

# Lab 1 – Testautomatisering

---

Deadline: 2013-02-25 19:59

Lab 1 utförs i par om två stycken elever.

Inlämning sker med en zippad fil "Lab1.zip" mailad till [david.g@jetas.se](mailto:david.g@jetas.se).

Zip-filen skall innehålla samtliga .rb-filer ni skapat för denna lab. Utöver ruby-filerna skall zip-filen även innehålla en textfil "info.txt". "info.txt" skall innehålla följande:

- Fullständigt Namn + Födelsenr. för båda eleverna.
- Skriv vilka uppgifter ni utfört. Ex: "Utfört Uppg: 1,3,4,5,6,8".
- Skriv svar på de frågor där det anges i uppgiften att ni skall svara i "info.txt". Ex: "Fråga 2: [Svar här]".

Vilken browser ni väljer att skriva testerna för spelar ingen roll.

Om ni upplever att någon av uppgiftsbeskrivningarna är otydliga – hör av er direkt till mig. Jag kommer göra en nyhetspostning på kurshemsidan ifall jag förtydligar någon uppgiftsbeskrivning (Ni kan prenumerera på kurshemsidan f.ö.).

När jag talar om att skriva ut olika resultat så syftar jag på ruby-funktionerna "puts", "print", etc.

Jag är intresserad av att höra ifall ni känner att ni kan ha användning för det ni lär er under denna lab för att skriva automatiserade tester för en verklig produkt i framtiden – eller om ni känner att det är för långt ifrån verkligheten för att vara nyttigt. Om ni har några tankar kring detta får ni gärna maila mig om det.

## Uppgifter på nivå godkänt (Krav, G - 13/25p)

Skapa en ny ruby-fil, "grund.rb". I Filen skall ni skriva ruby-kod för att lösa följande uppgift 1-9 med Watir-Webdriver. Tänk på att ni kan återanvända ett och samma browserobjekt för samtliga test i uppgift 1-9. För dessa uppgifter har ni förmodligen stor hjälp av exemplet på:  
<http://watir.com/examples/>

1. Vilken browser är dina test skrivna för? Ge ett exempel på hur man kan ändra scriptet för att använda en annan browser. (Skriv dina svar i "info.txt") (2p)
2. För webbadressen [bit.ly/watir-webdriver-demo](http://bit.ly/watir-webdriver-demo) – Skriv ruby-kod som använder Watir-WebDriver för att klicka på submit-knappen. Verifiera därefter att valideringstext som säger att vi inte fyllt i alla fält visas. (3p)
3. Vad händer om ni försöker klicka på submit-knappen men försöker välja knappen med hjälp av "id => 'submit'" istället för det sätt som ni identifierade den med i fråga 2? Varför? (Skriv svaret i "info.txt") (2p)

4. För webbadressen `bit.ly/watir-webdriver-demo` – Skriv ruby-kod som använder Watir-WebDriver för att Verifiera att det finns 5 st. element att välja i dropdown-listan på sidan. (Om ni hittar `select_list`-elementet på sidan kan ni använda `".options"` för att få ut en array med dess child-element. Arrayens längd kan ni då få med t.ex. `".count"`) (3p)
5. För webbadressen `bit.ly/watir-webdriver-demo` – Skriv ruby-kod som använder Watir-WebDriver för att klicka på länken "Google Docs", och vänta därefter tills ni kan verifiera att ni är på google-sidan. (3p)
6. Skriv ruby-kod som använder Watir-WebDriver för att spara screenshot av sidan där ni befinner dig efter Uppgift 5.

Mer info: <http://watirwebdriver.com/screenshots/> (Vad ni sätter för namn på denna screenshot spelar ingen roll).

Om ni vill kontrollera att ni faktiskt tar ett screenshot, kan ni använda t.ex. 'FileManager' (I Operativsystemet: Klicka på menyn längst upp till vänster, välj 'Accessories' -> 'File Manager'). Om ni lyckas ta screenshot så kommer det lägga sig i katalogen där ni kör scriptet eller irb irån.

Denna funktion kan vi använda för att t.ex. automatiskt ta screenshot ifall något test inte passarar. Man skulle kunna bygga vidare på detta och t.ex. automatiskt skicka mail senare)(2p)

7. För webbadressen `bit.ly/watir-webdriver-demo` – Skriv ruby-kod som använder Watir-WebDriver för att göra följande:
  - a. Skriv in "Test" i namn-fältet
  - b. Välj "Ruby" i select-listan för språk
  - c. Välj "A programming language" som definition av vad ruby är
  - d. Välj alla 3 valen för vilken version av ruby som stöds av watir., vänta tills submit-knappen försvinner och verifiera att ni tagit er vidare till nästa sida. (4p)
8. För webbadressen `http://watir.com/`: Skriv ruby-kod som använder Watir-WebDriver för att klicka på länken "Community" i huvudmenyn. Vänta på att den nya sidan laddats på lämpligt sätt. Verifiera sedan att det finns en länk till google-gruppen "Watir General" på sidan ni hamnat på. (3p)
9. För uppgift 2,4,5,6,7,8 skall ni skriva ut ett resultat mha av ruby. Det skall tydligt framgå vilket test som utförs och vad resultatet är. Exempel på hur detta ser ut för mig:

```
File Edit View Terminal Go Help
testaut@testaut-VirtualBox:~$ ruby grund.rb
Test 2: OK!
Test 4: OK!
Test 5: OK!
Test 6: OK!
Test 7: OK!
Test 8: OK!

6/6 Tests Passed.
testaut@testaut-VirtualBox:~$
```

(3p)

## Uppgifter på nivå väl godkänd (Krav, VG - 10/18p på VG-delen + 13/25p på G-delen)

### Uppgift 10- (6p)

Skapa en ny fil, "twitter.rb". I filen skall ni skriva ruby-kod för att lösa uppgift 10 med Watir-webdriver:

OBS! Om ni har ett privat twitter-konto som ni använder i dagsläget så rekommenderar jag er att INTE använda det kontot för denna uppgift. Läs mer under paragrafen om "throttle"-funktioner nedan.

