curiosity

Success Story

Automated API test generation at ThinkDonate

Learn more at curiositysoftware.ie

Contents

curiosity.

Test generation in minutes!

T#INKDONATE

ThinkDonate had less than 7 months to bring its first-of-kind fundraising platform to market. They needed to develop rapidly, while delivering the impeccable user experience required by charities and their donors.

Rigorous testing required an automated and optimised approach to API test creation, capable of generating data and scripts for a range of technologies, scenarios, and integrations.

curiosity

Automated test generation rapidly built a REST Assured framework, maintaining targeted API tests in two-week sprints. In a test-driven approach, test results led the documentation of API validation rules, future proofing development.

The benefits at a glance

- A 25x reduction in test volume relative to an "exhaustive" test suite.
- Optimised test coverage, avoiding risky under-testing and wasteful over-testing.
- The rapid generation of API tests and data. Targeted regression test generation, testing rigorously within two-week sprints.
- The discovery and remediation of API bugs while they remained quick and affordable to fix.
- The creation of clear and comprehensive API specifications, adopting a test-driven approach.
- The avoidance of costly rework in development, maintaining clear "living documentation".
- Accelerated debugging and defect remediation, generating logically-precise bug reports.

A business and campaign critical platform

ThinkDonate were launching a first-of-its-kind fundraising platform. The cutting-edge platform maximises in-the-moment giving on social media. It enables charities and fundraisers to raise money instantly using social media hashtags and payment links, tracking donations in a central dashboard.

Charities can use the online platform to create campaigns and invite corporate partners, influencers, and supporters to collaborate. ThinkDonate's hashtag fundraising tool, #Donate, allows charities to create and share fundraising hashtags on Twitter and LinkedIn, along with payment links on YouTube and Instagram. When a social media user uses the hashtag in the charity's Twitter or LinkedIn feed, it triggers an automated donation flow. This enables supporters to give in the moment using one click payment options like Apple Pay and Google Pay. All payments are processed by Stripe.

"As we prepared to launch our new fundraising platform, delivering quality software on time within budget was business critical. Quality Modeller enabled us to test rigorously across integrated technologies and scenarios, generating API tests far faster than if we tried to script them. It furthermore optimised API testing for coverage, reducing test volume without introducing negative risk. Curiosity's platform enabled rigorous in-sprint testing, while facilitating cutting-edge development practices like shift left API testing, fail-fast experimentation, and test-driven API design. It worked seamlessly alongside our teams and processes, fostering the collaboration, knowledge-sharing, and automation needed for successful software delivery."

Johnny Pitt

Founder of ThinkDonate

The quality of ThinkDonate's first release was imperative, both to ThinkDonate and to the charities who depend on the loyalty and donations of their supporters. At a time when charities are increasingly seeking to monetise social media, they require platforms that facilitate donations reliably and with a seamless user experience. ThinkDonate's success hinged on how well they could offer this impeccable quality in their product launch, with just 7 months to develop the first release.

ThinkDonate required rapid and robust API testing for their web application. In the 7 months from initial development to launch, ThinkDonate developed quickly and iteratively, adopting a test-driven, shift left approach. During two-week iterations, optimised API test generation occurred in parallel to development, with test results informing the design and documentation of API validation rules.

ThinkDonate's parallelised approach to design, testing and development enabled Continuous Delivery, leading to a successful first launch. Rapid development is today ongoing, innovating the ThinkDonate platform based on feedback from early adopters.

Too much to test in-sprint

Testing at ThinkDonate faces a wide array of test scenarios to choose from sprint-over-sprint, created by a host of integrated technologies and logic. Unable to test everything in a two-week iteration, testing must prioritise and formulate the right blend of unit, integrated and system tests. QA must prioritise tests, without introducing negative risk.

Three broad factors create the combined logical complexity of API testing at ThinkDonate:

