

WEB RESEARCH PROOF OF CONCEPT: UI TESTER

Sam De Wachter

3 WEB-UX KdG

1 Inleiding

Na enkele dagen en zelfs een paar weken brainstormen voor een proof of concept idee had ik nog steeds niks. Ik was ten einde raad en vroeg daarom hulp aan Sam Serrien zodat hij mij op weg kon helpen. Na een korte brainstorm sessie kwam Sam met het idee om een front-end/UI tester te maken. Het was een zeer ambitieus proof of concept idee en ik had geen idee hoe ik er aan moest beginnen. Dat was dus het perfecte proof of concept voor web research!

In dit document zult u het volgende vinden:

1. Onderzoek naar wat front-end/UI testing juist is en welke programma's er al voor zijn
2. Uitleg over het UI testing programma dat ik heb gemaakt
3. Conclusie
4. Websites die ik heb gebruikt voor mijn research

2 Research

2.1 Wat is front-end/UI testing juist?

Front-end/UI testing wordt gebruikt om uiteraard de front-end/user interface van een website te testen op fouten. Als de test is afgerond en alles is vlot verlopen zal het programma zeggen dat alle testvoorwaarden positief zijn en dat er geen fouten zijn. Wanneer er iets mis gaat tijdens het testen zegt het programma waar en wat er juist is fout gelopen tijdens de test. Zo kan je gemakkelijk zien waar de error zich juist bevindt zodat je de fout gemakkelijk kunt oplossen.

2.2 Welke programma's bestaan er voor front-end/UI testing?

Na het web redelijk lang af te schuimen heb ik de volgende programma's UI testing programma's gevonden:

- Selenium (Python, Java, C#, Ruby, Javascript(node))
- Sahi (Java & Javascript)
- iMacros (Javascript)
- Watir (Ruby)
- Capybara (Ruby)
- Cucumber (Ruby)

Dit zijn nog maar enkele programma's die ik heb opgesomd. Er zijn nog tal van andere UI testing programma's.

2.3 Welk programma ga ik gebruiken?

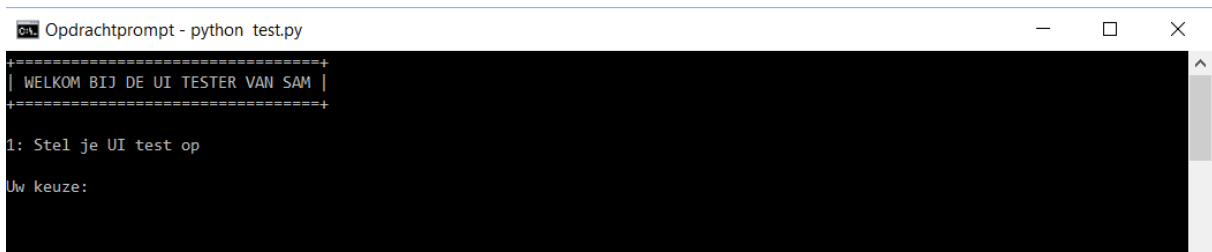
Voor mijn proof of concept heb ik gekozen voor Selenium omdat Selenium wellicht het bekendste web browser automation programma is op deze moment. Dat zorgt ervoor dat ik gemakkelijk dingen online kan vinden als ik vragen heb. Ik ga Selenium gebruiken in Python. Het enige wat ik moet doen om Selenium te installeren is 'pip install selenium' typen in de CLI van Windows en vanboven in mijn Python script 'from selenium import webdriver' typen.

3 Mijn programma

Zoals ik hiervoor al heb gemeld is mijn programma gemaakt in Python samen met het Selenium package. Wat nu volgt is een korte samenvatting van hoe mijn programma juist werkt.

3.1 Werking

Eerst komt u op het hoofdmenu terecht van het programma. Het programma zal je vragen of je je UI test wilt opstellen. Op dit menu zou het nog mogelijk zijn om bijvoorbeeld een extra link naar een documentatie te zetten, maar sinds ik hier al in grote lijnen uitleg hoe alles werkt heb ik dat niet gedaan.

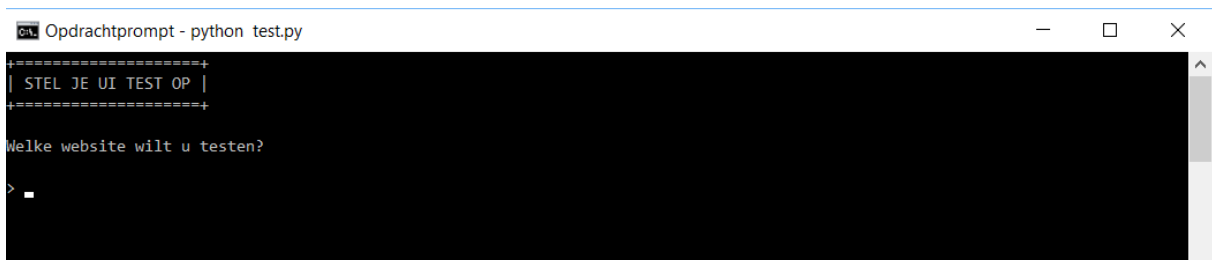


```
Opdrachtprompt - python test.py
=====+
| WELKOM BIJ DE UI TESTER VAN SAM |
=====+

1: Stel je UI test op

Uw keuze:
```

Eens je 1 hebt getypt en enter hebt gedrukt zal u op het volgende scherm terecht komen. Hier wordt er gevraagd welke website u graag zou willen testen.

