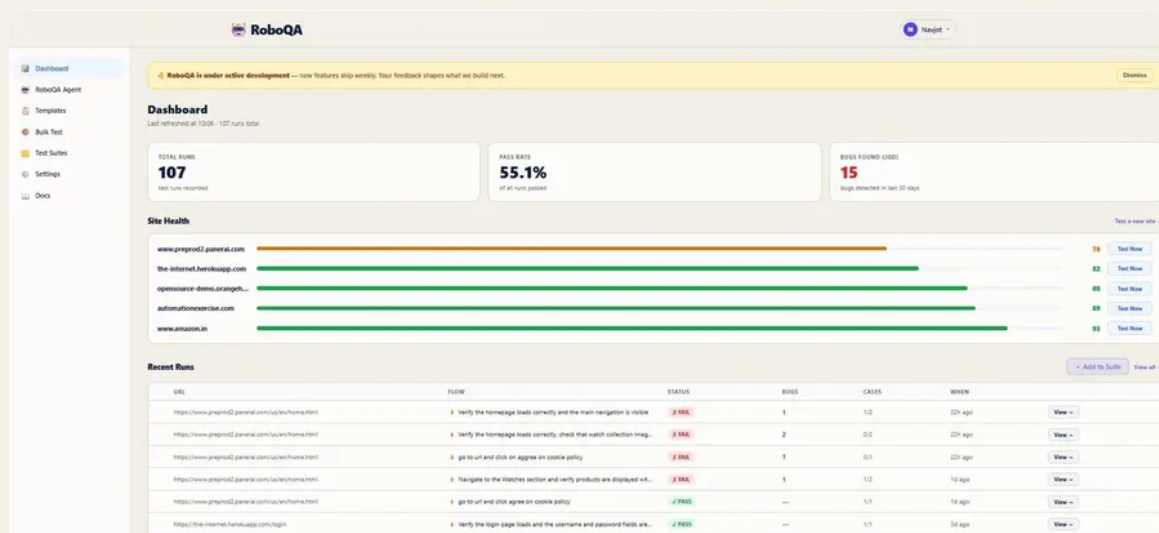
Connect RoboQA to your Jira project to automatically open tickets when tests fail. You'll need your Atlassian workspace domain, Atlassian email, an Atlassian API token (not your password), and your Jira project key.

Auto-Create Tickets

Once GitHub Issues or Jira is configured, enable **Auto-Create Tickets** in Settings → Integrations. RoboQA will automatically open a ticket with bug details, screenshots, and a link to the full run report after each failing test.

12 Dashboard

The Dashboard gives you a bird's-eye view of your entire testing activity — total runs, pass rates, bugs found, site health scores, and a live feed of recent runs.



Dashboard — site health bars, top-line metrics, and Recent Runs table

Top-line metrics

- **Total Runs** — cumulative count of all test runs in your account
- **Pass Rate** — percentage of all runs that returned a PASS status
- **Bugs Found (30d)** — bugs detected across all runs in the past 30 days

Site Health

Each domain you've tested gets a health score displayed as a coloured bar. Green = 80+, amber = 60-79, red = below 60. Click **Test Now** next to any domain to immediately run a new test on that site.

Recent Runs

The Recent Runs table shows your latest test activity — URL, prompt, pass/fail status, bug count, cases, and timestamp. Click **View** → to open the full results page for any run, or **+ Add to Suite** to save it to a Test Suite.

Real-World Use Cases

Development teams

- Run RoboQA in CI/CD on every pull request to catch regressions before merge
- Set up nightly scheduled tests on production and get Slack alerts on failures
- Save credentials for staging and run auth-gated tests automatically

QA engineers

- Build Test Suites covering your full regression checklist and run in one click
- Use Bulk Test to verify a code change across dev, staging, and prod simultaneously
- Enable GitHub or Jira auto-ticketing so bugs are filed without manual effort

Agencies & freelancers

- Bulk-test all client sites after a CMS or plugin update
- Send clients a Public Report link — no account required to view
- Track site health scores over time to demonstrate ongoing maintenance value

Product managers

- Run the Accessibility Audit template before each release
- Use Performance template to verify new features don't slow core pages
- Share results pages with engineering — screenshots and steps already included

14

FAQs & Troubleshooting

How long does a test take?

+

My test failed — how do I know if it's a real bug or a test issue?

+

Can RoboQA test sites that require login?

+

Can I test localhost or internal staging environments?

+

How do I improve my website score?

+

What's the difference between Quick and Deep scan?

+

How does Auto-Create Tickets work with my team's workflow?

+

Can I share results with someone who doesn't have a RoboQA account?

+

RoboQA is under active development — new features ship weekly.

roboqa.tech · [Download this guide \(.docx\)](#)