



SOLUTIONS BRIEF

HyperTest monitors your entire application traffic, builds integration tests and exposes API failures and incidents.



HyperTest uses your API traffic to not only auto-generate integration tests that you can use to test every new commit, but also discovers all the API artifacts, discovering details and documenting them for you, creating an up-to-date API inventory. Here are a few use cases:

Discover All Your APIs

Maintain an up-to-date inventory of all your APIs known and unknown

Prevent Fatal and Non-fatal Errors

Catch API regressions that can cause crashes, data errors or schema failures

Accelerate Root Cause Analysis

Reproduce errors using real user journeys and resolve before they hit production

Test and Secure Every Commit

Use CI to auto-trigger tests to check regression and security flaws for every PR

Prevent Account Takeover

Catch broken user authentication, authorisation to prevent account takeover

Quantify Testing With Coverage

Discover parts of your application covered and not covered with automation

APIs Performance Monitoring

Baseline API responses and track deviation

Prevent Sensitive Data Exposure

Track APIs that expose PII, passwords or other critical data

KEY CAPABILITIES OF HYPERTEST



API observability - testing, security and performance

HyperTest discovers and explores the entire API footprint of the application by monitoring it 24x7. The test cases it automatically generates from this traffic will report API failures around regressions, security and performance helps development, operations and testing teams a run-time observation of their applications

No-code like It Was Meant to Be

HyperTest eliminates the need to write or maintain any automation test hence solving the problem of API regression in a true no-code way. Leaves teams to completely focus on development.

True 'Shift-left', Trigger From CI

HyperTest shift-left approach helps teams checks for API issues right at the time of development. Teams can integrate HyperTest with their CI pipeline and trigger tests at every PR. These test reports are then sent to respective developers helping them identify and fix the problem at the source itself.