Engineer

Kontraktstesting

testing av avtalte grensesnitt mellom komponenter

Hva det er

Kontraktstesting undersøker om to komponenter er enige om forespørslene, svarene eller meldingene de utveksler. En skjerm for pakkesporing og tjenesten bak skjermen kan virke hver for seg. Skjermen feiler likevel hvis den leter etter feltet `status` og tjenesten sender `state`. En test kan beskrive det skjermen bruker og kontrollere at tjenesten kan levere det.

Pact beskriver en tilnærming der konsumenten kjører mot en kontrollert erstatning for leverandøren. Testen registrerer samhandlingen i en kontrakt. Leverandøren kontrollerer den så gjennom sin egen implementasjon under avtalte forutsetninger. For meldinger handler tilsvarende bevis om meldingen som blir laget og lest. De to sidene kan arbeide og gi ut nye versjoner uavhengig hvis bevisene gjelder versjonene som skal brukes sammen.

Noen bruker samme begrep om kontroll av en leverandør mot den publiserte grensesnittbeskrivelsen, for eksempel et OpenAPI-dokument. Det er nyttig når spesifikasjonen er den felles avtalen. Den kan angi felter, statuskoder og tillatte verdier. Konsumentdrevne eksempler viser dessuten hvilke kombinasjoner en bestemt klient trenger. Begge metodene kan brukes samtidig. Et forsøk gjennom den ekte integrasjonen kan undersøke ruting, tilgang og oppsett i tillegg.

Bruk begrepet i en beslutning

Bruk «Hva er kontraktstesting?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Delen brukerne aldri ser: hvor dataene ligger, hvilke regler som gjelder, og hvordan andre systemer får lov til å snakke med den.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Designet og driftet 6 ETL/ELT-pipelines — 6 ETL-pipelines på tvers av tre databasemotorer

Designet og driftet 6 ETL/ELT-pipelines Ta kontakt →

Hvorfor det er viktig

Et grensesnitt lever ofte lenger enn koden som innførte det. Tjenestens team kan gi et felt nytt navn og endre egne tester uten å oppdatere klienten i et annet prosjekt. En mobilapp kan være i bruk i en eldre versjon i flere måneder. Selv en liten typeendring kan endre teksten som vises til brukeren. En kontraktstest gjør avhengigheten synlig før utgivelsen.

Uavhengige utgivelser øker behovet for kunnskap om kompatibilitet. En eldre klient kan forstå bare noen få statuser, mens en ny tjeneste trenger en mellomstatus. Teamene kan oppdatere klienten først, godta begge former gjennom overgangen og fjerne den gamle når ingen versjon som støttes, bruker den. En trygg rekkefølge avhenger av kontraktene, faktisk bruk og utgivelsesreglene. En prøve med de to nyeste utgavene sier ikke hva som skjer med eldre klienter.

Testene bør uttrykke de vesentlige punktene. Dersom enhver gyldig referanse fungerer, sjekk formatet uten å kreve én bestemt verdi. Hvis et felt bestemmer hva brukeren ser, kontroller typen og verdiene klienten håndterer. Eksempler kan også vise en avtalt reaksjon på en ukjent status. Hvis klienten kan vise «status ikke tilgjengelig» uten å miste resten av siden, bør testen vise hva det betyr. Beslutningen hører til produktet.

Slik ser det ut i praksis

Tenk deg en app som sporer pakker. Den spør etter en pakke med en referanse. Når pakken finnes og er underveis, forventer appen et svar som angir suksess, riktig referanse og en tekstverdi den kan omsette til en status. Når pakken er ukjent, forventer den et annet svar og gir brukeren en måte å undersøke referansen på. Hvert eksempel har en bestemt situasjon som må opprettes før forespørselen kjøres.

Bruk klientens virkelige forespørselskode mot erstatningen og kontroller det synlige resultatet. Kjør deretter den samme forespørselen gjennom leverandørens virkelige svarvei. Bruk kjente testdata for hver situasjon, og

nullstill dem mellom forsøk. Hvis testen erstatter databasen eller en ekstern transportør, skal parsing, tjenestens behandling og produksjon av svaret fortsatt kjøres.

Bytt feltet `status` til `state` midlertidig uten å endre klienten. Kontrollen bør gjenkjenne det manglende feltet og peke på den aktuelle utvekslingen. Et nytt felt som konsumenten ikke bruker, kan fortsatt være tillatt. Knytt kontrollen til de nøyaktige versjonene. Et resultat fra én tjeneste og en uidentifisert kopi av en gammel kontrakt viser ingen sikkert testet kombinasjon.

Vårt notat om å åpne programmer i mobile Solana-lommebøker beskriver integrasjoner med ulike lenkestier og koding av parametere. Notatet dokumenterer ulikhetene og tester man kan gjøre på en enhet. En test av genererte lenker kan beskytte formen hver lommebok forventer. Et forsøk på en virkelig enhet viser om nettleseren overlater lenken til appen og om brukerforløpet lykkes. Det er to forskjellige observasjoner.

Hvor det går galt

En kontrakt kan bli for streng hvis den krever samme bytefølge som ett lagret eksempel. Tjenesten kan få lov til å legge til et ubrukt felt eller endre formateringen av et frittstående svar. Krev bestemt rekkefølge, enhet eller ordlyd når klienten bruker det eller den offentlige spesifikasjonen krever det.

En test mister verdi når den går utenom det virkelige grensesnittet. En håndskrevet JSON-fil kan la koden som lager svaret, stå uprøvd. En erstatning kan returnere det avtalte eksempelet selv om klienten ikke tolker det i normal bruk. Kjør forespørsels- eller svarkoden for siden som testes, og noter hvilke avhengigheter som er erstattet.

For en eldre klient kan det være for sent å legge inn ny håndtering etter at tjenesten er rullet ut. Avklar hvilke klientversjoner som faktisk er i bruk, og behold prøver for dem så lenge avtalen skal støttes. Da kan teamene diskutere overgangen med samme bevisgrunnlag.

Ett vellykket forløp sier til sist lite om manglende data, ukjente referanser og eldre versjoner. Finn feilene klienten kan håndtere og versjonene leverandøren faktisk støtter. Behold integrasjonstester av ruting og oppsett. Kontrakten gir et avgrenset, raskere bevis som teamene kan bruke ved hver relevant endring.

Les mer

- Pact-dokumentasjonen om hva kontraktstesting er
- Slik kontrollerer Pact konsumenten og leverandøren
- Pact-dokumentasjonen om tilstander hos leverandøren
- OpenAPI-spesifikasjonen for HTTP-grensesnitt

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er kontraktstesting?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/contract-testing/