



Automatisiertes Testen von Webapplikationen am Beispiel von Primo

Noémie Ammann & Sven Koesling

Deutscher Bibliothekartag 2018




Suche im Suchportal


Bücher, Zeitschriften und mehr ▾





ERWEITERTE SUCHE


Suche in:


 NEBIS-Bibliothekskatalog


 Research Collection

 ETH Data Archive

 E-Periodica


 E-Pics Bildkataloge


 Primo Central Index

 Hochschularchiv der ETH Zürich

 Max Frisch-Archiv Online

 Thomas-Mann-Archiv Online

 Graphische Sammlung Online

 Material-Archiv

Dienstleistungen und Ressourcen


Finden Sie alle Informationen auf der [Website der ETH-Bibliothek](#)

Kontakt

info@library.ethz.ch

+41 44 632 21 35

Aktuell

 [Neues Suchportal für Bibliotheksressourcen](#) >
Recherchetool mit verbesserten Funktionalitäten

[Alle Meldungen](#) >

Feature / Anforderungsformulierung

Wenn ich das Suchportal aufrufe,
Und ich in den Suchschlitz den Suchbegriff «Ajax» eingebe,
Und die Anzahl der Treffer nehme,
Und dann das Test-suchportal aufrufe,
Wenn ich in den Suchschlitz den Suchbegriff «Ajax» eingebe,
Und dort die Anzahl der Treffer nehme,
Dann sollten die Treffermengen ähnlich, d.h. die Abweichung unter 1%, sein.

Test mit zwei Suchbegriffen

```
# language: de
Funktionalität: Suche
```

```
Grundlage: # features/bibtag.feature:3
  Gegeben sei , dass die Startseite aufgerufen ist # features/step_definitions/suche_steps.rb:1
```

```
Szenariogrundriss: Die Suche ergibt auf Prod- und Testsystem eine ähnliche Anzahl Treffer # features/bibtag.feature:6
  Wenn ich die Seite "https://terza-test-co01-fe41.ethz.ch/primo-explore/search?vid=DADS&sortby=rank&lang=de_DE" aufrufe, # features/bibtag.feature:8
  Und ich in den Suchschlitz den Suchbegriff "<Suchbegriff>" eingebe, # features/bibtag.feature:9
  Und die Anzahl der Treffer nehme # features/bibtag.feature:10
  Und dann die Seite "https://terza-prod-co01-fe41.ethz.ch/primo-explore/search?vid=DADS&sortby=rank&lang=de_DE" aufrufe, # features/bibtag.feature:11
  Wenn ich in den Suchschlitz den Suchbegriff "<Suchbegriff>" eingebe, # features/bibtag.feature:12
  Und dort die Anzahl der Treffer nehme # features/bibtag.feature:13
  Dann sollten die Treffermengen ähnlich, d.h. die Abweichung unter 1%, sein. # features/bibtag.feature:14
```

Beispiele:

```
| Suchbegriff |
| museumplus_gs | *****
| Wald | *****
```

```
expected: < 1
```

```
got: 1.240972434136914 (RSpec::Expectations::ExpectationNotMetError)
```

```
./features/step_definitions/suche_steps.rb:110:in `/^sollten die Treffermengen ähnlich, d\.h\. die Abweichung unter 1%, sein\.$/'
```

```
features/bibtag.feature:19:in `Dann sollten die Treffermengen ähnlich, d.h. die Abweichung unter 1%, sein.'
```

```
features/bibtag.feature:14:in `Dann sollten die Treffermengen ähnlich, d.h. die Abweichung unter 1%, sein.'
```

Failing Scenarios:

```
cucumber features/bibtag.feature:19 # Szenariogrundriss: Die Suche ergibt auf Prod- und Testsystem eine ähnliche Anzahl Treffer, Beispiele (#2)
```

```
2 scenarios (1 failed, 1 passed)
```

```
16 steps (1 failed, 15 passed)
```

```
0m16.847s
```


