



# Leitfaden für das API-Testing

Ein automatisierter Ansatz für die Transformation des API-Testing



**ca**  
technologies

# Vom Monolithen zum Microservice

Die Verbrauchersteuerung (Consumerization) der IT hat die Erstellung von Anwendungen tiefgreifend verändert. Statt ein einzelnes, monolithisches System zu erstellen, das auf einem Server in einem Back Office oder Callcenter installiert wird, werden moderne Anwendungen für zahlreiche Mobile Devices und Webbrowser entwickelt, damit die Anwender jederzeit und von überall auf Services zugreifen können.

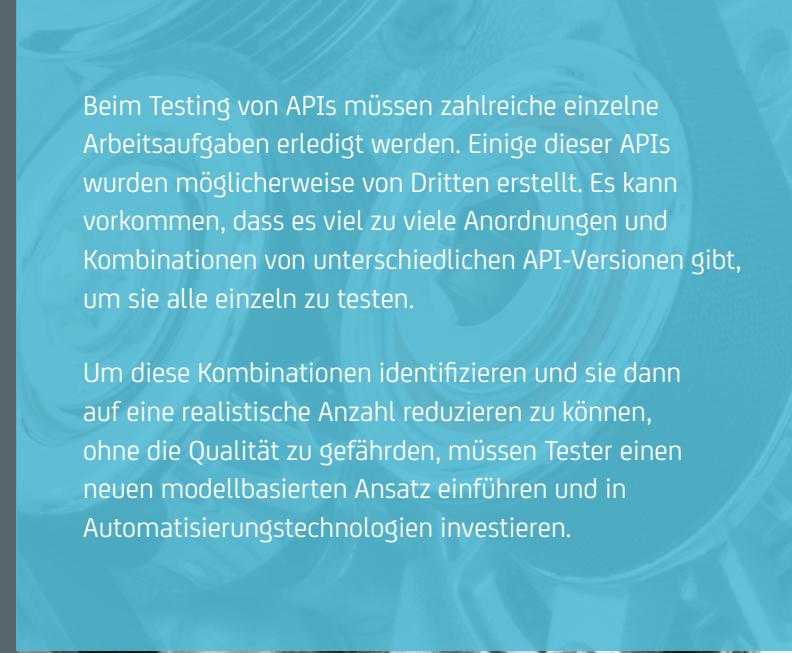
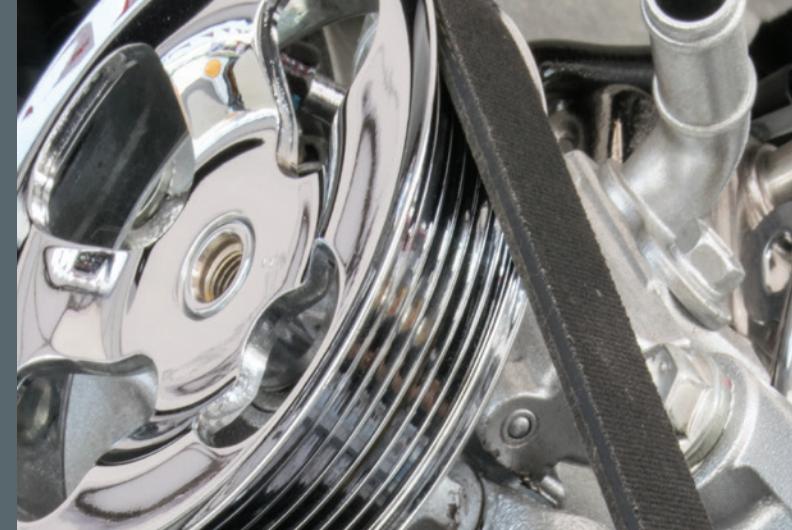
Außerdem sind moderne Anwendungen nicht mehr in sich geschlossen, sondern sie werden aus mehreren Servicekomponenten zusammengestellt, die zur Laufzeit zusammenarbeiten. Daher bilden nicht die eleganten UIs, die wir alle verwenden, den Ursprung der End User Experience, sondern APIs, die Interaktionen zwischen Codemodulen, Anwendungen und IT-Systemen im Back End ermöglichen.

Dies erfordert auch andere Methoden für das Testing. Teams in der Quality Assurance (QA) und Entwicklung müssen ihre Möglichkeiten erweitern, jede Schicht der Anwendung sowie die End-to-End-Geschäftstransaktionen zu testen. Moderne Anwendungen erfordern dann neben dem herkömmlichen UI-Testing ein umfassendes API-Testing. Dies stellt die Tester jedoch vor zahlreiche Herausforderungen. Es erfordert einen neuen Ansatz, der so weit wie möglich automatisiert und von Haus aus datengesteuert ist.