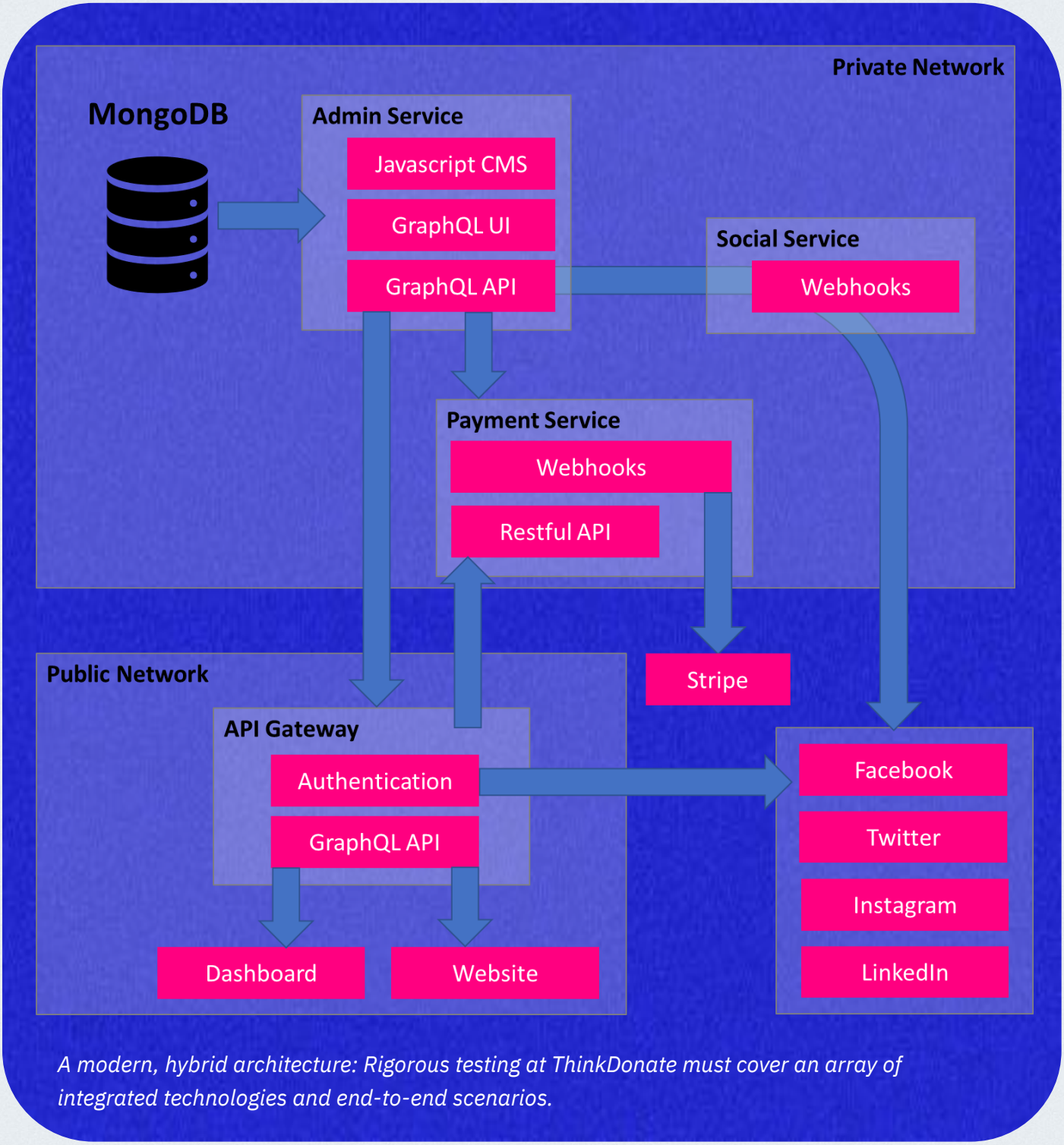
1. The cutting-edge, microservices architecture of the platform, with clusters formed of five containers.
2. ThinkDonate's integrations and extensibility into different social media and payment platforms.
3. The use of different APIs, built using different technologies and hosted on two methods.

The containerised architecture ensures the scalability, security, and performance of the platform, while providing the flexibility needed for rapid development. However, it also creates a vast array of integration points, which must be tested rigorously in combination.

Rigorous testing must also test against different technologies in a hybrid architecture. The ThinkDonate platform is made up of a MongoDB back-end, which is connected to multiple Javascript front-ends and third-party integrations. This interfacing is achieved by Restful and GraphQL APIs, hosted on a public and private network.

Rigorous API testing must test against both REST and GraphQL APIs, while covering a wide array of end-to-end scenarios. These scenarios are created by the internal system logic, along with integrations into different social media platforms and payment platforms.

The API test logic must cover user authentication, as well as the validation of different types of data, entered into dashboards by different types of users. The public API furthermore manages session cookies and ensures that users can only access data relevant to them. Additional functional test complexity stems from testing the rules for aggregating values and generating reports in ThinkDonate's campaigns portal:



A modern, hybrid architecture: Rigorous testing at ThinkDonate must cover an array of integrated technologies and end-to-end scenarios.

The new standard in test data management

Simplify complex application landscapes and provide confidence and clarity at every step of your test data management journey with our intuitive, AI-driven Enterprise Test Data® platform.

Book a Meeting

Manual test creation: Not an option

Faced with more logic than could ever be tested in two-week sprints, ThinkDonate required a targeted and automated approach to API test design. Manually creating the test scenarios and data inputs would simply be too time-consuming and would not hit the range of positive and negative scenarios needed for rigorous testing. ThinkDonate were furthermore keen to “go live” as soon as possible, requiring a rapid and rigorous testing strategy to shorten time to market.

Shift left API testing informs design and development

ThinkDonate’s agile methodology further called for iterative API design, occurring in parallel to development and testing.

When Curiosity joined the project, there was no documentation of the API validation rules, nor API test automation. The development team were focused on building a testable platform that could receive data. They were building and connecting the API endpoints, dashboards, and website, in a “fail fast” methodology that saw as much as 30% of an API rebuilt in a two-week iteration. The development team were performing unit testing and ad hoc manual testing as part of this rapid development, but had no time for integrated and systems testing.

development team were performing unit testing and ad hoc manual testing as part of this rapid development, but had no time for integrated and systems testing.

ThinkDonate required API test automation that could match this pace of development, while supporting the design and documentation of the platform. They opted for a test-driven approach, using API test generation for system discovery and requirements gathering. Flowchart models generated optimised API tests, doubling up as documentation of the API validation rules. Test results were then used to update the central models and regenerate tests, providing requirements and tests for the next sprint.

In this parallelised delivery methodology, API testing does more than simply “check” code after it had been developed. Shift left testing was pivotal in system discovery and design, supporting rapid development and documentation.