1. Skriv en ruby-funktion "login" som tar ett username och ett password som argument. Funktionen skall mha Watir-WebDriver logga in på twitter.com med det angivna användarnamnet och lösenordet.
2. Skriv ytterligare en funktion "testMall" som använder login-funktionen ovan för att logga in på twitter.
3. Lista ut något sätt att verifiera ifall den angivna användaren är inloggad eller ej och fyll på funktionen "testMall" med denna verifiering.
4. Vi skall nu kopiera och modifiera testet ni skrev i punkt 2-3 för att skapa 4 funktioner som verifierar följande:
  - a. Det skall inte gå att logga in utan att ange lösenord
  - b. En befintlig användare skall kunna logga in om användarnamn och lösenord anges
  - c. En befintlig användare skall inte kunna logga in om användarnamn och fel lösenord anges
  - d. En icke befintlig användare skall inte kunna logga in

Ni behöver inte skriva med användarnamn + lösenord för den användare ni använder i detta test. Jag kommer testa med en annan, befintlig användare. Ni kan skapa en slask-användare på twitter för att få en befintlig användare.

I slutändan behöver ni inte ha kvar funktionen "testMall". Ni skall ha en hjälp-funktion för login och 4 funktioner för deltesten under punkt 4 ovan. Skriv ut resultatet för varje deltest som "OK" eller "NOT OK"

OBS! Twitter har en throttle-funktion som innebär att man är tvungen att fylla i en captcha vid misslyckade inloggningar. Jag hoppas att de inte har ytterligare någon form av throttle-funktion som slås på ifall vi hamrar deras hemsida för hårt (max X inloggningsförsök per tidsperiod Y per ursprung W eller liknande), alternativt att de har en lockout-funktion (kontot fryses eller låses efter Z misslyckade inloggningsförsök.) Om ni stöter på problem med detta – hör av er, så får ni någon ersättningsuppgift.

Twitter tillåter bara 16 tecken för användarnamnet när man registrerar sig, men ni kan skriva in fler tecken när ni loggar in - så en användare med 17+ tecken i användarnamnet är en garanterad icke-befintlig användare.

Kommentar: Om mycket funktionalitet ligger dold bakom en login, så är det en god idé att skapa en login-funktion som vi kan återanvända i senare test. Det samma gäller för övriga moment vi återupprepar många gånger.

I ett verkligt scenario hade vi kunnat utöka detta test till att kontrollera att man inte kan registrera en användare med otillåtna tecken i användarnamnet, etc.

I ett verkligt scenario hade vi kunnat skapa den befintliga användaren i något tidigare steg av testen. (Det kan dock vara en dålig idé att skapa test som är beroende av varandra – vi kommer beröra detta på fredag – fundera kring varför tillsvidare).

Fundera kring: Hur hade ett rubyscript som tar användarnamn och lösenord som argument sett ut, ex: "ruby login.rb username password"

### **Uppgift 11- (6p)**

Skapa en ny fil, "google.rb". I filen skall ni skriva ruby-kod för att lösa uppgift 12 med Watir-webdriver

Skriv en sökterm på <http://www.google.se> och genomför en sökning (vad ni söker på spelar ingen roll – sålänge ni söker på något som är giltigt att söka på).

- verifiera att samma text står i sökrutan efter sökningen är gjord
- Skriv ut resultatet för testet, hur många sökresultat ni fick, samt hur lång tid sökningen tog.

Exempel för hur resultatet rapporteras då vi har samma text i sökrutan efter sökning, ungefär 230 000 sökresultat och sökningen tog 0.13 sekunder:

"OK - Results: 230 000 - Time: 0.13 seconds"

Exempel för hur resultatet rapporteras då vi inte har samma text i sökrutan efter sökning:

"NOT OK"

### **Uppgift 12- (3p)**

Hittills har vi alltid startat en webbrowser som har ett grafiskt gränssnitt (IE, ff, chrome, opera, etc). Detta tar onödigtid och vi är i detta fall bara intresserad av output från vårt script - inte vad som faktiskt händer i browserns gränssnitt. (3p)

Skriv om det föregående testet ifrån uppgift 11 så att en grafisk browser inte startas då testet körs (Tips: Att använda WebDriver på det här sättet kallas att testa "HEADLESS")

I ett verkligt scenario så körs ofta den här typen av script på en server som saknar skärm - ytterligare en anledning att inte starta upp en grafisk browser.

### *Uppgift 13- (3p)*

Skapa en ny fil, "complete\_test.rb". I filen skall ni skriva ruby-kod för att

Skriv ett script (ruby, bash, eller powershell - fritt val) som kör samtliga andra ruby script för Uppgift 1-12 (Lägg övriga ruby scripts i samma katalog som scriptet för uppgift 13 och räkna med att dessa filer alltid ligger i samma katalog). Skriv ut någon form av rubrik för varje annat script som körs. Det är ok om ni lägger koden från samtliga script i samma fil och använder ex. objektorienterad programmering för att skilja koden från olika script från varandra (3p)

I ett mer verklighetsnära scenario hade vi t.ex. kunnat ha underkataloger för varje grupp av test. Huvudscriptet går sedan in i varje underkatalog och exekverar samtliga test för den kategorin av test

I ett mer verklighetsnära scenario hade vi också kunnat lägga till flera scripts som bara kör vissa kategorier av test - ex. hade vi kunnat gruppera ett antal scripts som testar login-funktionalitet mer grundligt