## Weiterführende Lektüre

Artikel aus dem Professional Tester Magazine:  
[Meeting the Challenge of API Testing](#)



# Die Herausforderungen des API-Testing

Möglicherweise sind Sie zunächst geneigt, die Testing-Prinzipien für die UI auch auf APIs auszuweiten: die Entwickler eine UI für jede API erstellen zu lassen und dann „gründlich hinzusehen“, indem Sie manuell Datentabellen eingeben und die Reaktion beobachten. Dieser Ansatz ist jedoch nicht ideal, kostet zu viel Zeit und ist nicht skalierbar. Diese Herausforderungen werden durch mehrere Faktoren noch verschärft:

## Zunahme an Anwendungsfällen

Beim herkömmlichen UI-Testing geht es nur um die Funktionalität der Anwendung insgesamt. Ein Tester führt sie für eine Eingabe aus und interpretiert die Ausgabe im Vergleich zum erwarteten Ergebnis. Das Testing von APIs ist jedoch etwas ganz anderes. Da APIs die Schaltzentrale für die Logik und das Tor vieler Anwendungen zu den Daten bilden, sind die Anwendungsfälle fast unbegrenzt. Daher übersteigt die Anzahl der erforderlichen Tests schnell die Möglichkeiten der Entwicklungs- und Technologietester, die für das Testfalldesign verantwortlich sind.

## Asynchrone und synchrone Prozesse

Moderne Anwendungen sind komplex, und eine einzige API kann mehrere Microservices und andere APIs miteinander verknüpfen. Daher gibt es keine Grenze dafür, wie viele parallele und serielle Aktionen ein einziger API-Aufruf auslösen kann. Beispielsweise kann ein Service die Lieferadresse eines Kunden nachschlagen, während ein zweiter Service den Preis eines Artikels nachschlägt. Ein dritter Service nutzt dann die Ausgabe, um die vom Standort des Käufers abhängige Verkaufssteuer und die Kosten des Artikels zu berechnen. In diesem Beispiel wird angenommen, dass die Anwendung eine API aufruft. In Wirklichkeit besitzen moderne Anwendungen APIs, die Daten von anderen APIs abrufen, die wiederum auf anderen basieren. Die Komplexität einer API kann daher exponentiell ansteigen, indem sie mit Aufrufen anderer APIs kombiniert wird. Diese Komplexität wird noch um ein Vielfaches höher, da Tester beim Testfalldesign auch die Aufrufreihenfolge der APIs berücksichtigen müssen. Daher muss das Testing von APIs sowohl in grundlegenden, eigenständigen Arbeitsschritten als auch orchestriert durchgeführt werden.

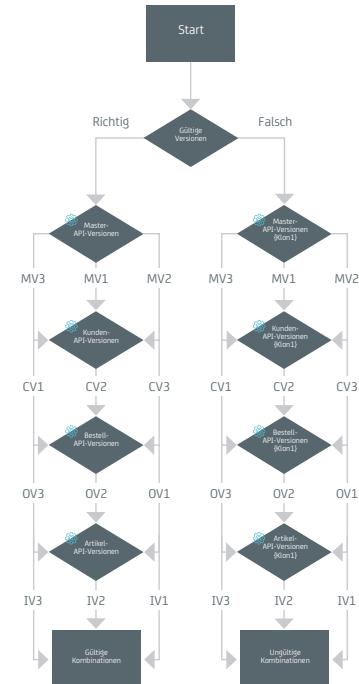
# Die Herausforderungen des API-Testing (Fortsetzung)

## Zugriff auf abhängige Systeme

APIs rufen Daten von mehreren APIs, Microservices und Back-End-Systemen ab. Die entstehende Architektur kann aussehen wie das breite und tiefe Wurzelwerk eines Baumes. Es ist nicht anzunehmen, dass Sie über Zugang zu jeder Umgebung in diesem Baum verfügen. Einige Abhängigkeiten sind möglicherweise noch in Entwicklung, während Drittanbietersysteme und Mainframes zu teuer zum Testen wären oder nur in begrenztem Zeitrahmen und Umfang zugänglich sind. Nun ist es wichtig, nicht verfügbare Ressourcen zu simulieren, um Engpässe beim Testing zu vermeiden. Die herkömmliche Methode der Erstellung kundenspezifischer Mocks und Stubs stellt sich jedoch als zu problematisch heraus. Beispielsweise werden Stubs und Mocks im Allgemeinen von Entwicklern geschrieben und ab und zu an das QA-Team weitergegeben – aber sehr selten an andere Entwicklungsteams, da dies zu Interoperabilitätsproblemen führen würde. Außerdem können sie keine Bedingungen simulieren, unter denen Konkurrenz um Ressourcen besteht. Für ein umfassendes Performance- und Belastungs-Testing ist dies jedoch erforderlich.