Shift left testing and fail-fast experimentation

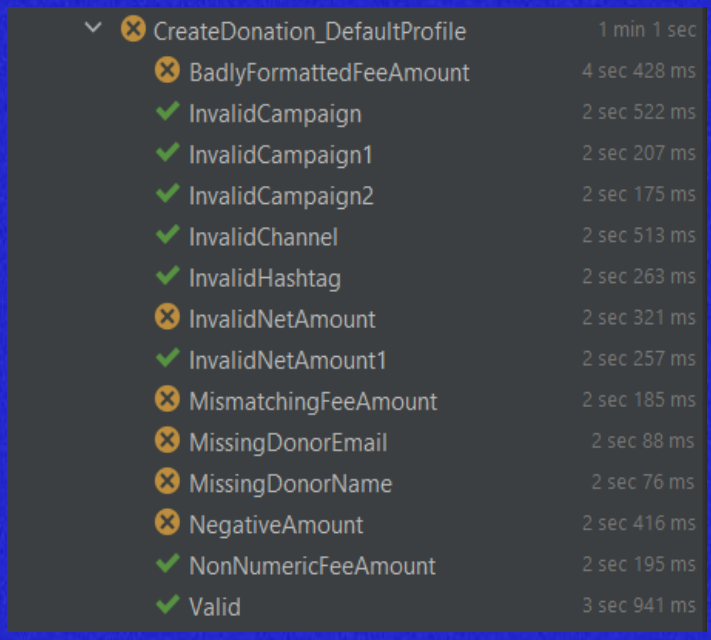
During this shift left approach to API testing, ThinkDonate initially performed low-level verification of the system data against a website and API that could collect data, but had not yet been developed to validate and reject it. This threw up potentially problematic results, which were mitigated against in rapid design and development.

Exploratory and automated API testing uncovered a range of unverified parameters and potential bugs, with as many as 50% of negative tests failing initially.

Some of the system inconsistencies were found during exploratory testing, performed to gather information about the system. For instance, users could edit existing campaigns to set an end date that predates a campaign start date.

Automated API testing found further validation errors. For example, it was possible to “pay” a negative or badly formatted amount via the API, and to create campaigns with empty names and invalid dates.

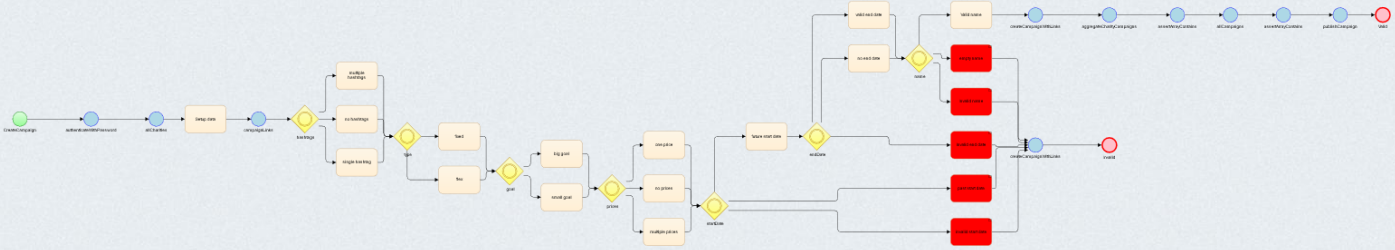
Auto-generated API tests additionally uncovered performance issues, particularly with aggregate functions. When making 100 donations, for instance, wait times exceeded 4 seconds per donation, potentially undermining the likelihood of converting traffic into donations.



Automated API tests identify unexpected results for negative data entered when making a donation.

Rapid and optimised API test generation

The API testing that uncovered these issues was powered by optimised API test generation. Curiosity and ThinkDonate worked together to generate optimised tests from flowcharts that double-up as documentation for the fundraising platform:



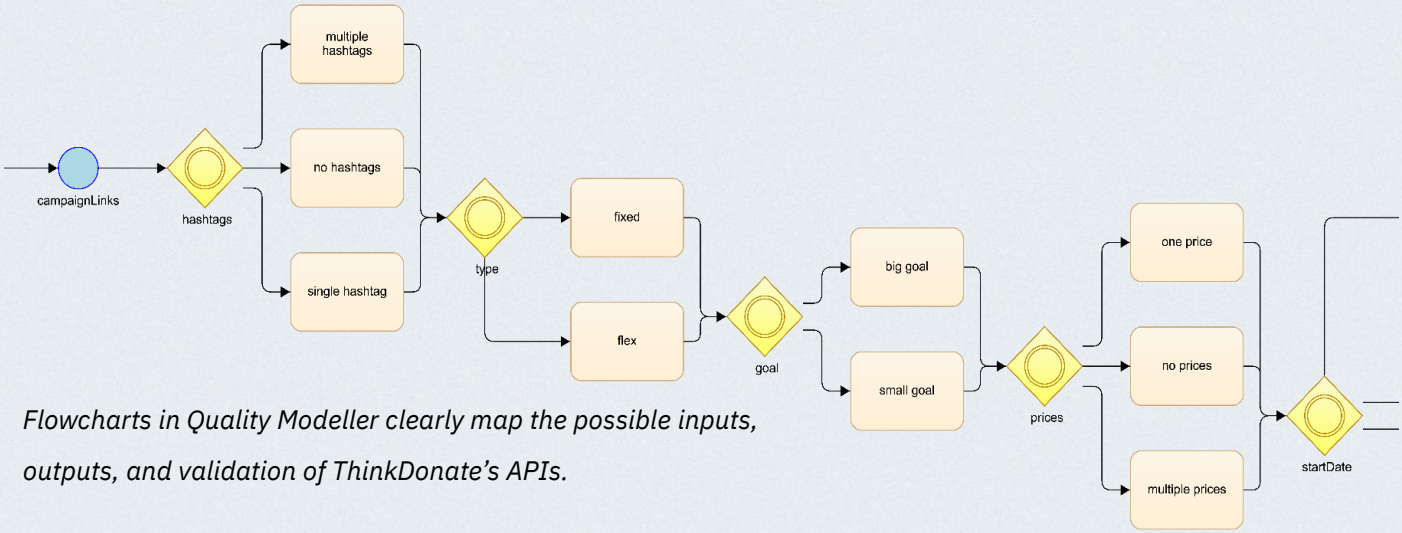
Quality Modeller's intuitive flowcharts generate optimised API tests and provide specifications for the validation logic of ThinkDonate's APIs.

