De selectie van Javascript gebaseerde front-end testtools voor een KMO gespecialiseerd in Drupal

Onderzoeksvoorstel Bachelorproef

Maarten De Roover[1]

Samenvatting
Vandaag de dag bestaan er veel verschillende tools voor front-end testen die elk hun eigen implementatie en filosofie hebben. Daarom is het voor een bedrijf zoals Dropsolid niet makkelijk om de beste keuze en combinatie van tools te maken die voor hen het beste passen. Dit onderzoek zal bestaan uit twee delen, in het eerste deel ga ik de verschillende tools testen en vergelijken en in het tweede deel ga ik de tools die het beste bij Dropsolid passen gebruiken om een test te maken die op een nieuwe installatie van Rocketship, het Drupal 8 skelet van Dropsolid, Paragraph Types aanmaakt en test. Ik verwacht dat ik aan de hand van de resultaten van het eerste deel een gegronde keuze ga kunnen maken voor een combinatie van tools die mij in staat stellen om het tweede deel van het onderzoek goed uit te voeren. Dit onderzoek zou voor Dropsolid de basis kunnen vormen voor de manier waarop zij front-end testen maken.

Sleutelwoorden
Automated Testing. Javascript — UI testen — Drupal

Co-promotor
Nick Veenhof[2] (Dropsolid)

Contact: [1] maarten.deroover.v8683@student.hogent.be; [2] nick.veenhof@dropsolid.com;

Inhoudsopgave

1 **Introductie** 1
2 **State-of-the-art** 1
3 **Methodologie** 2
4 **Verwachte resultaten** 2
5 **Verwachte conclusies** 2
Referenties 2

1. Introductie

Heden ten dage bestaan er veel verschillende tools, frameworks en libraries voor UI testen die elk hun eigen implementatie en vaak ook hun eigen filosofie in verband met testen hebben. Bijkomend kunnen sommige van deze tools gecombineerd worden voor verschillende use cases.

Dropsolid, een KMO gespecialiseerd in Drupal, gebruikt momenteel nog geen front-end testtools en zou dit in de toekomst graag doen maar weet niet welke tools het beste bij hen zouden passen. De ontwikkelaars binnen Dropsolid die later deze testen zouden schrijven zijn het meest vertrouwd met Javascript en dus verkiest Dropsolid tools die op Javascript gebaseerd zijn. Ook zou dit het beste passen bij de systemen die ze momenteel al gebruiken en de mogelijke integratie versoepelen.

De onderzoeksvraag luidt als volgt: Welke op Javascript gebaseerde front-end testtools kan een KMO gespecialiseerd in Drupal het best gebruiken? Een bijkomende doelstelling is om de verkozen testtools te gebruiken om een test te schrijven die op een nieuwe installatie van Dropsolids Drupal 8 skelet, genaamd Rocketship, de voorgeprogrammeerde Paragraph Types aanmaakt en test.

2. State-of-the-art

De drie belangrijkste testtypes voor websites zijn: unit testen, integratie testen en UI testen. Unit testen zorgen dat de individuele componenten (functies, klassen, ...) van een applicatie correct werken, integratie testen zorgen dat de verschillende componenten met elkaar werken zoals verwacht en UI testen, ook wel functionele of front-end testen genoemd, zorgen dat de applicatie vanuit het perspectief van de gebruiker correct werkt. (Elliott, 2016)

Dit onderzoek zal gaan over het laatste type, de UI testen. Voor UI testen heb je een combinatie nodig van verschillende tools, frameworks en libraries, de voornaamste bespreek ik hieronder.

De eerste soort tools zijn diegene die een dienst verlenen om je te helpen met het uitvoeren van testen op verschillende browsers en apparaten. Enkele voorbeelden zijn BrowserStack, Sauce Labs, BrowseEmAl en Browsershots. (Jackson, 2017)

De tweede soort tools zijn diegene waarmee je je testen schrijft. Selenium, Testcafe, Cypress, Puppeteer, PhantomJS,

Nightmare en Casper zijn hier voorbeelden van. Selenium is een tool die de browser automatiseert en heeft een eigen WebDriver maar die kan uitgebreid worden met libraries zoals Appium, Protractor, WebdriverIO of Nightwatch. Testcafe en Cypress werken niet zoals Selenium, deze automatiseren de browser niet, maar injecteren Javascript scripts in de browser zelf. Puppeteer en PhantomJS zijn tools die gebruikt worden om headless Chrome of Chromium te besturen. Nightmare is een library die bovenop Electron werkt, Electron is gelijkaardig aan PhantomJS. Casper is een hulpprogramma voor PhantomJS voor navigatie scripting en testen.

De derde soort tools zijn diegene die een abstractie laag bovenop een andere testtool toevoegen met Behavior Driven Development als doel. Voorbeelden hiervan zijn Cucumber en CodeceptJS. (Zaidman, 2018)

3. Methodologie

Dit onderzoek zal uit twee delen bestaan. In het eerste deel ga ik de verschillende tools en verschillende combinaties van deze tools vergelijken op verschillende vlakken zoals snelheid waarop de testen uitgevoerd worden, aantal extensies of plug-ins, support van de community, gebruiksvriendelijkheid, hoe goed deze tool past bij Dropsolid, … In het tweede deel zal ik de verkozen tools gebruiken om een test te schrijven die op Rocketship voorgeprogrammeerde Paragraph Types aanmaakt en deze test. Hiervoor zou ik gebruik maken van een virtuele machine waarin ik de site lokaal host. Telkens na het uitvoeren of proberen uitvoeren van deze test zou ik de databank terugstellen naar zijn oorspronkelijke staat zodat ik steeds start van volledig dezelfde website.

4. Verwachte resultaten

Voor het eerste deel van het onderzoek verwacht ik dat ik resultaten ga hebben over de snelheid waarop de testen uitgevoerd worden. Voor de gebruiksvriendelijkheid ga ik stukken code hebben om de syntax, leesbaarheid en complexiteit aan te tonen. Ook voor het aantal extensies en plug-ins en voor de support van een bepaalde tool verwacht ik dat ik resultaten ga hebben. Voor het tweede deel verwacht ik dat het resultaat in staat is om op een nieuwe installatie van Rocketship voorgeprogrammeerde Paragraph Types aan te maken en te testen.

5. Verwachte conclusies

Ik verwacht dat ik aan de hand van het eerste deel van het onderzoek een gegronde keuze ga kunnen maken over welke combinatie van op Javascript gebaseerde front-end testtools het beste passen bij Dropsolid. Dit zou samen met de test die ik in het tweede deel schrijf de basis kunnen worden van front-end testen binnen Dropsolid.

Referenties

Elliott, E. (2016). JavaScript Testing: Unit Vs Functional Vs Integration Tests. Verkregen van https://www.sitepoint.com/javascript-testing-unit-functional-integration/

Jackson, B. (2017). Top 12 Browser Compatibility Testing Tools. Verkregen van https://www.keycdn.com/blog/browser-compatibility-testing-tools/

Zaidman, V. (2018). An Overview of JavaScript Testing in 2018. Verkregen van https://medium.com/welldone-software/an-overview-of-javascript-testing-in-2018-f68950900bc3