## API-Versionen

Unterschiedliche Versionen sind ein weiterer Grund für die steigende Komplexität des API-Testing. Die meisten Systeme arbeiten mit einem Konzept von Veraltung, sodass eine API auch funktionieren muss, wenn eine alte Version neue Versionen oder eine Kombination unterschiedlich alter Versionen aufruft. Die API muss fehlende Werte erkennen und für sie Standardwerte zuweisen, damit die alten Versionen funktionieren. Außerdem ist es möglich, dass einige Versionen nur von bestimmten Versionen aufgerufen werden können. Daher müssen die zahlreichen möglichen Kombinationen getestet werden.



**Ein Beispiel:** Im Diagramm müssen 127 API-Kombinationen getestet werden. Unter Berücksichtigung der unterschiedlichen Versionen sind jedoch 10.287 mögliche Kombinationen zu testen, und es ist unwahrscheinlich, dass mit manuell erstellten Scripts ein ausreichend großer Anteil hiervon abgedeckt werden kann.

# Ein Modell für die Transformation des API-Testing

Um diese Herausforderungen zu bewältigen, benötigen Sie einen modernen Ansatz und eine Reihe von Tools, mit denen Sie API-Tests automatisieren und in einen kontinuierlichen Bereitstellungszyklus einbinden können.



„Eine umfangreiche Testautomatisierung erfordert Testmodelle, Testdatengeneratoren und automatische Orakel. Modeling, Simulation, Analysen, Visualisierung und toolgestützte Entscheidungsfindung werden zu wichtigen Leistungsmerkmalen für Testarchitekten und Testing-Teams. Tester müssen lernen, wie sie mit stärker technisch orientierten Modeling- und Simulationstools bessere Testmodelle erstellen und verwenden können.“<sup>1</sup>

Paul Gerrard, Gerrard Consulting



## Weiterführende Lektüre

- [How Will the Internet of Things Affect Testers?](#)
- [Berichte von Branchenanalysten: Gartner Market Guide for API Testing and Service Virtualization](#)



# Automatische Erzeugung von API-Tests auf mehrere Weisen

In einer perfekten Welt könnten Sie alle für eine angemessene Abdeckung benötigten Testfälle automatisch ableiten und dann relevante Test-Scripts erzeugen. Zugleich könnten Sie die geeigneten Datenmengen für die Ausführung dieser Tests automatisch provisionieren. Fortschritte der Testing-Technologie haben dafür gesorgt, dass das Ideal nicht weit entfernt ist. Einige manuelle Eingriffe werden jedoch weiterhin erforderlich sein. Daher ist die Frage: Wie minimieren Sie diese Anforderung?

- Implementieren Sie zuerst ein System, mit dem Sie alle API-Anforderungen und -Änderungsanforderungen formell erfassen und verwalten können.
- Beschaffen Sie sich die geeigneten Tools, um Testfälle und zugehörige Test-Scripts automatisch und auf mehrere Weisen aus diesen Anforderungen zu erstellen: beispielsweise aus Verträgen (Swagger), aus Stichproben (individuellen Nachrichten) oder aus Beobachtungen an realen API-Daten.
- Speichern Sie Testfälle zur Wiederverwendung in einer Bibliothek. Es ist wichtig, alle Testfälle den entsprechenden Anforderungen zuzuordnen. Nur so erhalten die Tester den notwendigen Kontext, um zu verstehen, welche Vorgänge die API ausführt und wo die Transaktion im Gesamt-Workflow verortet ist.
- Nutzen Sie Technologien für das Test Data Management, um vorhandene Daten aus mehreren Back-End-Systemen abzurufen und zu maskieren und bei Bedarf fehlende Daten synthetisch zu extrahieren, um eine maximale Testabdeckung zu erzielen.

