**Inlämningsuppgift**

Oliver Werthén

2021-09-0

Kurs: Testautomation med CI, 40 yhp

Kursdatum: 2022-01-17 – 2022-03-13

Utbildare: Martin Jansson, Inceptive

Utbildning: Mjukvarutestare 325 yhp, 2021-2023

**Valet av testverktyg**

Jag har valt att skapa mina tester i Selenium IDE och Webdriver. Valet av testverktyg var tämligen simpelt för mig då jag var väldigt inriktad på att testa en webbsidas GUI. Selenium är det GUI verktyget vi gick igenom under föreläsningar som jag även har hört talas om i flera sammanhang genom gästföreläsare och mentor. Det verkar även som det används av en hel del testare och företag när det gäller automatiserad testning.

Selenium kändes väldigt kraftfullt eftersom man kunde spela in tester, man kan snabbt skapa tester med endast några klick och koden skrivs åt en. När man spelat in testerna kan man enkelt spela upp dem igen för att se resultatet.

Det kändes även smart att använda Selenium då jag kunde använda C#, som jag lärde mig i gymnasiet för att fräscha upp de kunskaperna igen. Selenium verkade även som ett bra val då det stödjer flera olika webbläsare och programmeringsspråk, jämfört med exempelvis Cypress som bara stödjer Javascript och inte stödjer webbläsare såsom Safari och Edge.

**Testdesign**

I de flesta av mina tester började jag med att skapa grunden i Selenium IDE genom att spela in navigationsvägar i den mån det gick för varje test. Därefter exporterade jag testet och fortsatt att bygga ut testet med hjälp av Webdriver och kompletterade med ”waits” osv där det krävdes.

Jag hade velat skapa lite tester för att publicera bostäder och med fel värden osv men det kändes lite dumt på en ”verklig sida” där jag inte har direkt tillgång till databasen så det kan tas bort direkt.

Nu i efterhand när jag kommer ihåg lite mer av C# och vet lite mer om hur man skapar de automatiserade testerna. Så hade det varit bättre att göra om visa koddelar till klasser så de lätt kunde återanvändas på ställen där det behövdes.

**Setup**: I min setup så startar jag Chrome ”drivern” samt öppnar startsidan för Norban.se. Efter det har jag en lagt in en ”wait” som ser till att det har laddats in korrekt och sedan sätter webbläsarfönstret till 1552x832 pixlar.

**Initialize**: I initialize så öppnar startsidan för Norban.se, ser till så att den är inladdad med en ”wait” och sätter fönstret till 1552x832 pixlar.

**Teardown**: I min teardown stänger jag ner drivern.

**Cleanup**: I min cleanup rensar jag alla cookies.

**verifyHomeButton**: I detta test verifierar jag att hemknappen fungerar på webbsidan både genom att klicka på den på förstasidan samt från undersidan Så funkar Norban. Testet verifierar också att undersidan Så funkar Norban är länkad rätt.

**verifyNyproduktion**: Här verifierar jag att länken till undersidans nyproduktion fungerar.

**logIn**: logIn testet är designat för att testa så en befintlig användare kan logga in med hjälp av sin mejl och sitt lösenord.

**logInNopassword**: I detta test testas det att det inte går att logga in utan att skriva in lösenord samt att en text med ”lösenord saknas” dyker upp.

**loginNousername**: Här testas det så att det inte går att logga in utan att skriva in mejl samt att en text med ”e-post saknas” dyker upp.

**loginNoUserandPass**: I detta test körs båda ovanstående samtidigt och testar att både ”e-post saknas” och ”lösenord saknas” kommer upp.

**phonemeny**: Här sätts fönsterstorleken till lite mindre för att få fram ”mobilmenyn” och testar att alla undersidor länkas rätt.

**getAdressAndSearch**: Testet är gjort för att testa att det går att klicka fram senaste publicerade bostaden samt att det går att använda sökfältet för att hitta den specifika bostaden.

Oliver Werthén

2022-02-25