

# ADM Market Insight:

How Testing Keeps the API Economy Running



# Contents

How Testing Keeps the API Economy Running	3
Testing APIs: How and Why?	5
Testing APIs: How and Why?	6
How OpenText Products Enhance Testing	7
OpenText Functional Testing	8
OpenText Functional Testing	9
OpenText Service Virtualization	10
OpenText Functional Testing for Developers	11
Testing Your Own APIs	12



# How Testing Keeps the API Economy Running

The API economy underpins digital transformation, enabling businesses to access, share, and integrate their processes and business models. Testing creates a safe API economy where data and resources are shared freely without fear of errors. API testing checks the functionality, reliability, accuracy, performance, and security of the programming interfaces.

APIs enable companies to grow their businesses more quickly than ever before. Just like the web opened the internet's potential, APIs are driving a new wave of innovation centered on sharing services. We call it the API economy.

**Application programming interfaces (APIs) underpin successful and scalable applications. Well-defined and properly tested APIs provide an interface for applications to communicate among themselves, and they encourage service and resource reuse. APIs also enable applications to talk to each other and share resources, allowing one app to use another’s capabilities.**

Architecting solutions around APIs offer numerous advantages. Structurally, it accelerates application scaling since application modules can be split into microservices. It also discourages repetition as applications communicate and share data amongst themselves.

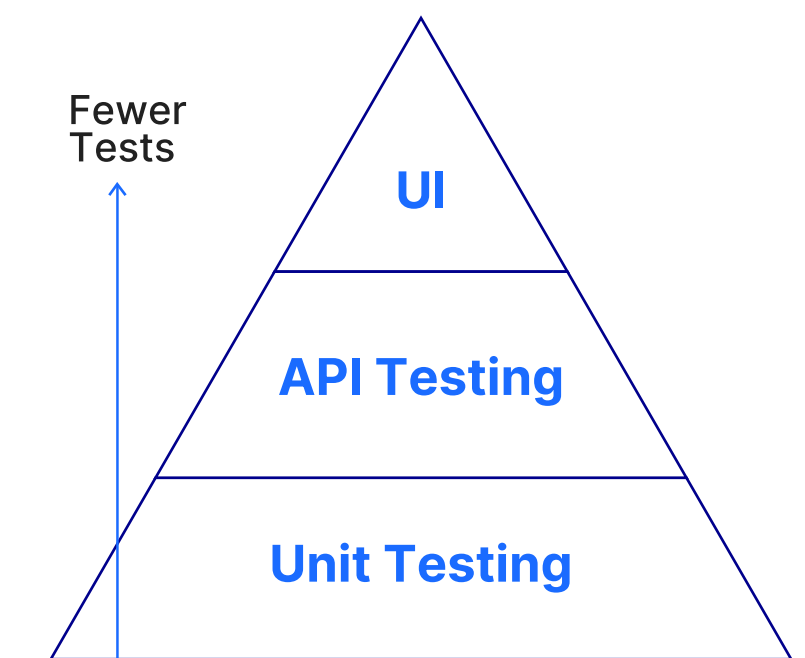
The API economy underpins digital transformation, enabling businesses to share, access, and integrate their processes and business models.

However, effectively participating in the API economy requires testing—whether you’re an API producer or consumer. Let’s examine how API testing works, why it’s so important, and how to ensure your own APIs are functioning well.

## **APIs Are the Glue that Binds an Application Ecosystem**

**Figure 1.1: A testing pyramid**

The testing pyramid pictured above depicts different layers of application testing, with API tests in the middle.



**You can test at any stage of the application lifecycle. However, the importance of testing early in the application lifecycle, where bugs are more easily discovered, and cheaper and easier to fix, can't be overemphasized.**

# Testing APIs: How and Why?

The goal of API testing is to check the functionality, reliability, accuracy, performance, and security of the programming interfaces. Testing creates a safe API economy where data and resources are shared freely without fear of errors. You may ask, what happens when I use—or provide—untested APIs? First, untested APIs cannot be trusted. You have no way to ascertain their reliability and accuracy. In the case of sensitive industries, such as healthcare and finance, using an untested API can lead to disastrous outcomes. API testing instills confidence in both the supplier and the consumer by reducing doubts around reliability.

Two key questions that often spring up whenever software testing is discussed are when and how do we test an API?

You can test at any stage of the application lifecycle. However, we cannot overemphasize the importance of testing early, where bugs are more easily discovered, and cheaper and easier to fix. Unit testing at an early stage ensures that all the individual units are in tune with user requirements.

**Unit testing at an early stage ensures that all the individual units are in tune with user requirements.**



# Testing APIs: How and Why?

Service testing ensures that all the application's components not only work perfectly as individual units, but also work as expected when combined to form a service. These tests can be automated with tools like [OpenText™ Functional Testing](#), open source, and other commercial tools to reduce test cycle time. Headless layer technologies like Functional Testing can test services which are not exposed through the GUI.