These intuitive flows auto-generate code that sends API requests to the Restful and GraphQL APIs, rapidly and rigorously testing ThinkDonate's interfaces. New validation rules were then documented in Quality Modeller's flowcharts. This communicated changing requirements clearly to development, while refining the regression test model for future iterations.

Iterative system modelling

With no existing documentation, flowchart modelling initially worked backwards from the system to create a visual map of the integrated APIs. This built logically precise pictures of how data could flow through the integrated APIs, producing clear documentation of the complex system.

Modelling was rapid and iterative, taking the system piece-by-piece. A modeller would build a flowchart, generate tests, and create a run report. The report would then be shared with the product ownership at ThinkDonate, confirming and refining the validation rules.

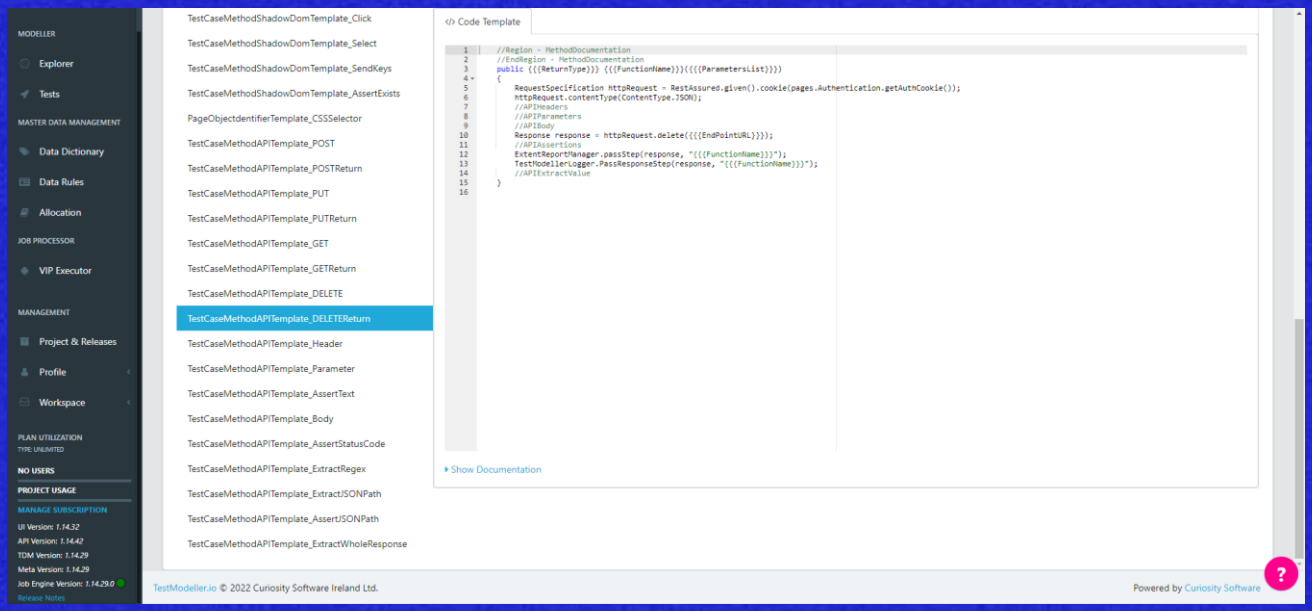


Flowcharts in Quality Modeller clearly map the possible inputs, outputs, and validation of ThinkDonate's APIs.

Errors were then fixed in development, while newly uncovered scenarios were added to the easy-to-maintain flowcharts. This maintained clear and complete documentation of the APIs, while also building a robust regression pack for future development.

