



GUIDE TO

# **Build E2E Integration Tests Without Managing Test Environments or Test Data**

[www.hypertest.co](http://www.hypertest.co)

As software development cycles become increasingly agile and the demand for faster release schedules grows, the ability to efficiently test applications end-to-end (E2E) in an integrated manner becomes crucial. QA professionals and SDETs often face significant challenges in managing test environments and preparing test data.

## CHALLENGES WITH PREPARING TEST DATA

Creating relevant and compliant test data is a significant hurdle. Test data must not only reflect realistic scenarios for accurate testing but should also be consistent in nature. Generating such data in sufficient volumes to cover all test cases is a time-consuming task that is often repetitive.

- Test data must accurately reflect production data to ensure that tests are realistic and meaningful. However, creating and maintaining such data sets can be difficult due to changing production data schemas and the need for data to cover all possible test cases.

“  
*Discrepancies between test and production data led to over 30% of defects going undetected until after deployment*  
”

- Mocking complex dependencies (e.g., APIs, databases) accurately can be difficult to configure and keep in sync with actual services. Mocked data can diverge from production behavior, resulting in tests that pass in isolation but fail in real-world scenarios.

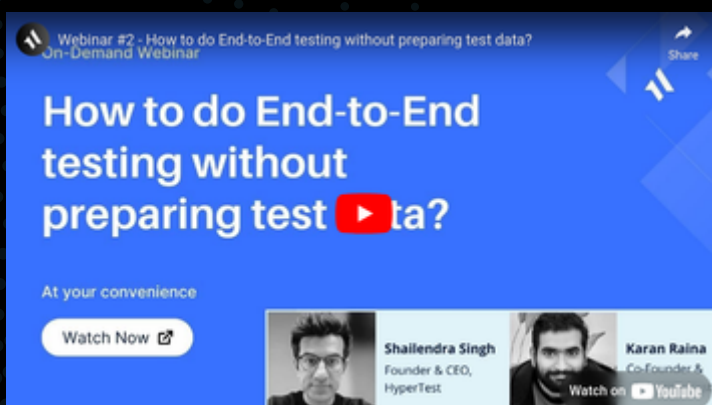
**Check out our guide on No More Writing Mocks: The Future of Integration Testing**

- Assertions might break if the data isn't in the expected state. For data to remain consistent, it requires regular resets or complex handling, introducing overhead and delay test execution.

## MANAGING TEST ENVIRONMENTS IS A TASK

Maintaining multiple test environments is both complex and resource-intensive. Each environment needs to replicate the production setup to ensure tests are realistic. This includes mirroring database schemas, third-party services, and various server configurations.

- The discrepancy between environments often leads to the "works on my machine" syndrome, where code behaves differently in production than in test environments, leading to critical bugs in live applications.
- When one test modifies the environment or data, it can interfere with other tests, leading to assertion failures due to unintended side effects. Managing isolation within shared environments, or creating isolated test environments, can be both technically challenging and resource-intensive.
- E2E integration tests often rely on multiple dependencies, such as databases, APIs, and third-party services, which need to be available and behave predictably during testing. Managing these dependencies within the test environment is challenging because they may not always be under direct control.



Webinar #2 - How to do End-to-End testing without preparing test data?  
On-Demand Webinar

How to do End-to-End testing without preparing test data?