70 %  
aller Tests werden noch manuell  
durchgeführt.<sup>2</sup>

# Bearbeitung und Erweiterung von API-Tests ohne Schreiben von Code

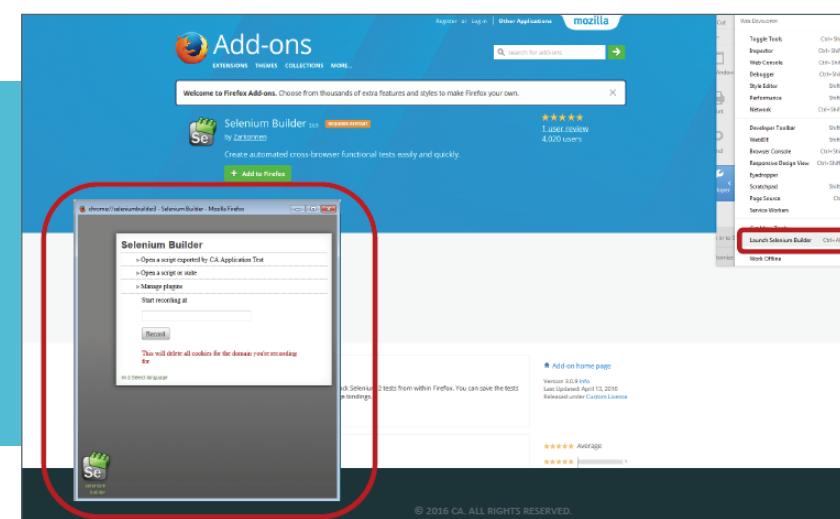
Mit Scripts erstellte automatisierte Tests hängen vom zu testenden System ab. APIs werden von Zeit zu Zeit geändert. In diesem Fall funktionieren automatisierte Tests für sie möglicherweise nicht mehr, sodass auch die Scripts geändert werden müssen. Dieser Prozess kann schließlich zu großen, kaum verwaltbaren und fehleranfälligen Codemengen führen. Zugleich sind moderne Anwendungen kompliziert, und Tests müssen auf mehreren Anwendungsschichten überprüft werden. Dies erfordert eine weit umfangreichere Testorchestrierung.

Mit den richtigen Testautomatisierungstools und dem richtigen Framework können Sie API-Tests jedoch bearbeiten und erweitern, ohne eine einzelne Codezeile zu verändern.

- Sehen Sie sich nach einer API-Testing-Lösung um, die Modularität ermöglicht, um die Wiederverwendbarkeit zwischen unterschiedlichen Releases zu erleichtern. Die Konfiguration (Umgebung, IPs, Anmeldenamen/Passwörter) sollte von den Daten getrennt gespeichert werden. Diese sollten außerdem getrennt von der Aufrufmethode (Protokollen wie Webservices, MQ, JMS, TIBCO, SQL usw.) verwaltet werden. Wenn Sie in der Lage sind, einen dieser Bereiche separat von den anderen beiden zu verändern, können Sie die Tests wiederverwenden und an die sich schnell ändernde API-Landschaft anpassen.
- Wählen Sie eine API-Testing-Lösung mit Selenium-basiertem Web-UI-Testing und einem systemeigenen grafischen Editor. Mit solchen Tools können Sie mit der Selenium-UI entwickelte Tests laden und sie für mehrere Browser skalieren.

CA Application unterstützt eine nahtlose Integration in Selenium. Sie können webgestützte Aufzeichnungen oder Testfälle von Selenium Builder in CA Application Test importieren. Den importierten Test können Sie dann in Mozilla Firefox, Google Chrome, Apple Safari oder Microsoft Internet Explorer® ausführen.

[Online-Demonstration anzeigen](#)



Weiterführende Lektüre

[Automated Testing: Coping with Change](#)

# Verwendung nur eines Tests und eines Tools für alle Schichten und Systeme

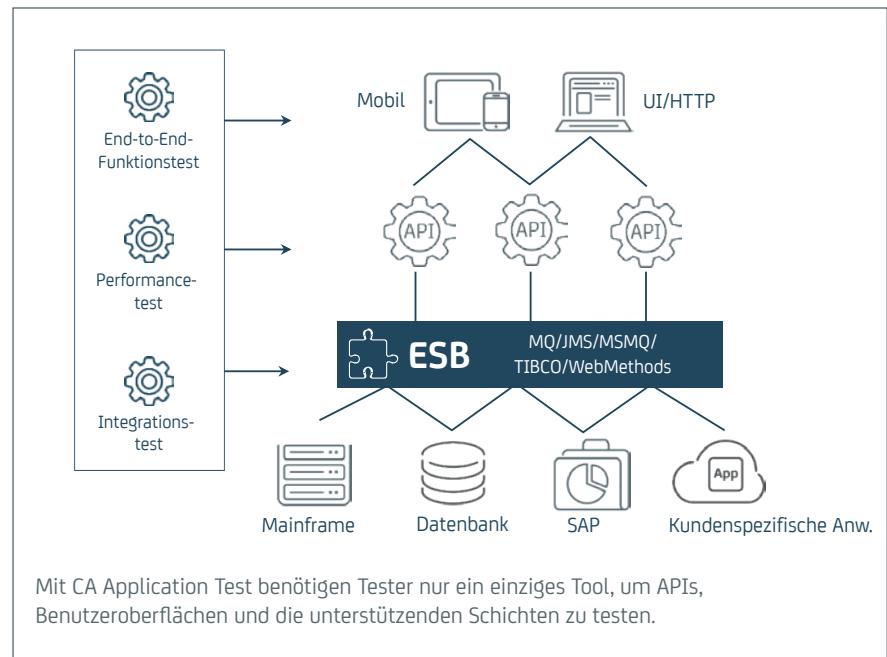
Tests müssen Ihnen mehr Arbeit abnehmen. Ein Test sollte nicht nur einzelne API-Aufrufe und zugehörige Reaktionen auslösen und überprüfen, sondern auch ein nicht funktionsbezogenes Testing ermöglichen und mit Open-Source-Technologien wie Selenium zusammenarbeiten.