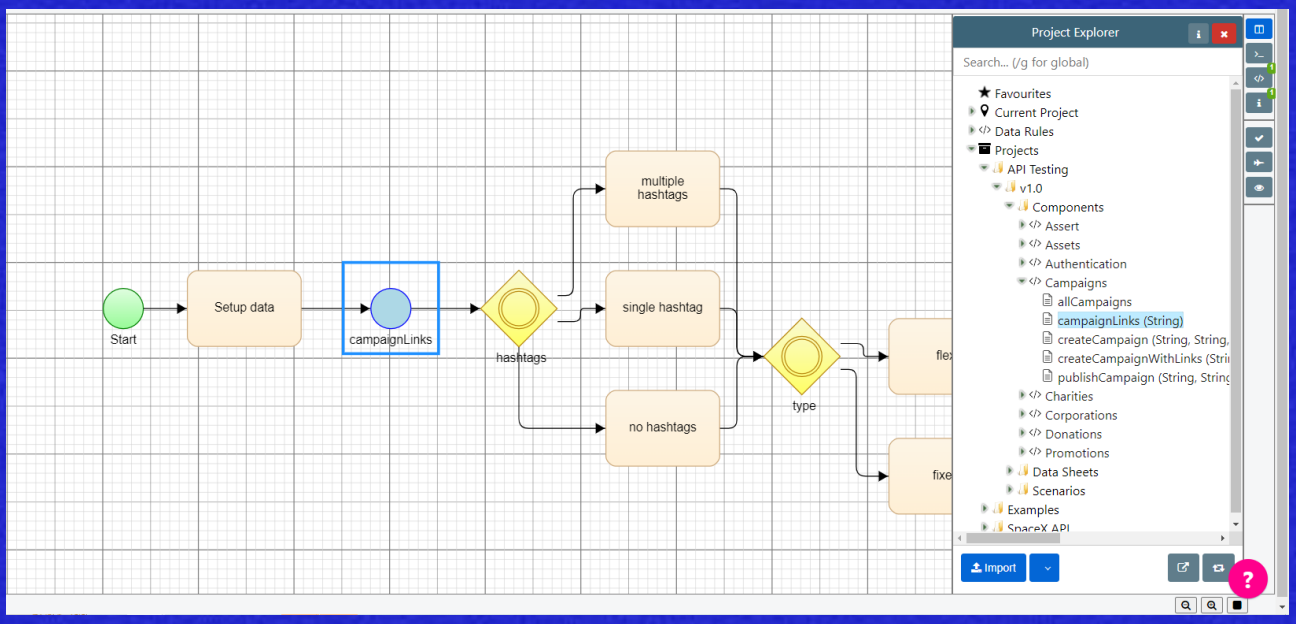
API test script generation

The API test creation was rapid and rigorous using Quality Modeller, autogenerating tests for a Java REST Assured framework. Quality Modeller's quickstart framework provided a foundational library of out-of-the-box automation. This was augmented with custom code, defining code snippets in Quality Modeller's template editor:



Quality Modeller's quickstart API framework provided an easily configurable and extendable Java framework.

This approach rapidly built a comprehensive library of reusable automation actions, while retaining the full flexibility of scripting. An automation engineer using Quality Modeller only needs to configure a code snippet once. Each configured action can then be overlaid onto visual flowcharts, linking them together and adding variables to generate optimised test suites:

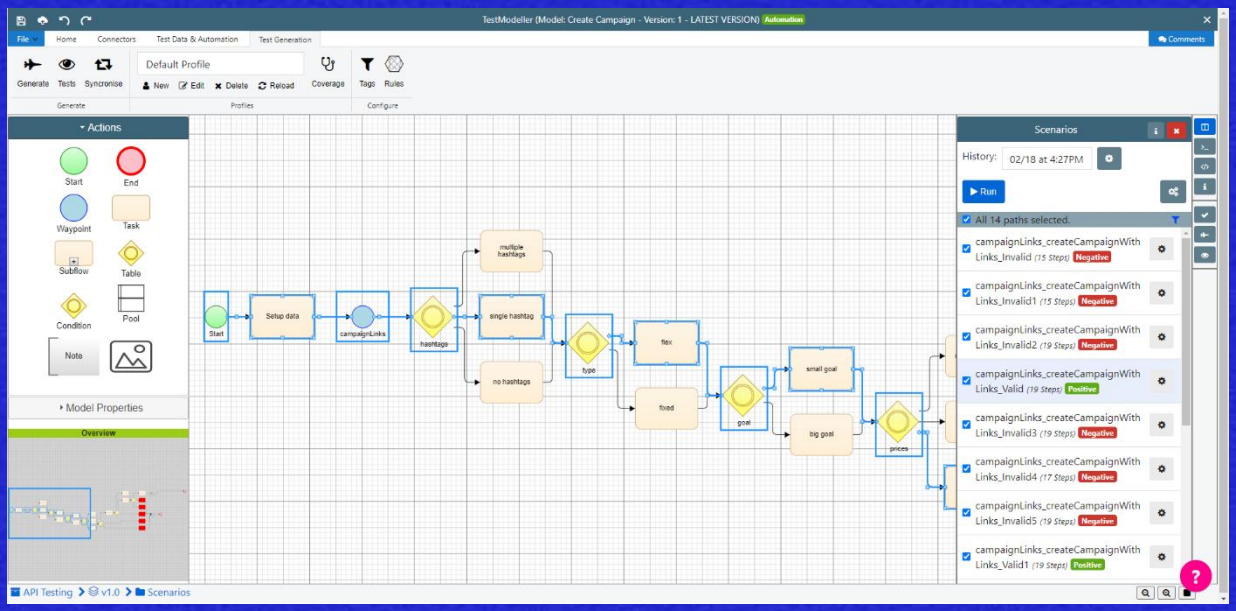


Quality Modeller users can overlay reusable automation in blue "waypoints" to auto-generate test scripts.