Und das machen die Informatiker daraus:

```
Wenn(/^dort die Anzahl der Treffer nehme$/) do
  inhalt = page.find(".results-count").text
  resultat = /^\d+(\.\d{3})*/.match(inhalt)
  @treffer2 = resultat.to_s.gsub(/[^0-9]/, "")
end
```

```
Dann(/^sollten die Treffermengen ähnlich, d\.h\. die Abweichung unter 1%, sein\.$/) do
  if @treffer1.to_i > @treffer2.to_i
    proz = (@treffer1.to_f - @treffer2.to_f) * 100 / @treffer1.to_f
  else
    proz = (@treffer2.to_f - @treffer1.to_f) * 100 / @treffer2.to_f
  end
  puts "*****"
  puts "#{@treffer1} <> #{@treffer2} => #{proz}%"
  puts "*****"
  expect(proz).to be < 1
end
```

Praktisch: Ausgabe als Webseite

- Tests lassen sich automatisieren.
- Auch das Durchführen lässt sich automatisieren.
- Die Ausgabe kann bequem als Webseite angezeigt werden.

Funktionalität: Suche

Grundlage

Gegeben sei, dass die Startseite aufgerufen ist features/step_definition

Szenariogrundriss: Die Suche ergibt auf Prod- und Testsystem eine ähnliche Anzahl Treffer feature

Wenn ich die Seite "https://search-test.library.ethz.ch/primo-explore/search?vid=DADS&lang=de_DE&sortby=rank" aufrufe, feature

Und ich in den Suchschlitz den Suchbegriff "<Suchbegriff>" eingebe, feature

Und die Anzahl der Treffer nehme feature

Und dann die Seite "https://search.library.ethz.ch/primo-explore/search?vid=DADS&lang=de_DE&sortby=rank" aufrufe, feature

Wenn ich in den Suchschlitz den Suchbegriff "<Suchbegriff>" eingebe, feature

Und dort die Anzahl der Treffer nehme feature

Dann sollten die Treffermengen ähnlich, d.h. die Abweichung unter 1% sein. feature

Beispiele

Suchbegriff	
museumplus_gs	*****, 23481 <> 23476 => 0.0212938120182275%, *****
Wald	*****, 19418 <> 19662 => 1.240972434136914%, *****

```

expected: < 1
got: 1.240972434136914
./features/step_definitions/suche_steps.rb:110:in `/'sollten die Treffermengen ähnlich, d.\h.\ die Abweichung unter 1%, sein)./
features/bibtag.feature:19:in `Dann sollten die Treffermengen ähnlich, d.h. die Abweichung unter 1%, sein.'
features/bibtag.feature:14:in `Dann sollten die Treffermengen ähnlich, d.h. die Abweichung unter 1%, sein.'

```

Szenariogrundriss: In der Suche ohne Umlaut dürfen keine Treffer mit Umlaut gefunden werden feature

Wenn ich in den Suchschlitz den Suchbegriff "<Suchbegriff>" eingebe, feature

Dann dürfen in der Ergebnisliste keine Treffer mit der Markierung "<NichtErgebnis>" auftauchen. feature

Beispiele

Suchbegriff	NichtErgebnis
Flügel	Flügel
Veränderung	Veränderung

Szenario: Slider bietet eine plausible Zeitangabe zur Einschränkung der Trefferliste feature

Wenn ich in den Suchschlitz den Suchbegriff "cucumber" eingebe, features/step_definitions/vol

Dann sollte der Slider plausible Anfangs- und Endwerte haben. features/step_definitions/s

**** alt: 1922--> DS-Startdatum: 1920



Vielen Dank für Ihre Aufmerksamkeit

Weiterführende Informationen im ABI Technik vom Januar 2017 im Artikel:
«**automated testing of web applications**»

DOI: [10.1515/abitech-2017-0025](https://doi.org/10.1515/abitech-2017-0025)