Hierfür benötigen Sie ein erweiterbares Testing-Framework, mit dem Ihre Teams moderne RESTful-APIs, SOA und ESBs ebenso wie eher herkömmliche Protokolle wie JMS, MQ und Java™ sowie Datenbanken testen können. Außerdem müssen Systeme, Designer und Implementierer zusammenarbeiten, um Testfälle miteinander zu verknüpfen, damit sie ein End-to-End-Testing für einen Unternehmensprozess ausführen können.

Mit dem richtigen Testing-Framework können Sie:

- die Ausgabe aus einem Test als Eingabe für nachfolgende Tests verwenden
- umfangreiche Leistungsmerkmale nutzen, um die Zusammenarbeit zwischen Entwicklern und Analysten beim Testing von Assets und Daten zu erleichtern, sowie ein leistungsstarkes Framework einsetzen, um Anfragen an unterschiedliche Services zu stellen und zu überprüfen
- Testfälle zu Workflows verbinden, um vollständige Unternehmensprozesse nachzuahmen
- vorhandene Funktionstests als Performancetests wiederverwenden
- mehrere Tests gleichzeitig ausführen, um Funktionspfade sowie Back-End-APIs und -Services zu überprüfen
- Tests planen, um die Stabilität der Umgebung zu überprüfen

## Automatisiertes API-Testing des gesamten Softwarestacks ohne Scripts



Mit CA Application Test benötigen Tester nur ein einziges Tool, um APIs, Benutzeroberflächen und die unterstützenden Schichten zu testen.

„Um kontinuierliche Qualität und schnellere Entwicklung zu unterstützen, vor allem in Agile- und DevOps-Kontexten, muss das Testing von einem rein UI-orientierten Ansatz zu UI und APIs übergehen.“<sup>3</sup>

The Forrester Wave™: Modern Application Functional Test Automation Tools, Q2 2015



Weiterführende Lektüre  
Der Preis des Misserfolgs

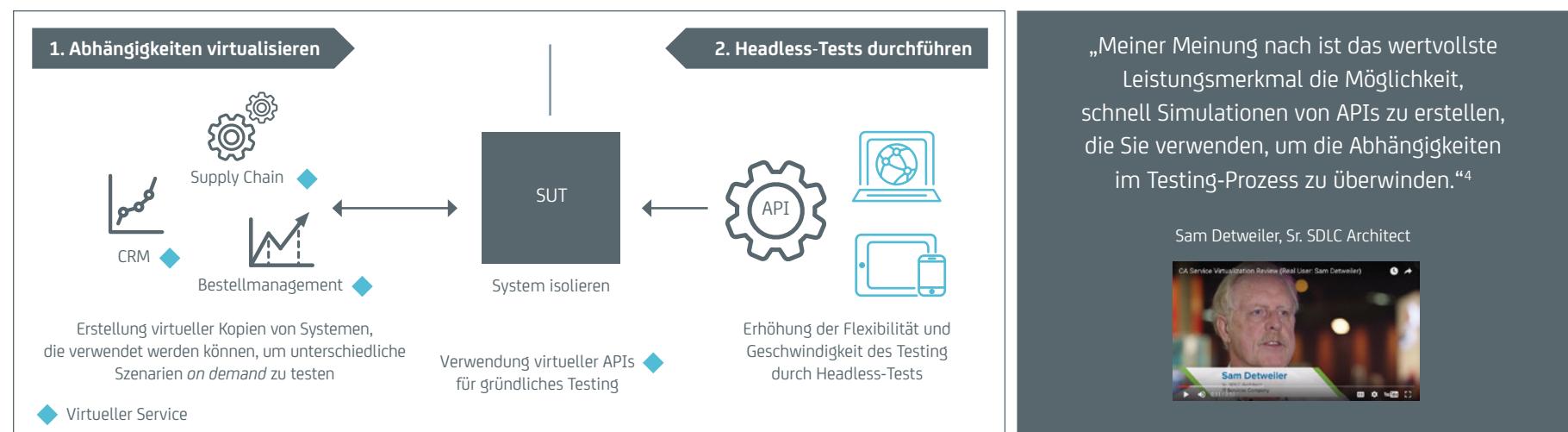
# Durchführung unbegrenzt vieler Tests mit Service-Virtualisierung