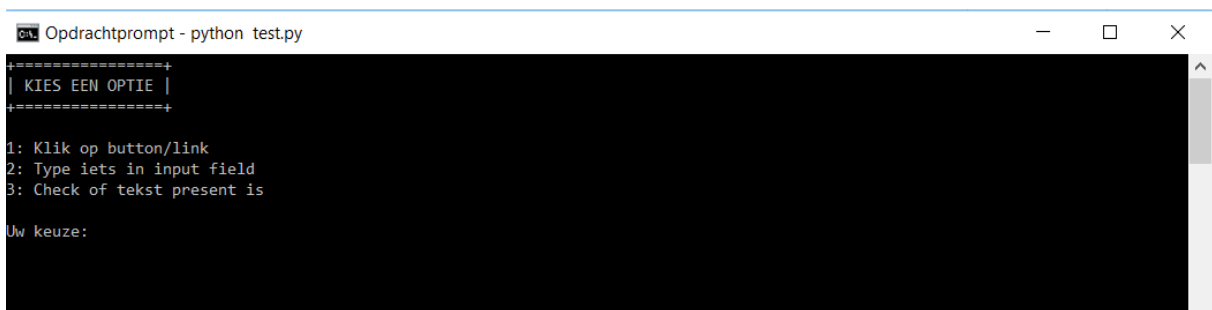


```
Opdrachtprompt - python test.py
=====+
| STEL JE UI TEST OP |
=====+

Welke website wilt u testen?

> -
```

Als je de website gekozen hebt is het volgende dat je moet doen de test cases instellen. In dit veld kan je kiezen tussen 3 mogelijkheden. De eerste is klikken op een button of een link, de tweede is iets intypen in een input field en de derde mogelijkheid is checken of een stuk tekst present is.



```
Opdrachtprompt - python test.py
=====+
| KIES EEN OPTIE |
=====+

1: Klik op button/link
2: Type iets in input field
3: Check of tekst present is

Uw keuze:
```

Als u bijvoorbeeld voor het input field kiest zal er gevraagd worden of u naar het input field wilt zoeken via class name of ID. Als je 1 van de twee hebt gekozen zult u de naam van de klasse of ID moeten ingeven en de tekst die u in het input veld wilt typen.

```
Opdrachtprompt - python test.py
=====
| VIA WAT WILT U HET INPUT VELD ZOEKEN |
=====
1: Class name
2: ID
Uw keuze:
```

Nadat je een optie hebt gekozen en alles heb ingevuld kom je op een volgend scherm terecht waar gevraagd zal worden wat de volgende stap is die u wilt doen. U kan dan kiezen of u nog een test case wilt toevoegen, de UI test wilt beginnen of dat u de ingevoerde data wilt aanpassen.

```
Opdrachtprompt - python test.py
=====
| WAT WILT U NU DOEN |
=====
1: nog een test toevoegen
2: begin de test
3: ingevoerde data aanpassen
Uw keuze:
```

Als u merkt dat u een fout hebt getypt kiest u gewoon optie drie en krijgt u een overzicht van alle opties die u hebt ingegeven. Kies dan de test case die u wilt aanpassen, kies de juiste waarde die je wilt veranderen en typ de nieuwe waarde in. Als je voor optie twee kiest kan u gemakkelijk de website die je wilt testen veranderen. Eens u klaar bent kiest u gewoon voor de optie 'Ga terug'.

```
Opdrachtprompt - python test.py
=====
| WELKE OPTIE WILT U AANPASSEN |
=====
1: Ga terug
2: Verander website
3: searchInput|id|s|Firth
4: click|id|searchsubmit
5: text|tag|body|Firth
Uw keuze: _
```

Als alles in orde is kan je de test van start laten gaan. Als de test volledig is afgerond krijgt u een scherm te zien met het overzicht van alle test cases. Daar zal al staan of ze gelukt of mislukt zijn. Onder het overzicht krijgt u nog een aantal opties. Je kan de test opnieuw uitvoeren, de data van de test aanpassen, een nieuwe test maken, de resultaten naar een xls bestand exporteren, de test exporteren naar een .py bestand of het programma beëindigen. Hieronder volgt een beknopte uitleg van wat die opties allemaal juist doen.