Hitting "generate" then compiles complete test suites at speeds unthinkable with manual test scripting.

API test optimisation

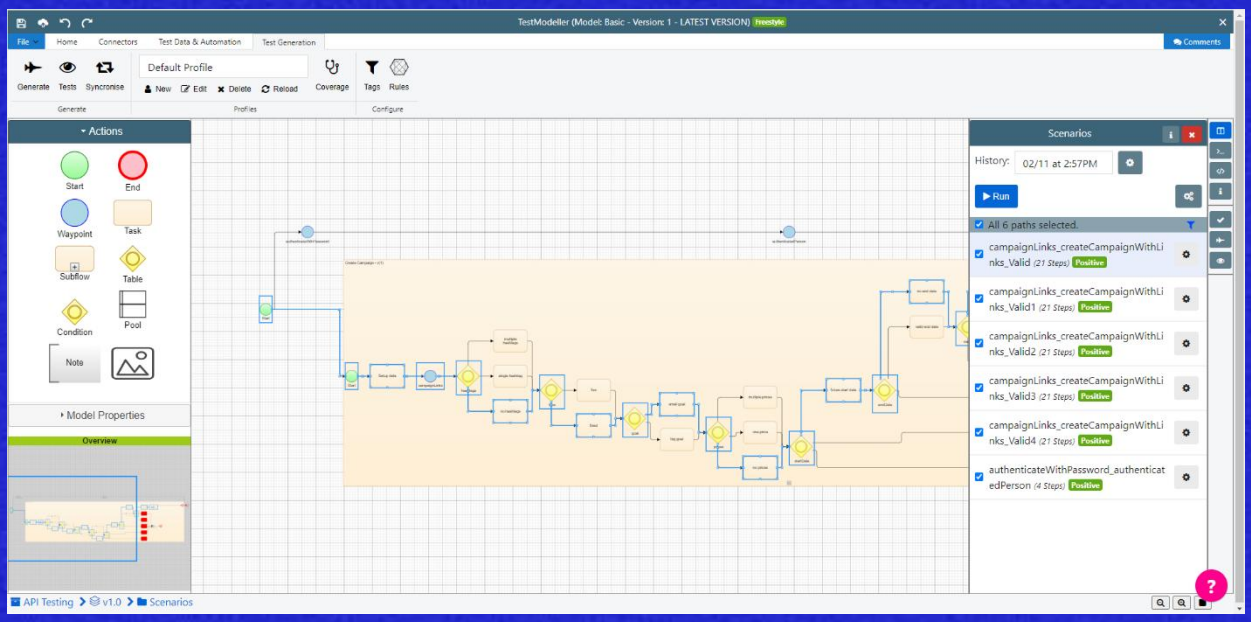
The API tests are furthermore optimised for coverage. Automated algorithms generate the smallest set of tests needed to satisfy different risk profiles. This reduces test volume without creating negative risk, generating targeted tests from among the vast scenarios contained in the ThinkDonate platform:



Quality Modeller auto-generates the smallest of positive and negative tests needed to satisfy a given coverage profile, complete with test data and expected results.

Each path through a Quality Modeller flowchart is equivalent to a possible API test scenario. Applying different coverage profiles identifies the optimal set of paths needed to “cover” the model. For example, 14 test cases were generated for the process of creating and editing campaigns, out of a possible 324 tests. The optimal tests “covered” the distinct logical combinations contained in the flowchart, rigorously testing with just 4% of the possible test volume.

Test optimisation enables ThinkDonate to test rigorously across integrated systems, generating compact test suites for end-to-end journeys. When generating API tests for the donation process, for example, Quality Modeller generated 18 distinct scenarios to “cover” the integrations with four different social media channels.

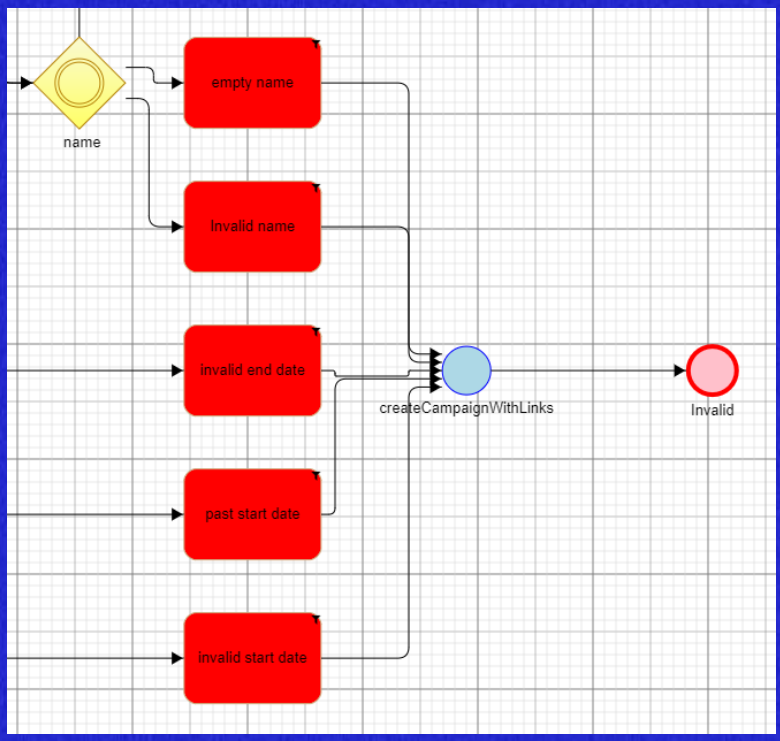


Every model created in Quality Modeller becomes a reusable component, enabling rapid modelling and optimised test generation for end-to-end scenarios.

