

Citrus – Integrationsframework mit Demonstration

Christoph Deppisch¹

Abstract: Citrus ist ein Open Source Framework für den automatisierten Test von Schnittstellen einer modernen Enterprise Anwendung. Das Framework ist in Java geschrieben und bietet fertige Komponenten für den Nachrichtenaustausch als Client und Server mit verschiedensten Protokollen und Formaten (z.B. Http REST, SOAP, JMS, FTP, File, Websocket, XML, JSON und vieles mehr). Das Framework bietet eine XML sowie Java DSL für das Schreiben von voll automatisierten Integrationstests, die sich einfach in den Continuous Build Lifecycle einbinden lassen. Der Vortrag zeigt eine kurze Einführung in das Framework und demonstriert die Arbeitsweise anhand eines Live-Code Beispiels.

Keywords: Automatisiertes Testen, Schnittstellentests, Integrationstests, Open Source, Messaging, Microservices

1 Einleitung

Softwarelandschaften bestehen heutzutage nicht mehr aus wenigen schwergewichtigen Monolithen, sondern aus vielen eigenständigen Komponenten und Anwendungen. Die Anzahl der an einem Geschäftsprozess beteiligten Softwarekomponenten steigt zunehmend an. Trends wie Microservices zeigen uns wie man schwergewichtige Alleskönner in mehrere kleinere Services aufteilt und so die Vorteile von mehr Flexibilität und schnelleren Releasezyklen ausschöpfen kann.

Mit diesem Trend hin zur verteilten Anwendung steigt natürlich auch die Anzahl der beteiligten Schnittstellen. Alle Komponenten tauschen Daten über wohl definierte Schnittstellenverträge miteinander aus. Diese nachrichtenbasierten Schnittstellen (asynchron wie synchron) nehmen einen zunehmend essentiellen Platz in der Softwareentwicklung ein.

Nahezu jede Software besitzt Schnittstellen nach Außen zu Partnern und anderen Komponenten im Unternehmen. Diese Kommunikation mit dem zugehörigen Datenaustausch muss automatisiert getestet werden, um eine verlässliche Aussage über die Qualität des gesamten Bildes von mehreren zusammenarbeitenden Services zu treffen.

¹ ConSol Software GmbH, Development, Franziskanerstr. 38, D-81669 München,
christoph.deppisch@consol.de

2 Herausforderungen im Integrationstest

Unit-Tests sind in der Regel klein und schnell in der Ausführung. In voller Isolation zu anderen Einheiten (Klassen, Methoden) wird ein Code-Abschnitt durchlaufen und überprüft. Im Integrationstest ist diese Isolation nicht mehr gegeben. Hier wollen wir gerade die Zusammenarbeit mehrerer Komponenten im Test nachstellen und überprüfen. Daher hat der Integrationstest naturgemäß Abhängigkeiten und eine komplexere Infrastruktur als Grundlage.

Die Software wird zum Beispiel im Vorfeld des Tests in einem Application Server deployed. Eventuell wird ein Message Broker oder ein Webserver benötigt. Schnittstellenpartner stehen während des Tests nicht zur Verfügung und müssen simuliert werden. Das alles stellt Entwickler und Tester vor einige Herausforderungen, ins Besondere wenn an einem Geschäftsprozess mehrere Schnittstellen unterschiedlicher Art beteiligt sind.

3 Citrus Framework

Das Citrus Open Source Framework für den automatisierten Integrationstest hat sich zum Ziel gesetzt die zuvor angesprochenen Herausforderungen im Schnittstellentest zu meistern. Mit fertigen Komponenten für unterschiedliche Nachrichten und Transportwege bietet das Framework alles, um den Datenaustausch zwischen verschiedenen Services zu testen.

Als Client oder Server verbindet sich Citrus mit dem System Under Test (SUT) und verschiebt Nachrichten über die wirklichen Transportwege (z.B. Http, JMS, FTP, ...). Das SUT wird dabei möglichst produktionsnah gestartet und arbeitet mit der zu testenden Schnittstelle. So können die Regeln der Schnittstelle (WSDL, XSD) im Test überprüft werden und auch die fachlich korrekte Arbeitsweise der Services sichergestellt werden.

