**Engineer** Programvare

# Bygde en testsuite for Go, SQL og nettleser

Bygde en lagdelt automatisert testsuite — 981 Go-tester, 543 spesifikasjoner for frontend og nettleser, 494 atferdstester i SQL — med mutasjonstesting, egenskapsbaserte tester og en kvalitetsport for universell utforming.

2025

## 981

Go-tester, ved siden av 543
nettleser- og 494 SQL-spesifikasjoner

**Situasjon.** En plattform som har forretningslogikken sin i databasen, har et testproblem de fleste prosjekter ikke har. Logikken ligger ikke i språket testrammeverket er godt på — den ligger i SQL, bak funksjonsgrenser, og SQL er nettopp den typen kode som forblir utestet fordi det er kløtete å teste den. Legg et Go-API og en Next.js-frontend oppå det, så blir «de viktige delene er dekket» stille og rolig til «de delene som var lette å dekke, er dekket».

**Oppgave.** Hvert lag måtte få oppførselen sin kontrollert der oppførselen faktisk hører hjemme, i stedet for at alt ble kontrollert utenfra gjennom en nettleser.

**Handling.** Fire lag fikk fire typer tester. Go-API-et har 981 testfunksjoner fordelt på 310 filer. Frontenden har 543 Vitest- og Playwright-spesifikasjoner, inkludert 48 ende-til-ende-filer og 30 nettleserspesifikasjoner. Databasen har 494 atferdstestfiler — 116 876 linjer SQL som kontrollerer resultater gjennom 6 654 utløste unntak — slik at en lagret funksjon testes i databasen i stedet for gjennom tre lag med applikasjon over den. Over disse ligger testene som tester testene: Stryker-mutasjonstesting ødelegger med vilje en linje og feiler når ingenting merker det, og fast-check genererer inndata ingen tenkte på å skrive ned. En kvalitetsport med axe-core krever null brudd på WCAG 2.0 og 2.1 nivå A og AA, noe som gjør brudd på

universell utforming til en byggfeil i stedet for et revisjonsfunn måneder senere. Tersklene for testdekning flyttes bare oppover. Hele testsuiten kjører som 10 jobber i en arbeidsflyt på 612 linjer.

**Resultat.** Å endre noe strukturelt sluttet å være skremmende, og det er det eneste som hindrer en kodebase av denne størrelsen i å forkalkes. Den ærlige kostnaden er tid — testsuiten er treg, den gjør hver endring mer tidkrevende, og i denne størrelsen trenger den eget vedlikehold. Det den kjøper, er evnen til å fortsette å bevege seg raskt, og det er verdt mer enn minuttene den tar.

### Bruk erfaringen i prosjektet deres

Bruk «Bygde en testsuite for Go, SQL og nettleser» som utgangspunkt for en lignende utfordring i organisasjonen deres. Tjenestene bak arbeidet er tilgjengelige for prosjektet deres. 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 →

### Se bak resultatet

Les om situasjonen, beslutningene og resultatet i eksemplet. Vi kan drøfte hva som passer deres rammer, og hva som krever en annen tilnærming. Bygde en testsuite for Go, SQL og nettleser — 981 Go-tester, ved siden av 543 nettleser- og 494 SQL-spesifikasjoner

**Se tjenestene våre** Ta kontakt →

## Vil du ha tester som faktisk fanger opp noe?

Snakk om prosjektet med ingeniøren som har ansvaret for leveransen. Vi avtaler omfang, prisgrunnlag og overlevering skriftlig før arbeidet begynner.

Rolle i oppdraget: Test- og kvalitetsingeniør. Det bygger på Backend-utvikling, Frontend-utvikling, PostgreSQL, DevOps, Automatisering og CI/CD, Pålitelighet og sikkerhetskopier og Testing og kvalitetssikring. Og nå som dere er her: vi er ganske gode på Backend- og API-utvikling, DevOps og CI/CD-automatisering og Teknisk ledelse og rådgivning.

**Ta kontakt** Se tjenestene våre

Ett stykke av **en lengre oversikt** — og den ligger i sin helhet her.

Hver oppføring er skrevet på samme måte: situasjonen, oppgaven, hva vi gjorde, og hva som endret seg.

**Se hele oversikten**

Definer forbedringen dere trenger

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 →

## Vil du ha tester som faktisk fanger opp noe?

Spør om beslutningene bak «Bygde en testsuite for Go, SQL og nettleser». Dere snakker med folk som kan forklare arbeidet og vurdere deres eget prosjekt.

**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://software.engineer.company/nb/portfolio/built-a-test-suite-across-go-sql-and-browser/**