Engineer

Kontraktstestning

testning av överenskomna gränssnitt mellan komponenter

Vad det är

Kontraktstestning undersöker om två komponenter är överens om sina förfrågningar, svar eller meddelanden. En skärm för paketspårning och dess tjänst kan fungera var för sig. Skärmen kan ändå fallera om den förväntar sig fältet `status` och tjänsten skickar `state`. Ett test beskriver vad skärmen faktiskt använder och verifierar att tjänsten kan leverera det.

Pact beskriver en metod där konsumenten körs mot en kontrollerad ersättare för leverantören. Testet registrerar utbyten i ett kontrakt. Leverantören kontrollerar sedan samma utbyten i sin faktiska implementation under angivna förutsättningar. För meddelanden gäller motsvarande bevis själva meddelandet som produceras och konsumeras. Båda sidor kan utvecklas och släppas oberoende om underlaget hör till de versioner som ska fungera tillsammans.

Begreppet används även för att verifiera en leverantör mot dess publicerade gränssnitt, exempelvis ett OpenAPI-dokument. Det är användbart när specifikationen är den gemensamma överenskommelsen. Den kan ange fält, statuskoder och tillåtna värden. Konsumentdrivna exempel visar dessutom vilka kombinationer en viss anropare använder. Metoderna går att kombinera. Ett försök genom den verkliga integrationen kan också pröva dirigering, åtkomst och miljöinställningar.

Använd begreppet i ett beslut

Använd ”Vad är kontraktstestning?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Den del användarna aldrig ser: där data lagras, vilka reglerna är och hur andra system får kommunicera med den.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Utformade och drev 6 ETL/ELT-pipelines — 6 ETL-pipelines över tre databasmotorer

Utformade och drev 6 ETL/ELT-pipelines Hör av er →

Varför det spelar roll

Ett gränssnitt kan leva längre än koden som införde det. Tjänstens team kan döpa om ett fält och uppdatera sina tester utan att röra klienten i ett annat projekt. En mobilapp kan fortsätta att köras i en äldre version under flera månader. Även en annan datatyp har betydelse när fältet väljer texten som syns. Ett verifierbart kontrakt synliggör de beroendena före leverans.

Separata leveranser gör det viktigt att känna till kompatibla versioner. En gammal klient förstår kanske några få statusvärden medan tjänsten behöver ett nytt mellansteg. Teamen kan förbereda konsumenten först, tillåta båda formerna och ta bort äldre beteende när ingen version som stöds använder det. Ordningen beror på kontrakten, den verkliga användningen och leveransreglerna. En lyckad provning av två nya versioner visar inget om klienter som ännu inte uppdaterats.

Kontrollerna ska gälla det som behöver vara sant. Om varje giltig referens fungerar, kontrollera dess form och undvik ett påtvingat exempelvärde. Om ett fält styr vad användaren ser, kontrollera typen och vilka värden klienten förstår. Bestäm också hur ett eventuellt okänt värde får visas. Ett exempel kan visa att en text som ”status saknas” lämnar resten av sidan användbar, om den lösningen passar uppgiften.

Så ser det ut i praktiken

Tänk dig en fiktiv app för paketspårning. Den begär ett paket med hjälp av sin referens. Finns paketet och är det på väg, väntar skärmen ett lyckat svar, rätt referens och ett textvärde som går att koppla till en etikett. Är paketet okänt, förväntar den sig ett annat svar och hjälper besökaren att kontrollera referensen. Varje utbyte har ett givet utgångsläge, till exempel att paketet är på väg eller att referensen inte finns.

Kör den verkliga frågekoden i konsumenten mot ersättaren och kontrollera det synliga resultatet. Spela sedan upp samma utbyten genom leverantörens riktiga svarsväg. Lägg in förutsägbara data för varje tillstånd och återställ mellan försöken. Om en databas eller en fraktleverantör ersätts, låt den relevanta tolkningen av frågan, programlogiken och skapandet av svaret köras.

Byt tillfälligt fältet `status` mot `state` utan att ändra klienten. Verifieringen ska ange det saknade fältet och identifiera utbytet. Ett nytt valfritt fält som konsumenten inte använder kan vara tillåtet. Koppla kontrollerna till de faktiska versionerna. Ett besked från en ny tjänst och ett oidentifierat kontrakt visar inte vilket par som prövades.

Vår anteckning om att öppna en app i mobila Solana-plånböcker beskriver integrationer som väntar olika adresser och kodning av parametrar. Anteckningen visar skillnaderna och föreslår provning med en enhet. Ett test av genererade länkar kan skydda deras form. En verklig enhet kan visa om webbläsaren lämnar över länken till installerad app och om flödet går att slutföra. Observationerna kompletterar varandra.

Var det går fel

Kontraktet kan bli för strikt om varje byte måste motsvara ett sparat exempel. Leverantören kan ha rätt att lägga till ett oanvänt fält eller byta formatering på ett fristående meddelande. Kräv en bestämd ordning, exakt text eller enhet när en klient använder det eller specifikationen fordrar det.

Ett test mister värde om det går förbi den verkliga gränsen. Ett handskrivet JSON-objekt kan lämna koden som skapar svaret oprövad. Ersättaren kan skicka rätt exempel trots att klienten inte ens tolkar det under normal användning. Kör den faktiska fråge- eller svarskoden för den sida som prövas och ange vilka beroenden som ersatts.

Det kan också vara värt att dokumentera varför ett visst beteende hör till avtalet. Ett växande antal klienter kan till exempel börja använda ett fält som först var valfritt. Att varje test visar vilken klient det kommer från gör det lättare för leverantören att upptäcka det nya beroendet.

Ett lyckat normalfall säger föga om saknade data, okända referenser eller äldre versioner. Leta upp de fel som klienten hanterar och de versioner som leverantören stöder. Behåll integrationstester som går genom dirigering och inställningar. Kontraktet ger båda teamen ett snabbare, riktat underlag för varje relevant ändring.

Läs mer

- Pacts dokumentation om vad kontraktstestning är
- Så verifierar Pact konsument och leverantör
- Pacts dokumentation om leverantörens tillstånd
- OpenAPI-specifikationen för HTTP-gränssnitt

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är kontraktstestning?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

welcome@engineer.company +45 50 10 61 84 Köpenhamn, Danmark

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/contract-testing/