Engineer

Vertragstest

Prüfung vereinbarter Schnittstellen zwischen Komponenten

Was es ist

Ein Vertragstest prüft, ob sich zwei Komponenten über die ausgetauschten Anfragen, Antworten oder Nachrichten einig sind. Eine Anwendung zur Paketverfolgung und ihr Dienst können isoliert fehlerfrei arbeiten. Die Oberfläche scheitert dennoch, wenn sie ein Feld namens `status` erwartet und der Dienst `state` sendet. Eine Prüfung der Schnittstelle beschreibt, was die Oberfläche verwendet, und überprüft, ob der Dienst es liefern kann.

Pact beschreibt einen Ansatz, bei dem der Verbraucher gegen einen kontrollierten Ersatz des Anbieters ausgeführt wird. Der Test zeichnet Interaktionen in einem Vertrag auf. Der Anbieter prüft den Vertrag mit seiner tatsächlichen Implementierung unter den festgelegten Bedingungen. Bei Nachrichten betrifft der Nachweis die erzeugte und gelesene Nachricht. Beide Seiten können sich unabhängig weiterentwickeln und ausgeliefert werden, wenn der Nachweis zu den gemeinsam eingesetzten Versionen passt.

Ein vorher vereinbarter Umgang mit unbekannten Angaben kann Teil des Vertrags sein. Eine Anzeige wie „Status nicht verfügbar“ kann für zusätzliche Werte sinnvoll sein. Ob sie ausreicht, entscheidet das Produkt anhand der Aufgabe des Lesers. Der Test zeigt, wie sich der konkrete Verbraucher in diesem Fall verhält.

Der Begriff wird auch für Prüfungen eines Anbieters gegen seine veröffentlichte Schnittstellenbeschreibung verwendet, etwa ein OpenAPI-Dokument. Das ist sinnvoll, wenn die Spezifikation den gemeinsamen Vertrag darstellt. Sie kann Felder, Statuscodes und erlaubte Werte festlegen. Verbrauchergetriebene Beispiele zeigen zusätzlich, welche Kombinationen ein bestimmter Aufrufer

verwendet und welche Annahmen er trifft. Beide Ansätze haben ihren Platz. Ein echter Integrationsdurchlauf prüft außerdem Routing, Zugang und Konfiguration.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Vertragstest?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Der Teil, den Nutzer nie sehen: wo die Daten liegen, welche Regeln gelten und wie andere Systeme darauf zugreifen dürfen.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird. 6 ETL/ELT-Pipelines entworfen und betrieben — 6 ETL-Pipelines über drei Datenbankengines hinweg

6 ETL/ELT-Pipelines entworfen und betrieben Kontakt aufnehmen →

Warum es wichtig ist

Eine Schnittstelle bleibt häufig länger bestehen als die erste Implementierung. Das Dienstteam kann ein Feld umbenennen und seine eigenen Tests aktualisieren, ohne eine separat gepflegte Anwendung zu ändern. Eine mobile Anwendung kann Monate in einer älteren Version betrieben werden. Selbst ein anderer Typ ist wichtig, wenn damit eine sichtbare Meldung ausgewählt wird. Überprüfbare Verträge zeigen solche Abhängigkeiten vor der Freigabe.

Bei unterschiedlichen Freigabefolgen ist das Wissen über verträgliche Versionen besonders wertvoll. Ein älterer Verbraucher kennt vielleicht nur wenige Statuswerte, während der neue Dienst einen Zwischenstatus benötigt. Die Teams können zuerst den Verbraucher anpassen, während der Umstellung beide Formen verarbeiten und altes Verhalten erst entfernen, wenn keine abhängige Version es nutzt. Die Reihenfolge hängt von Verträgen, tatsächlicher Nutzung und Freigaberegeln ab. Eine erfolgreiche Prüfung zweier neuer Versionen deckt ältere Anwendungen nicht mit ab.

Tests sollten die entscheidenden Eigenschaften ausdrücken. Reicht eine formal gültige Kennung, prüfen Sie das Format anstelle eines festen Werts. Bestimmt ein Feld die Anzeige, sind der Typ und die verstandenen Werte wichtig. So schützt der Vertrag gemeinsames Verhalten und lässt Änderungen am inneren Aufbau zu.