Risk-based API testing

Coverage profiles can further be used to target particular logic, focussing API testing based on time and risk. ThinkDonate, for instance, generate test suites that focus primarily on negative tests, colour-coding the test models to clearly mark the negative API inputs.

This risk-based approach at ThinkDonate "rightsizes" testing during Continuous Delivery. As ThinkDonate's APIs develop rapidly during two-week release cycles, they can test only as much logic as needed to de-risk new releases. Rigorous testing is not then a blocker to speed or agility, targeting testing where it is likely to have the greatest impact in finding bugs.



Quality Modeller's test models can be colour-coded to clearly mark negative tests.

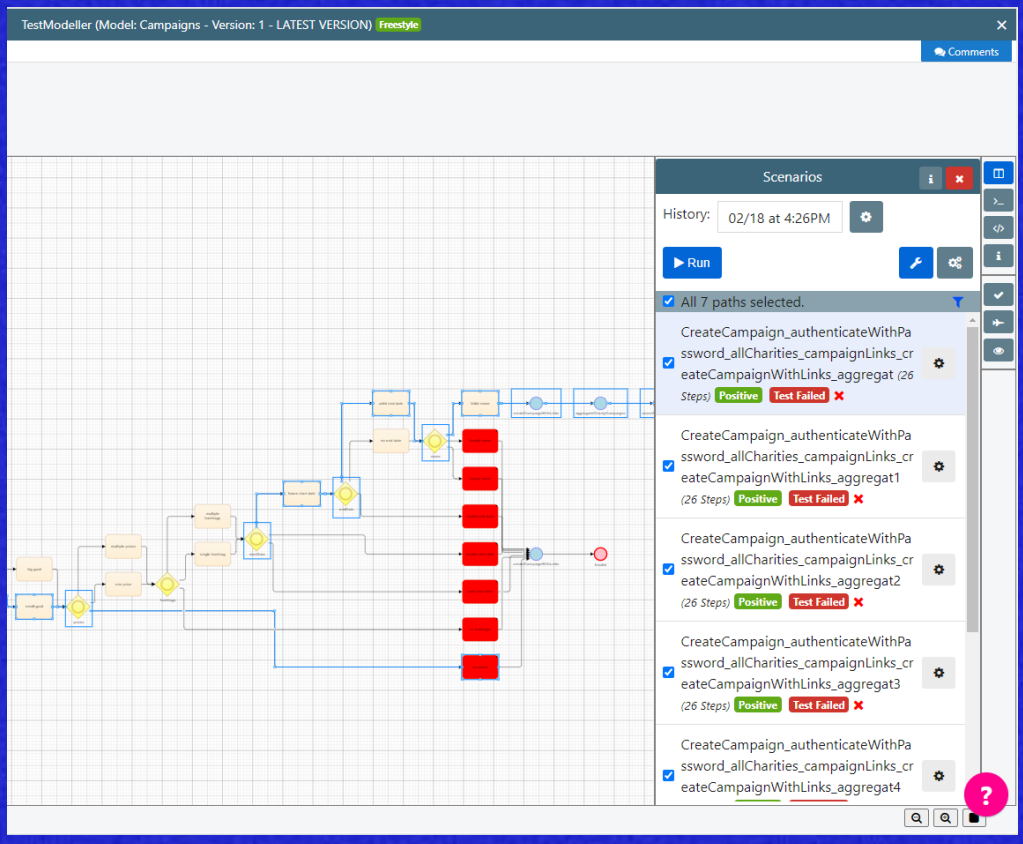
Clear and collaborative reporting

In ThinkDonate's test-driven approach, clear test reporting informed both API design and development. At the end of each two-week sprint, reports were shared for review by business stakeholders. This supported collaboration between development and product owners, while equipping both with knowledge of the APIs' validation logic. Meanwhile, developers were provided with granular understanding of failing API logic, enabling rapid defect remediation.