Services should be tested independently of their consuming applications to ensure that the expected responses are returned, ensuring accuracy and quality. Tools like [OpenText™ Service Virtualization](#) can also independently test services.

# How OpenText Products Enhance Testing

OpenText is one of the world's largest enterprise software providers, trusted for our proven mission-critical software that keeps the digital world running. Our pragmatic, disciplined, customer-centric approach enables customers to succeed in today's rapidly evolving marketplace. Some of our products are tailored to make API testing easier, improving the API economy's safety.

OpenText functional testing products include:

- [OpenText™ Functional Testing](#)
- [OpenText™ Service Virtualization](#)
- [OpenText™ Functional Testing for Developers](#)
- [OpenText™ Functional Testing Lab for Mobile and Web](#)



# OpenText Functional Testing

OpenText Functional Testing accelerates and simplifies end-to-end functional testing by intelligently automating tests for enterprise applications using embedded AI-based capabilities.

OpenText Functional Testing's unified functional testing seamlessly tests complex interactions across all three main types of application layers. It includes the ability to test the graphical user interface (GUI) layer, the API layer, and end-to-end business processes.

**GUI is the part of the application visible to the user. Testing it ensures the software application functionalities work in line with the application specifications by checking the application screens and controls—such as menus, buttons, icons, etc. You can even run browser tests in headless mode, so these tests can easily be run from CI/CD servers that don't have a GUI environment installed. Headless tests run quickly as they avoid draw operations, eliminating UI rendering.**

**API layer testing involves testing backend services directly, instead of indirectly via UI interactions. Direct API testing is useful because not every API call is triggered by user interaction. Some APIs are even accessed exclusively by other APIs or backend services, so testing them directly is the only way to ensure they're working properly. Since not all aspects of an application can be tested through the GUI, UFT One's focus on the API layer reduces risk.**

With OpenText Functional Testing, you can test APIs using built-in standard activities, such as file and string manipulation, data conversion, and messaging. For time-bound projects, import your existing resources (SoapUI, WSDL, WADL, Swagger, OData) and let OpenText Functional Testing automatically generate API tests for you. Define which critical aspects to test: positive, boundary, security, and/or compliance.

OpenText Functional Testing also supports enterprise businesses using cutting-edge IoT technologies with support for Message Queuing Telemetry Transport (MQTT) and constrained application protocol (CoAP). OpenText Functional Testing has extensive support for creating and importing REST service models, as well as sending and receiving JSON requests for REST API services.



# OpenText Functional Testing

**Testing an application's end-to-end business process entails testing the entire software from beginning to end, ensuring the application flow behaves as expected. It defines the product's system dependencies and ensures all integrated pieces work together as expected. The main objective of end-to-end testing is to test from the end user's perspective by simulating a real scenario and validating the system and its components for accuracy.**

OpenText Functional Testing has extensive API testing capabilities and unique benefits. It improves application quality by testing more functionality in composition apps across multiple layers. By testing the functionality of an integrated scenario, OpenText Functional Testing reduces errors by allowing you to find defects you would otherwise miss.

OpenText Functional Testing is cost-efficient as it allows you to adopt a single solution for GUI, headless, and multi-layer testing. It can even be extended to additional technologies not supported out of the box.

It enables running multiple tests simultaneously which amounts to a faster ROI. OpenText Functional Testing allows you to execute up to four tests in parallel, run API tests that call GUI tests in parallel, and call API tests in GUI test parallel runs.

OpenText Functional Testing unified functional testing seamlessly tests complex interactions across all three main types of application layers. It includes the ability to test the graphical user interface (GUI) layer, the API layer, and end-to-end business processes.



# OpenText Service Virtualization

Service virtualization simulates API services, testing applications that depend on APIs even when the APIs are not yet available. It enables teams to easily create realistic service behavior simulations. This facilitates testing applications in isolation and allows you to determine how the application responds to simulated error conditions from the API.

Service virtualization software removes the development and testing “wait time” that can slow the delivery of applications. You can stay on schedule even when dependent application components or systems are unavailable.