At your convenience

Watch Now

Shalendra Singh  
Founder & CEO,  
HyperTest

Karan Raina  
Co-Founder &  
Watch on YouTube

Watch a webinar on

## Perform E2E Testing Without Preparing Test Data

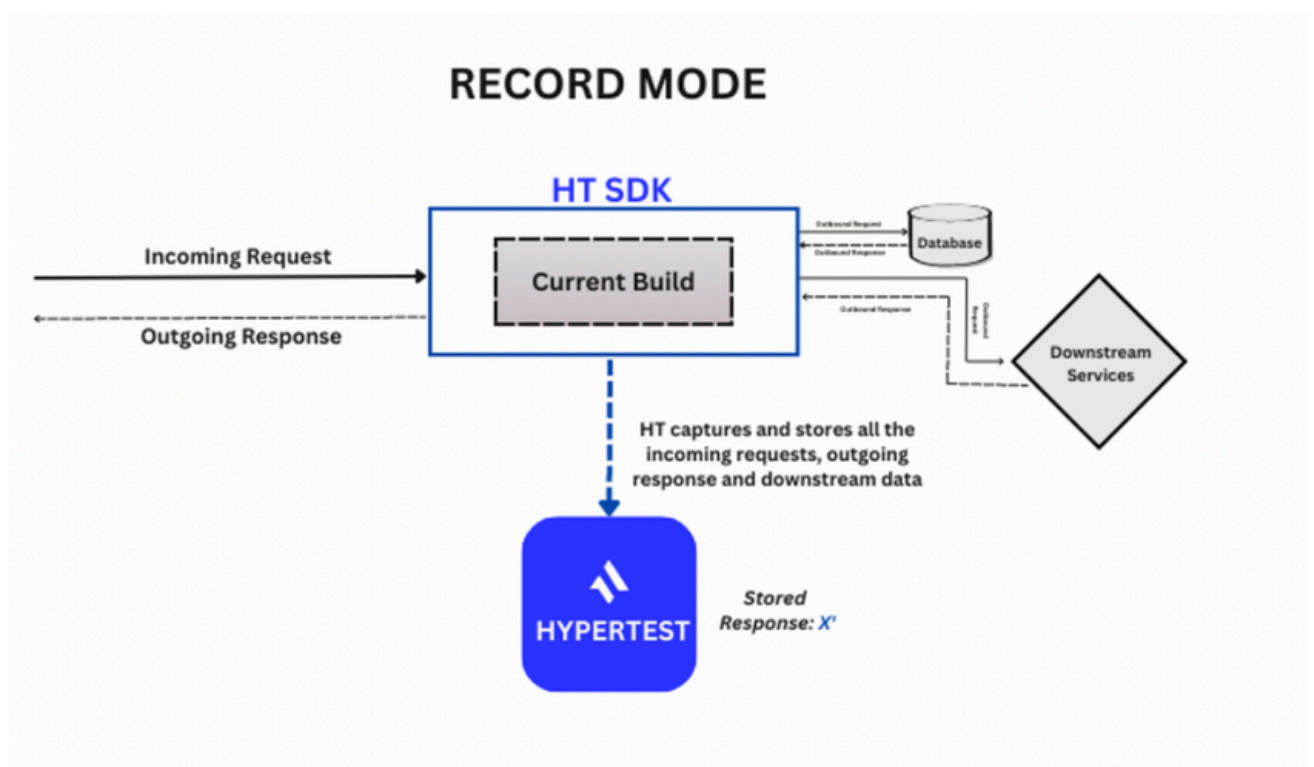
# How HyperTest is going to change the way you test?

HyperTest captures real interactions between code and external components using actual application traffic, then converts it into tests which can be replayed at your pre-commit stage.