Die Service-Virtualisierung ist ein Verfahren, mit dem das Verhalten, die Daten und die Performancemerkmale eines abhängigen Systems simuliert werden. Diese Simulation können Sie in allen Phasen von Entwicklung und Testing verwenden. Im Gegensatz zu Stubs und Mocks werden virtuelle Services erstellt, indem tatsächliche Services anhand von Verträgen oder von Beispielenfragen und -antworten aufgezeichnet werden. Sie sind dynamischer und müssen nicht von einem Entwickler geschrieben werden.

Mit der Service-Virtualisierung können Sie jederzeit einen praktisch unbegrenzten Satz von Testfällen und -szenarien verwenden. Sie können Tests einrichten, um alle unabhängigen Services und Endpunkte zu überprüfen – nicht jeden einzeln in einem Vakuum, sondern alle zusammen. Sie können also zugleich die API testen, die Inventardaten abruft, die Einkaufswagen-API und so weiter – so, wie der Kunde sie verwenden soll. Die Service-Virtualisierung befreit Sie von Aufruflimits, sodass Sie testen und testen und weiter testen können, bis das Debugging Ihrer App wirklich abgeschlossen ist.

Der Zugriff auf virtualisierte APIs ist sogar besser als der auf echte, weil Sie mit ihnen Szenarien aller Arten testen können – mit unterschiedlichen Levels von Funktionalität, Performance und Maintenance – was mit einer echten API nicht möglich wäre. Virtualisierte APIs können jedes Verhalten zeigen, das Sie möchten.

## Virtualisierung von Abhängigkeiten und APIs zur Erreichung echter Testbarkeit



# Eine vollständig integrierte End-to-End-API-Lösung

Das API-Testing ist komplex. Ein automatisierter Ansatz ist erforderlich, um so gründlich arbeiten zu können, wie es skalierbare Testing-Frameworks erfordern, die mit der hohen Änderungsrate mithalten können. CA Technologies bietet die Tools, die Ihre Teams benötigen, um APIs gleichzeitig zu entwickeln, zu managen und zu testen.

## CA Test Data Manager und CA Agile Requirements Designer

CA Test Data Manager ermöglicht die Erstellung produktionsähnlicher Testdaten und ihre fortlaufende Synchronisierung für alle Systeme, die sie während der unterschiedlichen Testphasen erreichen. Dies eliminiert einen großen Teil des manuellen Arbeits- und Zeitaufwands für die Unterteilung, Maskierung und Anpassung von Daten für jeden Anwendungsfall und jede Datenaktualisierung.

CA Agile Requirements Designer ist eine automatisierte Modeling-Lösung, mit der Sie sich an veränderliche API-Anforderungen und -Designs anpassen können. Sie können identifizieren, welche Tests ausgeführt werden sollen, und dabei die richtige Anzahl Anwendungsfälle ermitteln, die sich nur auf die Bereiche beziehen, in denen die Anwendung geändert wurde. So können Sie die Testabdeckung maximieren, ohne übermäßig zu testen.

Wenn eine API-Spezifikation verändert wird oder Testfälle aktualisiert werden, kann CA Agile Requirements Designer virtuelle Daten schnell aktualisieren, um die maximale Abdeckung zu wahren. Wenn dem Flowchart eine neue Variable hinzugefügt wird, werden ihre Auswirkungen auf das Datenmodell automatisch genau identifiziert. Alle neuen Variablenkombinationen, die für eine hundertprozentige Abdeckung erforderlich sind, können mit CA Test Data Manager erstellt werden. Beschädigte oder ungültige Kombinationen werden entfernt oder repariert. Dies eliminiert die Zeitverschwendungen für manuelle Daten-Maintenance und bietet Ihren Testern die aktuellen Daten und Umgebungen, die sie für ein gründliches Testing von APIs benötigen.

## Ein modernes Toolset für die Transformation des API-Testing

### CA Test Data Manager

Finden, erstellen und provisionieren Sie Testdaten automatisch und schnell.