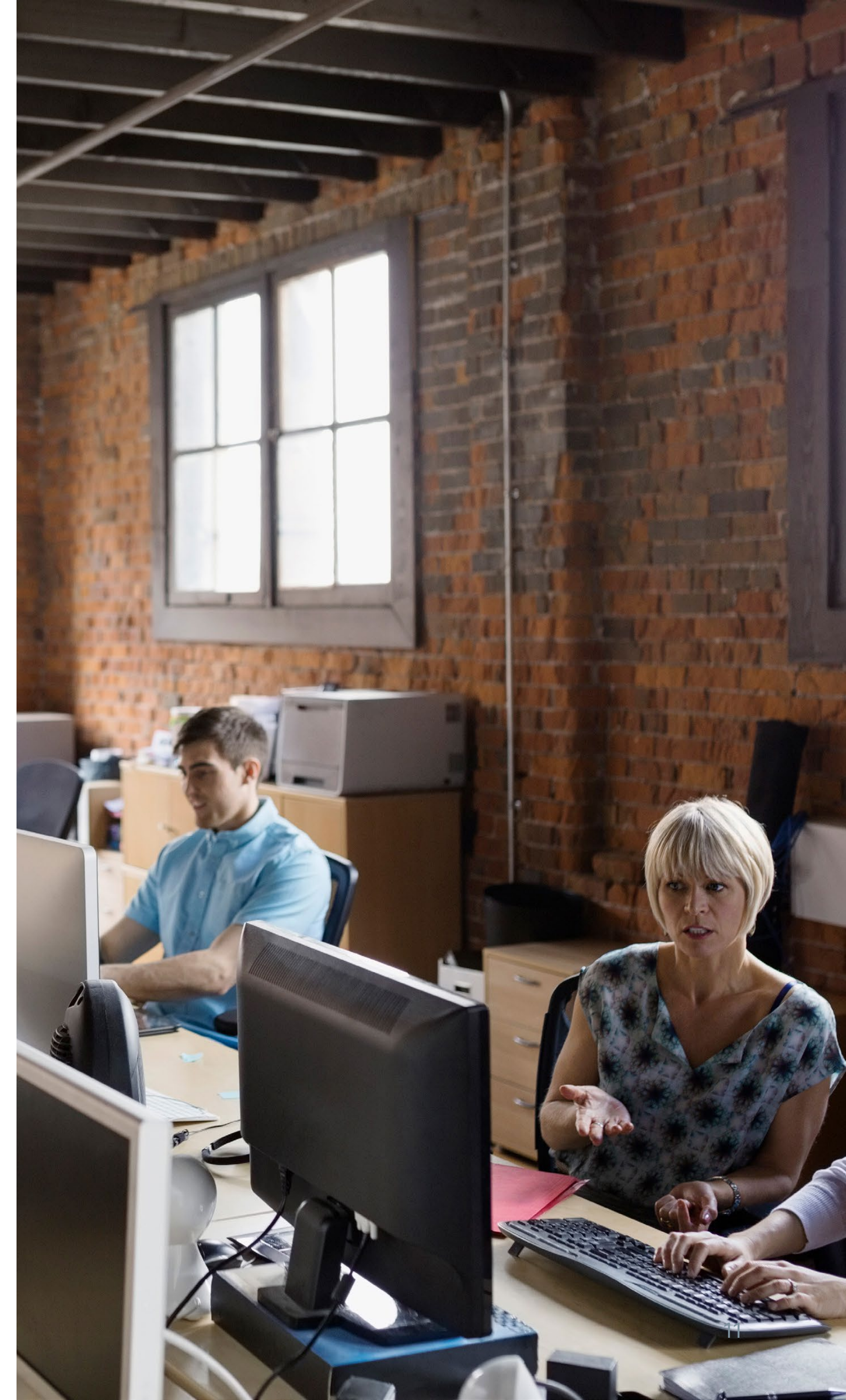
OpenText™ Service Virtualization easily integrates with application life cycle management ([OpenText Application Lifecycle Management](#)), functional testing ([OpenText Functional Testing family](#)), and performance engineering ([OpenText Performance Engineering family](#)) products. It has data and performance modelling capabilities, and it provides unit, functional, and performance testing. It supports large-scale testing and easily integrates with third-party systems. OpenText Service Virtualization also has a cross-service usage capacity and is effective in finance, telecom, utilities, insurance, manufacturing, and other sectors.

# OpenText Functional Testing for Developers

OpenText™ Functional Testing for Developers is a shift-left functional test automation tool that accelerates software delivery. This tool enables easy test creation with your integrated development environment, languages, and testing framework of choice.

It was designed for continuous testing and integration, enabling devtesters and automation engineers to test earlier while improving code quality. OpenText Functional Testing for Developers makes building on already existing tests seamless. It also makes the creation of new, robust, and reusable Selenium tests less tedious.

OpenText Functional Testing for Developers also enables running multiple tests in parallel in physical, virtual, or Dockerized environments. Plus, the embedded service virtualization reduces bottlenecks. OpenText Functional Testing for Developers's embedded service virtualization also provides delay-free, continuous testing with a built-in capability to execute tests with simulated APIs and virtual services—even if the real services are not available, data access is restricted, data is difficult to attain, or the services are not suitable for the particular test.





# Testing Your Own APIs

A comprehensive set of well-tested APIs is necessary for a successful system, and testing with the right tools is fundamental to ensuring your API is stable enough to meet your business challenges and requirements.

OpenText Functional Testing plays a key role in effective API delivery by providing end-to-end testing solutions, visual test-driven design, and extensive support for your existing technologies with automatic API test creation. OpenText products are all carefully designed to boost your application performance.

It's worth spending a few minutes checking out [OpenText Functional Testing](#).

Give it a try today!

## About OpenText

OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit [opentext.com](https://opentext.com).

[opentext.com](https://opentext.com) | [X \(formerly Twitter\)](#) | [LinkedIn](#) | [CEO Blog](#)