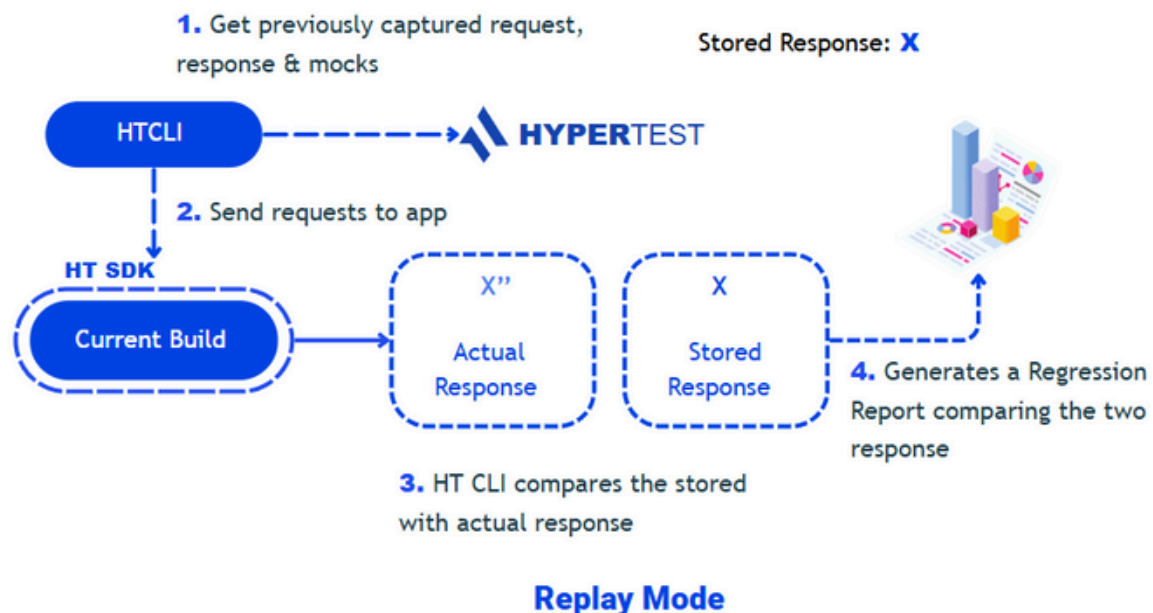
Built on top of open telemetry, it gets initialized in your service just like an application performance monitoring tool (APM) would and gathers telemetry data such as logs, traces and metrics from this.

“  
*HyperTest is a game changer, it has significantly saved time and effort by green-lighting changes before they go live with our weekly releases.*  
”

It is built on the principal of something called RECORD mode and REPLAY mode.



In the **RECORD** mode, it will record all the incoming requests to the application, along with its payload, out bound calls made to any 3rd party application like a database, messaging queue or a caching application and their responses. It will do this 24/7 to establish the baseline response of an application.



As soon as the developer makes a change in the code, HyperTest runs these requests again in **REPLAY** mode and this time, the new response or the actual response will be compared to the previously generated baseline response to catch any regressions caused by the change in the code.

**THIS LEADS TO CERTAIN ADVANTAGES:-**

### ➡ Never worry about creating or managing test data

HyperTest can test stateful flows without needing teams to create or manage test data. Just from service's traffic, it can easily create test data that can be utilized to build your tests automatically.

It keeps on updating these tests from the most recent iteration of data that it gets from this traffic.



## ➔ No need to manage test environments

These tests can be recorded from any environment and can be run on a local machine, thereby, eliminating the need for separate environment for just testing.

## ➔ No need to provide any assertions

Since HyperTest records the whole traffic of the application, it automatically records the whole workflow of your application. It essentially registers the logic and the mannerism of the application, which thereby eliminates any need for asserting the business logic for the test.

## ➔ All external dependencies are mocked automatically

HyperTest SDK, that sits on the service, can mock it's interaction with any 3rd party service, e.g. a database layer or a messaging queue, automatically without needing to use any particular framework like JEST, MOCHA etc.

This becomes advantageous as you don't need to perpetually run any of these 3rd party applications in sync with your service when testing your service.

Why we built HyperTest? Dev Talks 01

Share

HYPERTEST'S  
**DEV TALKS**

**Click here to know  
why we built  
HyperTest**

Watch on  YouTube







# Summary

Building E2E style Integration Tests Without Managing Test Environments or Test Data

- Managing multiple tests environments and preparing test data can be time-consuming and repetitive. With HyperTest's ability to smartly mock-out the real data, you can skip the test data preparation entirely and you're also free to run your tests seamlessly from any environment, freeing you to focus on building and scaling your tests without extra setup overhead.

2022

HyperTest was started

2023

Node SDK Launch

2023

100 services go live

2024

Launched our Java SDK



## Generating Greatness

Companies like **Porter**, **Paysense**, **Nykaa**, **Airmeet**, **Skud** and **Fyers** leverage HyperTest to accelerate time to market, reduce delays and improve code quality without needing to write or maintain automation

Explore [↗](#)

 **HYPERTEST**