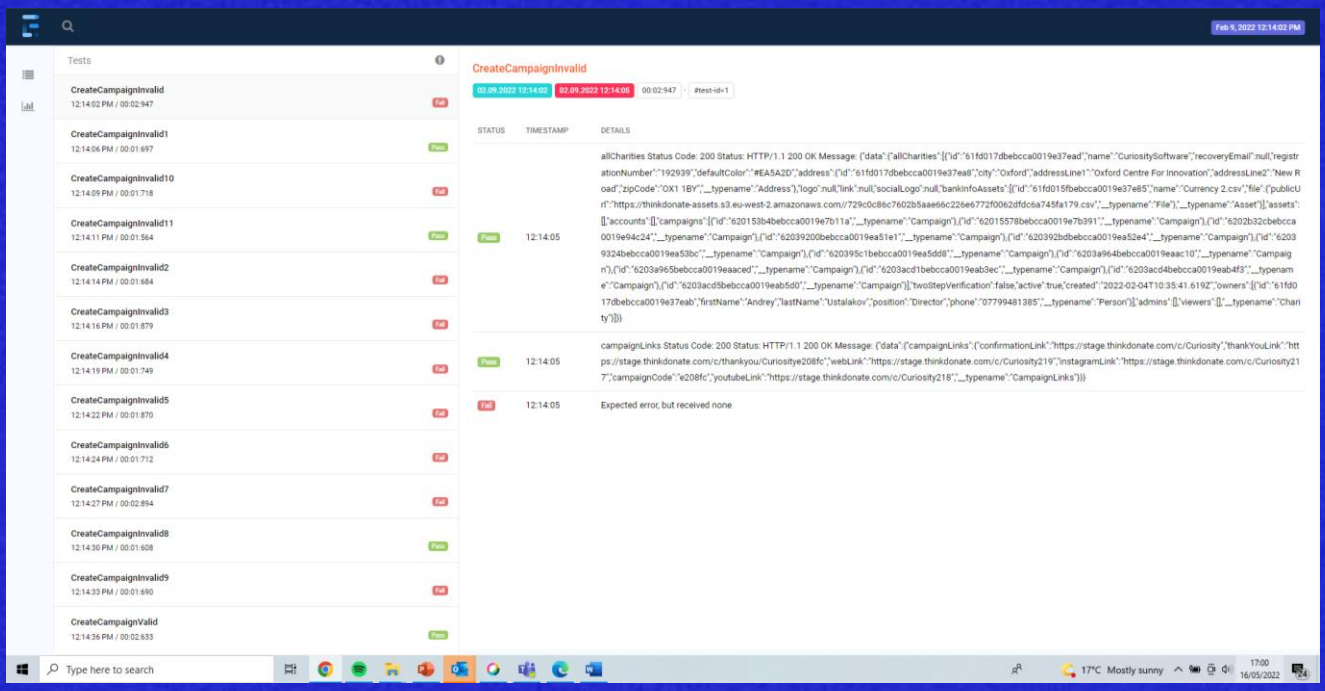
API tests generated by Quality Modeller updated pass/fail results in Quality Modeller's central flowcharts, while generating screenshots and HTML reports.

Overlaying run results in Quality Modeller's flowcharts provided granular bug reports for precise and rapid debugging. Run results were updated at the path level, enabling easier route cause analysis of defects. Developers could visually pinpoint a failing step in multiple API requests, contextualising the bug within the broader API logic:

Pass/fail run results overlaid visually onto flowcharts support rapid route cause analysis and accelerate defect remediation.



The Java REST Assured test framework additionally captured screenshots during test execution. These were shared with business and developer stakeholders and were attached to Quality Modeller's flowcharts. HTML reporting further supported debugging:



HTML reports, screenshots, and run results at the model level supported a test-driven approach to API design and development.

The test reporting provided a set of possible validation rules, which were converted by ThinkDonate's product owners into documentation for the API validation. This collaborative, test-driven approach provided documentation for every field, informing future development and regression testing. Iteratively maintaining clear documentation further future-proofs development, as it maintains in-house knowledge of the changing system.

In-sprint test maintenance

Generating API tests directly from "living documentation" in Quality Modeller further maintains up-to-date regression suites, avoiding technical debt and rigorously testing the fast-changing APIs.

API test maintenance in Quality Modeller is quick and automated, supporting rapid development. Updating central flows regenerates API test suites in the click of a button, avoiding time lost checking and updating repetitive test suites.

At ThinkDonate, the componentisation of Quality Modeller's model-based approach enabled rapid maintenance of documentation and tests. New validation rules formulated by product owners were quickly incorporated into Quality Modeller's flowcharts. Updating the API logic in one central subprocess then updated the validation rules across interrelated scripts and flowcharts.

A successful launch – and many more to come

After just 7 months from starting development, ThinkDonate launched its platform to seven charities. The successful launch enabled one charity to raise urgently needed funds, and ThinkDonate are now working with several charities in the UK top 100.

The ThinkDonate platform continues its roadmap of rapid development, incorporating feedback gathered following the initial launch. Continuous development remains imperative, driven by the robust design and rigorous testing. At ThinkDonate, Quality Modeller has provided a range of benefits across API design, development and testing:

A 25x reduction in test volume relative to an "exhaustive" test suite.

Optimised test coverage, avoiding risky under-testing and wasteful over-testing.

The rapid generation of API tests and data. Targeted regression test generation, testing rigorously within two-week sprints.

The discovery and remediation of API bugs while they remained quick and affordable to fix.

The creation of clear and comprehensive API specifications, adopting a test-driven approach.

The avoidance of costly rework in development, maintaining clear "living documentation".

Accelerated debugging and defect remediation, generating logically-precise bug reports.

Get started today!

Visit www.curiositysoftware.ie to learn more or book a demo with a Curiosity expert today!

Additionally, you can also email us at info@curiosity.software

Call USA:

+1 914 218 0180

Curiosity Software Ireland

Unit 6 The Mill, The Maltings, Bray, Co. Wicklow, A98 XV40, Ireland

Curiosity Software USA

4136 Del Ray Ave. Suite 658, Marina Del Rey, CA 90292, USA