Citrus ist nämlich in der Lage alle Nachrichten des SUT mit erwarteten Inhalten zu überprüfen. Dabei bietet das Framework fertige Validatoren für XML, JSON, XPath, JsonPath und HTML. Die ausgetauschten Inhalte werden damit überprüft und auf Regression getestet.

Citrus ist als Open Source Framework unter der Apache Lizenz 2.0 frei verfügbar. Nähere Informationen zum Framework selbst sind in [CF16] abrufbar. Der freie Code des Frameworks ist unter [Gi16] verfügbar.

4 Das Beispiel

Der Vortrag demonstriert die Arbeitsweise von Citrus an einem Code Beispiel. Im

Beispiel werden mehrere Schnittstellen einer verteilten Microservices Anwendung in einem automatisierten Test überprüft. Dabei kommen unterschiedliche Transportwege (z.B. JMS, Http REST) zum Einsatz und die Validierung von XML und JSON Nachrichteninhalten wird exemplarisch vorgeführt.

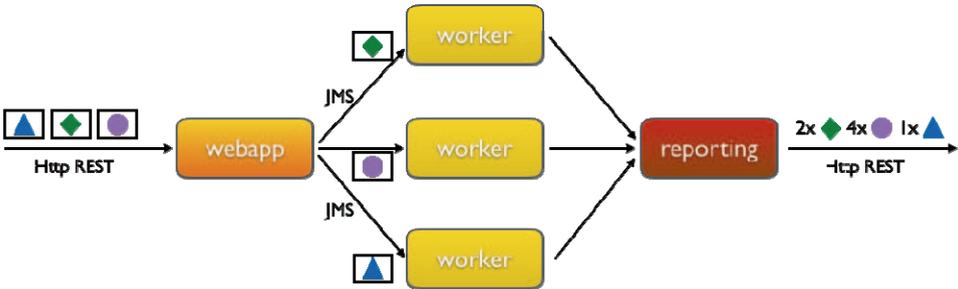


Abb. 1: Beispiel Anwendung

Die zu testende Anwendung besteht aus mehreren Komponenten, welche Nachrichten untereinander austauschen. Eine Web-Anwendung nimmt Aufträge über eine Http REST Schnittstelle entgegen und verteilt diese, je nach Art des Auftrags, an so genannte Worker Komponenten. Nachdem ein Worker seine Arbeit erfolgreich beendet hat, teilt er dies einem zentralen Reporting Server mit, so dass man die Aufträge eines Tages im Reportüberblick verfolgen kann.

Im Vortrag wird ein voll automatisierter Test dieser Vorgänge gezeigt. Der Test baut dabei die benötigte Infrastruktur auf und versendet echte Nachrichten über die Transportwege. Die ausgetauschten Nachrichteninhalte werden automatisch syntaktisch und semantisch überprüft, damit die Integrität aller Schnittstellen im gesamten Bild gewährleistet ist.

5 Über die Autoren

Dipl. Inf (FH) Christoph Deppisch arbeitet als Consultant und Software Architekt bei der ConSol Software GmbH und verfügt über mehr als zehn Jahre Erfahrung bei der Umsetzung großer Enterprise Projekte. Er hat sich auf die Bereiche Middleware Integration und Testautomatisierung spezialisiert und interessiert sich besonders für neue Technologien und Frameworks.

Als aktiver Open Source Entwickler ist Christoph verantwortlich für das Test Framework „Citrus“ und entwickelt dies stetig weiter. In letzter Zeit beschäftigt sich Christoph vor allem mit der Frage welche Einflüsse Microservices und Container auf die Continuous Delivery Pipeline haben.

Literaturverzeichnis

- [CF16] Citrus Framework, <http://citrusframework.org>, Stand: 30.06.2016.
[Gi16] Github, <http://github.com/christophd/citrus>, Stand: 30.06.2016.