Wie es in der Praxis aussieht

Ein ausgedachtes Beispiel ist eine Anwendung zur Paketverfolgung. Sie fragt ein Paket mit seiner Kennung ab. Existiert es und ist unterwegs, erwartet die Oberfläche eine erfolgreiche Antwort, die Kennung und einen Textwert für die passende Anzeige. Bei einem unbekannten Paket erwartet sie eine andere Antwort und schlägt vor, die Kennung zu prüfen. Zu jeder Interaktion gehört ein Ausgangszustand, etwa ein Paket im Transport oder kein Paket mit der angegebenen Kennung.

Lassen Sie den Verbraucher seine wirkliche Anfrage an den kontrollierten Ersatz senden. Prüfen Sie jeweils das angezeigte Ergebnis. Führen Sie dann dieselben Interaktionen auf dem wirklichen Antwortweg des Anbieters aus. Bereiten Sie reproduzierbare Daten für den jeweiligen Zustand vor und setzen Sie sie zwischen Versuchen zurück. Ersetzt der Test eine Datenbank oder einen Frachtanbieter, sollen die relevante Verarbeitung der Anfrage, die Dienstlogik und die Erzeugung der Antwort weiterhin laufen.

Ändern Sie zur Probe das Feld `status` in `state`, ohne den Verbraucher zu ändern. Die Verifikation sollte das fehlende Feld und die betroffene Interaktion melden. Ein zusätzliches optionales Feld kann zulässig sein, wenn es der Verbraucher ignoriert und der Vertrag es erlaubt. Ordnen Sie jede Prüfung ihren Versionen zu. Vermischte Ergebnisse aus neuem Dienst und unklarer Vertragsfassung erschweren die Aussage, welche Kombination überprüft wurde.

Unsere Notiz zum Öffnen einer Anwendung in mobilen Solana-Wallets zeigt Schnittstellen mit unterschiedlichen Linkpfaden und Codierungen von Parametern. Die Notiz beschreibt diese Unterschiede und Prüfungen auf Geräten. Eine Prüfung erzeugter Links könnte die erwartete Form je Wallet absichern. Ein Durchlauf mit einem echten Gerät kann zeigen, ob der Browser den Link an die installierte App übergibt und der Vorgang abgeschlossen wird. Das sind unterschiedliche Fragen.

Wo es schiefgeht

Verträge können zu starr werden, wenn jedes Byte einer Antwort einem gespeicherten Beispiel entsprechen soll. Ein Anbieter kann berechtigt sein, ein ungenutztes Feld hinzuzufügen oder eine Zusatzmeldung anders zu formatieren. Verlangen Sie genaue Werte, Reihenfolge oder Einheiten dort, wo Verbraucher oder Spezifikation sich darauf verlassen.

Ein Test verliert auch an Wert, wenn er die Grenze umgeht. Ein von Hand geschriebenes JSON-Objekt kann den Code zur Erzeugung der Antwort auslassen. Ein Ersatz kann die vorbereitete Probe zurückgeben, obwohl der Verbraucher sie im normalen Ablauf falsch liest. Führen Sie den echten Anfrage- oder Antwortcode der geprüften Seite aus und kennzeichnen Sie ersetzte Abhängigkeiten.

Ein gelungener Normalfall sagt zuletzt wenig über fehlende Daten, unbekannte Kennungen und ältere Versionen. Berücksichtigen Sie bekannte Fehlerwege und zugesicherte Kompatibilität. Behalten Sie Integrationstests für Routing und Konfiguration. Ein präziser Vertrag gibt beiden Teams eine gezielte, schnelle Prüfung für jede betroffene Änderung.

Weiterführende Lektüre

- Pact-Dokumentation zum Begriff Vertragstest
- Verifikation von Verbraucher und Anbieter mit Pact
- Dokumentation zu Anbieterzuständen in Pact
- OpenAPI-Spezifikation für HTTP-Schnittstellen

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Vertragstest?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/contract-testing/