Engineer Mjukvara

Byggde en testsvit för Go, SQL och webbläsare

Byggde en skiktad automatiserad testsvit — 981 Go-tester, 543 specifikationer för frontend och webbläsare, 494 beteendetester i SQL — med mutationstestning, egenskapsbaserade tester och en tillgänglighetsgrind.

2025

981

Go-tester, vid sidan av 543 webbläsar- och 494 SQL-specifikationer

Situation. En plattform som har sin affärslogik i databasen har ett testproblem som de flesta projekt slipper. Logiken finns inte i det språk som testramverket är bra på — den finns i SQL, bakom funktionsgränser, och SQL är precis den sortens kod som förblir otestad eftersom den är besvärlig att testa. Lägg till ett Go-API och en Next.js-frontend ovanpå det, så blir ”de viktiga delarna är täckta” i tysthet till ”de delar som var lätta att täcka är täckta”.

Uppgift. Varje lager behövde få sitt beteende verifierat där beteendet faktiskt finns, i stället för att allt verifierades utifrån genom en webbläsare.

Handling. Fyra lager fick fyra sorters tester. Go-API:t har 981 testfunktioner fördelade på 310 filer. Frontenden har 543 testspecifikationer i Vitest och Playwright, däribland 48 end-to-end-filer och 30 webbläsarspecifikationer. Databasen har 494 beteendetestfiler — 116 876 rader SQL som verifierar genom 6 654 kastade undantag — så att en lagrad funktion testas i databasen i stället för genom tre lager applikation ovanför. Ovanpå dem ligger testerna som testar testerna: mutationstestning med Stryker förstör medvetet en rad och fallerar när ingenting märker det, och fast-check genererar indata som ingen tänkte på att skriva ned. En axe-core-grind kräver noll överträdelser av WCAG 2.0 och 2.1 på nivå A och AA, vilket gör tillgänglighet till ett byggfel i stället för ett granskningsfynd månader senare. Tröskelvärdena för testtäckning flyttas bara uppåt. Hela sviten körs som 10 jobb i ett arbetsflöde på 612 rader.

Resultat. Att ändra något strukturellt slutade vara skrämmande, och det är det enda som hindrar en kodbas av den här storleken från att förkalkas. Den ärliga kostnaden är tid — sviten är långsam, den belastar varje ändring, och i den här storleken behöver den eget underhåll. Det man får för den är förmågan att fortsätta i högt tempo, och det är värt mer än de minuter den tar.

Använd erfarenheten i ert projekt

Använd ”Byggde en testsvit för Go, SQL och webbläsare” som utgångspunkt för en liknande utmaning i er organisation. Tjänsterna bakom arbetet finns tillgängliga för ert projekt. 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 →

Se bakom resultatet

Läs om situationen, besluten och resultatet i exemplet. Vi kan diskutera vad som passar era förutsättningar och vad som kräver ett annat angreppssätt. Byggde en testsvit för Go, SQL och webbläsare — 981 Go-tester, vid sidan av 543 webbläsar- och 494 SQL-specifikationer

Se våra tjänster Hör av er →

Vill du ha tester som faktiskt hittar fel?

Prata om ert projekt med ingenjören som ansvarar för leveransen. Vi avtalar omfattning, hur priset beräknas och överlämning skriftligt innan arbetet börjar.

Roll i uppdraget: Test- och kvalitetsingenjör. Det bygger på Backendutveckling, Frontendutveckling, PostgreSQL, DevOps, Automatisering och CI/CD, Tillförlitlighet och säkerhetskopiering och Test och kvalitetssäkring. Och nu när ni är här: vi är ganska bra på Backend- och API-utveckling, DevOps och CI/CD-automatisering och Tekniskt ledarskap och rådgivning.

Hör av er Se våra tjänster

En del av **en längre historik** — och den finns i sin helhet här.

Varje post är skriven på samma sätt: situationen, uppgiften, vad vi gjorde och vad som förändrades.

Se hela historiken

Definiera förbättringen ni behöver

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 →

Vill du ha tester som faktiskt hittar fel?

Fråga om besluten bakom ”Byggde en testsvit för Go, SQL och webbläsare”. Ni talar med personer som kan förklara arbetet och bedöma ert eget projekt.

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