1. Voer dezelfde test opnieuw uit
 - a. Deze optie zal de test opnieuw uitvoeren met juist dezelfde waarden

2. Pas de data aan van deze test
 - a. Met deze optie kan je de waarde van de test die je juist hebt uitgevoerd aanpassen. Nadat je de gegevens hebt aangepast kan je de test opnieuw uitvoeren
3. Maak een nieuwe test (huidige test wordt verwijderd)
 - a. Deze optie zorgt ervoor dat je een gloednieuwe UI test kan maken. De vorige test die je hebt uitgevoerd zal wel overschreven worden.
4. Zet de resultaten om naar een xls bestand
 - a. Deze functie zal de resultaten die je in het overzicht ziet omzetten naar een overzichtelijk xls bestand. In dit xls bestand staan de waarde die je hebt ingegeven en welke functie er juist is gebruikt.
5. Exporteer test naar .py bestand
 - a. Deze optie zal ervoor zorgen dat de test cases die je hebt ingegeven worden geëxporteerd naar een kleiner .py bestand. Als je dat klein bestandje runned zal er enkel opkomen of de cases gelukt of mislukt zijn.
6. Stop het programma
 - a. En last but not least, deze optie zorgt ervoor dat het programma zal worden afgesloten.

```
Opdrachtprompt - python test.py

=====+
| OVERZICHT VAN DE UI TEST |
=====+

Passed: Input field with ID 's' and search term 'Firth'
Passed: Succesfully clicked on element with ID 'searchsubmit'
Passed: text 'Firth' is in tag 'body'

=====+
| WAT WILT U NU DOEN? |
=====+

1: Voer dezelfde test opnieuw uit
2: Pas de data aan van deze test
3: Maak een nieuwe test (huidige test wordt verwijderd)
4: Zet de resultaten om naar een xls bestand
5: Exporteer test naar .py bestand
6: Stop het programma

Uw keuze:
```

4 Conclusie

Ik kan besluiten dat deze opdracht in het begin zeker geen gemakkelijke taak was. Ik had geen idee hoe ik er aan moest beginnen en zelf na een tijd te zoeken op internet was ik nog steeds niet zeker of ik het wel zou kunnen en dat het niet te hoog was gegrepen. Maar na enige volharding en inzet vond ik de juiste manier om er aan te beginnen en eens ik begonnen was verliep alles vrij goed.

Buiten de wankelende start is alles goed verlopen. Selenium is een vrij bekende package binnenin python en dat merk je direct als je ergens vast zit en je probleem even googled. Je vindt direct minstens 2 tot 3 links naar stackoverflow die zeer goed zijn beantwoord. Alsook de documentatie van Selenium in python was een grote help. Enkele tutorials op Youtube hebben me ook zeer goed geholpen in het maken van mijn programma.

Al bij al ben ik zeer tevreden dat ik dit proof of concept heb gekozen want ik heb hierdoor mijn kennis in Python en mijn kennis in UI testing goed verbreed.

5 Documentatie

Dit zijn de documenten die mij doorheen mijn proof of concept hebben geholpen. Er zitten een paar belangrijke stackoverflows tussen, maar dat zijn ze bijlange niet allemaal natuurlijk.

- <https://sites.google.com/a/chromium.org/chromedriver/getting-started>
- <http://stackoverflow.com/questions/13166395/fill-input-of-type-text-and-press-submit-using-python>
- <http://selenium-python.readthedocs.io/>
- <https://www.youtube.com/watch?v=x18V8SyNiXo>
- <http://stackoverflow.com/questions/11741738/frontend-testing-what-and-how-to-test-and-what-tool-to-use>
- <https://www.smashingmagazine.com/2014/11/how-to-create-your-own-front-end-website-testing-plan/>
- <https://www.youtube.com/watch?v=prxv8VsYljA>
- <https://www.youtube.com/watch?v=gsHyDlyA3dg>
- [https://en.wikipedia.org/wiki/Selenium_\(software\)](https://en.wikipedia.org/wiki/Selenium_(software))
 - Heb ook verschillende Wikipedia pagina's bezocht voor andere UI testers, maar Selenium was de belangrijkste
- <http://www.seleniumhq.org/>

6 Logboek

Datum	Notities
28/12/2016	Eerste onderzoek gedaan naar wat UI testing juist is en welke programma's er allemaal voor zijn.
29/12/2016	Selenium package gedownload in python en begonnen aan UI testing programma.
03/01/2017	Verder gewerkt aan programma. Input test case toegevoegd.
06/01/2017	Click test case toegevoegd.
09/01/2017	Ervoor gezorgd dat je de input en click test cases kunt wijzigen samen met de ingegeven URL.
21/01/2017	De tekst vergelijking test case toegevoegd. De tekst vergelijking test case wijzigen.
22/01/2017	De resultaten kunnen exporteren naar excel en .py
23/01/2017	Extra functies zoals nieuwe test maken, test opnieuw uitvoeren en programma afsluiten toegevoegd. Het maken van minor bugs. De documentatie van proof of concept maken.