### CA Agile Requirements Designer

Erzeugen und pflegen Sie automatisch den kleinsten Satz von Testfällen, der eine maximale Abdeckung ermöglicht.

### CA Application Test

Bearbeiten und erweitern Sie API-Tests, ohne Code zu schreiben, und testen Sie den gesamten Technologiestack.

### CA Service Virtualization

Virtualisieren Sie APIs von Drittanbietern sowie nicht verfügbare Systeme, um agil testen zu können.



### CA API Management

Verwalten Sie APIs, und erstellen Sie ein Verzeichnis virtueller APIs für den sicheren Zugriff.



## Weiterführende Lektüre

[Gründliches und schnelleres Testing von Composite-Anwendungen mit CA Test Data Manager und CA Agile Requirements](#)

# Eine vollständig integrierte End-to-End-API-Lösung

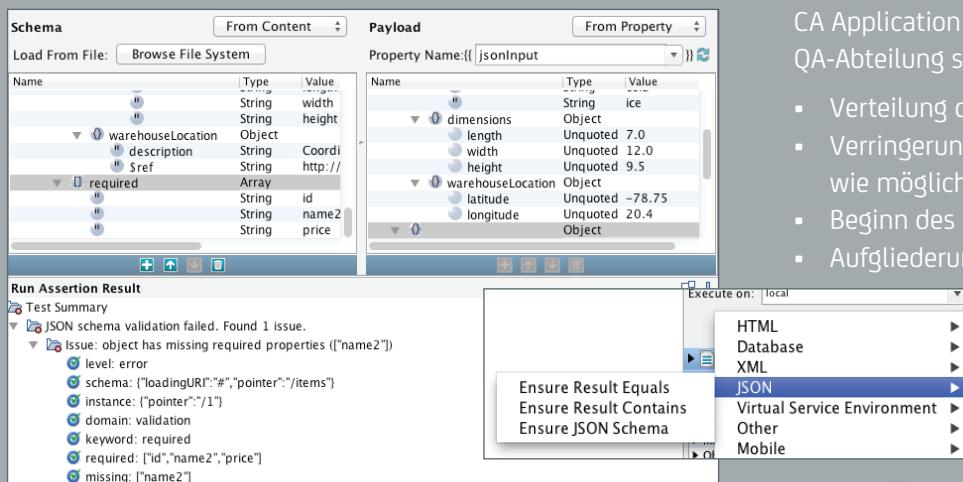
## CA Application Test

ist eine Testing-Plattform für mehrere Protokolle. Sie unterstützt Webservices, Nachrichtenwarteschlangentechnologien und andere Endpunkte, wie Datei- und Dateiübertragungs-Endpunkte und Datenbanken. CA Application Test ermöglicht Ihnen Folgendes:

- **Umfassende Testabdeckung:** Sie können REST, JSON und SOAP sowie eine Reihe von Nachrichtenschnittstellen und anderen Endpunkten abdecken, wie Datenbanken, Flatfiles, EJBs usw.
- **Einfache Erstellung und Verwaltung von API-Tests:** Verwenden Sie eine grafische Visualisierung für JSON-basierte Nutzlasten und Pfaderzeugung oder einen anwenderfreundlichen Assistenten für die Erzeugung von Testfällen für Webservice-APIs.
- **Zusammenarbeit über sehr produktive visuelle Tests:** Testen Sie mittlere „Headless“-Hierarchieebenen, also solche ohne UIs.
- **Anpassung von Belastungstests:** Richten Sie Parameter ein, testen Sie REST-Services und andere Services, führen Sie Hunderte von Simulationen zugleich durch.

Mit CA Application Test benötigen Tester nur ein einziges Tool, um UIs, APIs und die unterstützenden Schichten der Implementierung zu testen.

Anwender können einen einzelnen Testfall mit Schritten modellieren, die das Verhalten und die Performance einer API sichtbar machen und überprüfen. Außerdem können sie Datenbankaufrufe, SOAP-Nachrichten und viele weitere Transaktionstypen überprüfen.



CA Application Test ist nicht die automatisierte Testing-Lösung, die Ihre QA-Abteilung sich früher einmal vorgestellt hat. CA Application Test ermöglicht:

- Verteilung der Tests über die Architektur
- Verringerung der Komplexität, indem das Testing so nah an der Logik erfolgt wie möglich
- Beginn des Funktions-Testings vor Abschluss der Entwicklung
- Aufgliederung langer Reaktionszeiten in kleine Teile

# Eine vollständig integrierte End-to-End-API-Lösung

## CA Service Virtualization und CA API Management Suite

Zusammen beschleunigen [CA Service Virtualization](#) und [CA API Management](#) die Anwendungsbereitstellung, indem sie Lösungen für die Planungs-, Erstellungs- und Ausführungsphase des Lebenszyklus von Anwendungen und Services bereitstellen.

Das [CA API Developer Portal](#) bietet Entwicklern die Möglichkeit, virtuelle Services über ein Verzeichnis zu erkennen. Dies verbessert die Nutzung extern verfügbarer Services. Services und APIs können jetzt schneller und mit besserer Dokumentation erkannt werden, um die Entwicklung zu planen.

CA Service Virtualization und [CA API Gateway](#) bieten ein intelligentes, richtlinienbasiertes Routing von Datenverkehr zwischen „realen“ und „virtuellen“ Services. Mit CA API Gateway können Organisationen außerdem Sicherheitsstandards wie OAuth nutzen, damit Entwickler sich auf die Geschäftslogik konzentrieren können, statt Sicherheitsfunktionen direkt in die Anwendung integrieren zu müssen.

CA API Gateway und CA Service Virtualization bieten einen Echtzeit-Datenfluss von der Produktion zur Entwicklung. Dies ermöglicht eine schnellere, präzisere Erstellung virtueller Services. Je umfassender, kontextbezogener und genauer die Daten sind, umso besser ist die virtualisierte Umgebung, die für Entwicklung und Testing erstellt werden kann.



CA Technologies hat Forrester Consulting mit einer Total Economic Impact™-Studie (TEI) zum potenziellen ROI beauftragt, den Unternehmen mit CA Service Virtualization erzielen können. Forrester befragte vier IT-Abteilungen, die diese Lösung für ihr Unternehmen implementiert hatten. Insgesamt berichtete das daraus konstruierte fiktive Unternehmen:<sup>5</sup>

- frühe Fehlererkennung bei der Quality Assurance, die zu durchschnittlichen jährlichen Einsparungen von 1 Mio. US-Dollar führt
- höhere Effizienz bei Entwicklung und Testing, die jährliche Einsparungen von 306.000 US-Dollar ermöglicht
- Vermeidung von Infrastrukturkosten für die Testing-Umgebung in Höhe von 2.747.971 US-Dollar



[Vollständige Studie lesen](#)

# Weitere Informationen erhalten Sie unter:

## <http://www.ca.com/de/products/develop-test.html>

CA Technologies (NASDAQ: CA) entwickelt Software, die Unternehmen bei der Umstellung auf die Application Economy unterstützt. Software steht in allen Branchen und in allen Unternehmen im Mittelpunkt. Von der Planung über die Entwicklung bis hin zu Management und Security arbeitet CA Technologies weltweit mit Unternehmen zusammen, um die Art, wie wir leben, Transaktionen durchführen und kommunizieren, neu zu gestalten – ob mobil, in der privaten oder öffentlichen Cloud oder in verteilten Systemen oder Mainframe-Umgebungen. Weitere Informationen finden Sie unter [ca.com/de](http://ca.com/de).

<sup>1</sup> Gerrard Consulting im Auftrag von CA Technologies, „How Will the Internet of Things Affect Testers?“ Juni 2016,  
<http://www.ca.com/content/dam/ca/us/files/white-paper/how-will-the-internet-of-things-affect-testers.pdf>

<sup>2</sup> Bloor Research, „Automated Test Case Generation“, September 2014,  
<https://www.ca.com/de/collateral/industry-analyst-report/bloor-research-spotlight-paper-automated-test-case-generation.register.html>

<sup>3</sup> Forrester, „Forrester Wave™: Modern Application Functional Test Automation Tools, Q2, 2015“, 17. April 2015,  
<https://www.forrester.com/report/The+Forrester+Wave+Modern+Application+Functional+Test+Automation+Tools+Q2+2015/-/E-RES115627>

<sup>4</sup> IT Central Station, CA Service Virtualization Review (Real User: Sam Detweiler), 15. März 2016,  
[https://www.itcentralstation.com/product\\_reviews/ca-service-virtualization-review-36610-by-sam-detweiler](https://www.itcentralstation.com/product_reviews/ca-service-virtualization-review-36610-by-sam-detweiler)

<sup>5</sup> Forrester Total Economic Impact™-Studie im Auftrag von CA Technologies, „The Total Economic Impact™ of CA Technologies Service Virtualization“, Dezember 2015,  
<http://www.ca.com/content/dam/ca/us/files/industry-analyst-report/the-total-economic-impact-of-ca-service-virtualization.pdf>