

Externe bijlage

Plan van aanpak

Onderzoeksrapport

Plan van aanpak

Naam: Othmar Hawker

Versie 1.1

Opdrachtgever: Mark Baaijens, Klaas Punselie en Maurik Hellebrekers

Datum: 8-augustus-2016

Inhoudsopgave

Externe bijlage	1
Plan van aanpak	1
Onderzoeksrapport	1
Bedrijf	3
Probleembeschrijving	3
Doelstelling van de opdracht	3
Afbakening opdracht	3
Scrum	4
Beschrijven producten	5
Plan van aanpak	5
Onderzoek tools	5
Ontwerpdocument	5
Product backlog	5
Testrapport	5
Afstudeerverslag	5
Planning	6

Bedrijf

Protoneers is een netwerk van professionals dat in Den Haag is gevestigd. Dit bedrijf ontwikkelt minimum viable products (MVP) om te bepalen of het product rendabel is voor de markt. Een MVP is een product dat net genoeg functionaliteiten heeft om te bepalen of de klanten er behoefte aan heeft. Als het product goed is, wordt het verkocht aan de klanten van Protoneers. Deze zijn meestal midden- en kleinbedrijven of startups. De afstudeeropdracht wordt voor EM24ME uitgevoerd en EM24ME is een klant van Protoneers.

Probleembeschrijving

MKB ondernemers maken in veel gevallen gebruik van het internet om hun klanten te bereiken. Voor een groot deel van deze ondernemers vormt de website de hoeksteen van hun product of dienstverlening. Voor partijen zoals webwinkels is de toegankelijkheid en veiligheid van de website van 'levensbelang'. EM24ME heeft ambitie om MKB'ers te helpen met hun security en performance zaken.

In een fysieke winkel kun je als eigenaar inspelen op de zaken die ad hoc gebeuren. Bijvoorbeeld je kunt het omgevallen reclamebord weer rechtzetten, achtergelaten rommel opruimen, etc. Ter vergelijking: in het online kanaal zijn deze zaken op dit moment net wat ingewikkelder maar niet onmogelijk. Op dit moment hebben de MKB ondernemers onvoldoende grip op de security en performance aspecten van hun website.

Doelstelling van de opdracht

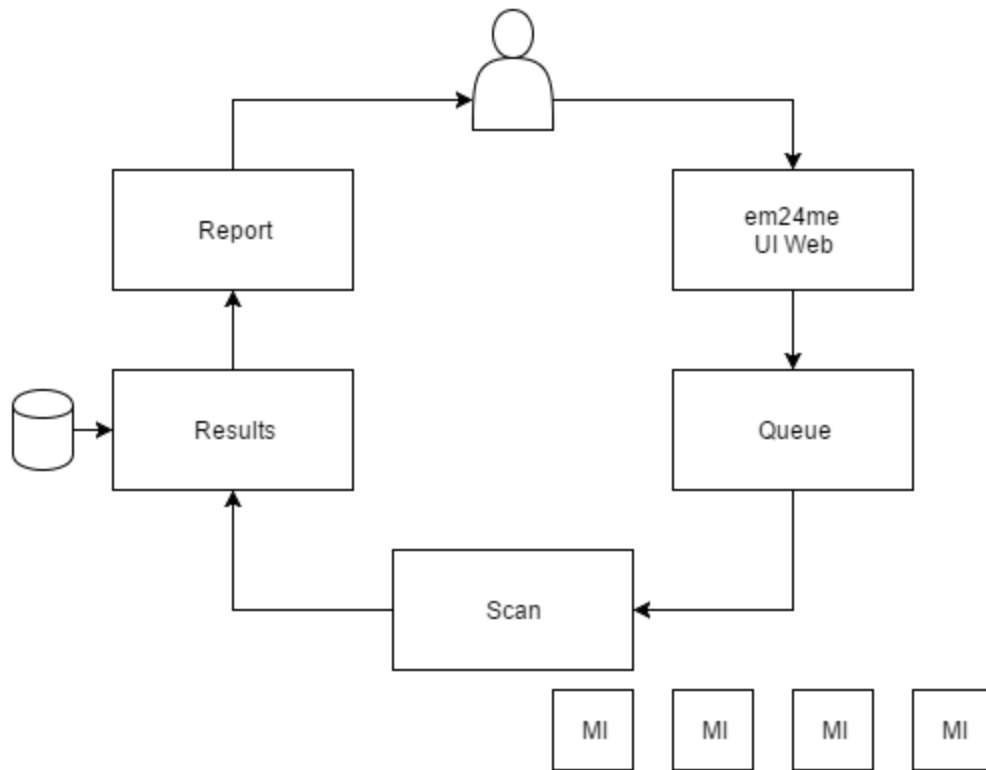
Door middel van een webapplicatie kan de MKB ondernemers meer controle of toezicht krijgen op hun online infrastructuur. Het doel is om de performance, beschikbaarheid, beveiliging en webpagina-integriteit te meten of bepalen. In de toekomst kunnen hier andere aspecten toegevoegd worden. Hier worden de performance en security aspecten gedefinieerd:

- Onder beschikbaarheid wordt de uptime van de server en de status van elke webpagina verstaan.
- Onder performance wordt er gemeten hoe lang het duurt totdat een webpagina volledig laadt.
- Beveiliging wordt verstaan of er een aantal HTTP-security-headers bij de website aanwezig zijn.
- Als laatst wordt onder webpagina-integriteit verstaan of een webpagina op de website is veranderd.

Ook moet er rekening gehouden worden met de onderhoudbaarheid en uitbreidbaarheid.

Afbakening opdracht

In het begin van de opdracht gaan we eerst de beschikbaarheid en performance aspecten van webpagina's meten. Daarna zullen er meer meetinstrumenten zoals webpagina-integriteit toegevoegd worden.



Afbeelding 1: Overzicht van systeemdelen met verbanden.

De gebruiker plant een scan via de website in. Daarna wordt de scan-opdracht in een wachtrij geplaatst. Bij het uitvoeren van de scan worden meetinstrumenten gebruikt en de resultaten worden in de database opgeslagen. Hierna worden de resultaten geanalyseerd en naar de gebruiker via e-mail gestuurd.

Scrum

Voor dit project wordt er Scrum gekozen als softwareontwikkelmethode, want het is een bekende methodiek voor mij en Protoneers. Ik heb in mijn derdejaars stage Scrum gebruikt, dus ik heb voldoende ervaring ermee. Er wordt met een aangepaste Scrum methodiek gewerkt, omdat niet alle activiteiten zijn van toepassing binnen dit project en het ontwikkelteam is niet groot genoeg. De activiteiten die wel gebruik worden zijn: Sprints, user stories, Scrum board en Standup Meeting.

Er zijn totaal 8 sprints. De eerste twee sprints zijn als voorbereiding voor de onderzoeksopdracht. Het maken van een plan van aanpak en een onderzoek voeren. Sprint 3 tot en met sprint 7 voer ik uit de noodzakelijke werkzaamheden, zoals ontwikkelen en testen. De laatste sprint is voor het afronding van het project.

User stories worden vanuit interviews met de opdrachtgever verzameld. Hierna worden de user stories ingevoerd in Trello (online Scrum board). Vervolgens worden de user stories ingepland voor een sprint en daarna worden ze in taken verdeeld.

Voor het bijhouden van de user stories wordt er een Scrum board gebruikt. De Scrum board wordt digitaal op Trello.com bijgehouden. De Scrum board is verdeeld in vier kolommen namelijk: product backlog, inbox, doing en done. In de kolom product backlog staat de user stories en deze userstories worden in taken verdeeld in de inbox kolom. In de kolom doing staan de taken waar ik mee bezig is. In de done kolom staan de user stories en taken die uitgevoerd zijn.

De Standup Meeting wordt drie keer per week gehouden. Twee keer per week met mijn technische begeleider en een keer per week met mijn algemene begeleiding. De Standup Meeting wordt gedaan om de voortgang van het project bij te houden.

Beschrijven producten

Plan van aanpak

Het plan van aanpak is een document dat de opdracht en het bedrijf beschrijft.

Onderzoek tools

Hierin wordt er een onderzoek gedaan naar beschikbare softwarepakketten en of deze pakketten kunnen gebruikt worden om de meetinstrumenten te maken.

Ontwerpdocument

Dit document bevat het ontwerp van alle systeemdelen.

Product backlog

Hierin komen alle taken of pakketjes werk terecht die moeten uitgevoerd worden.

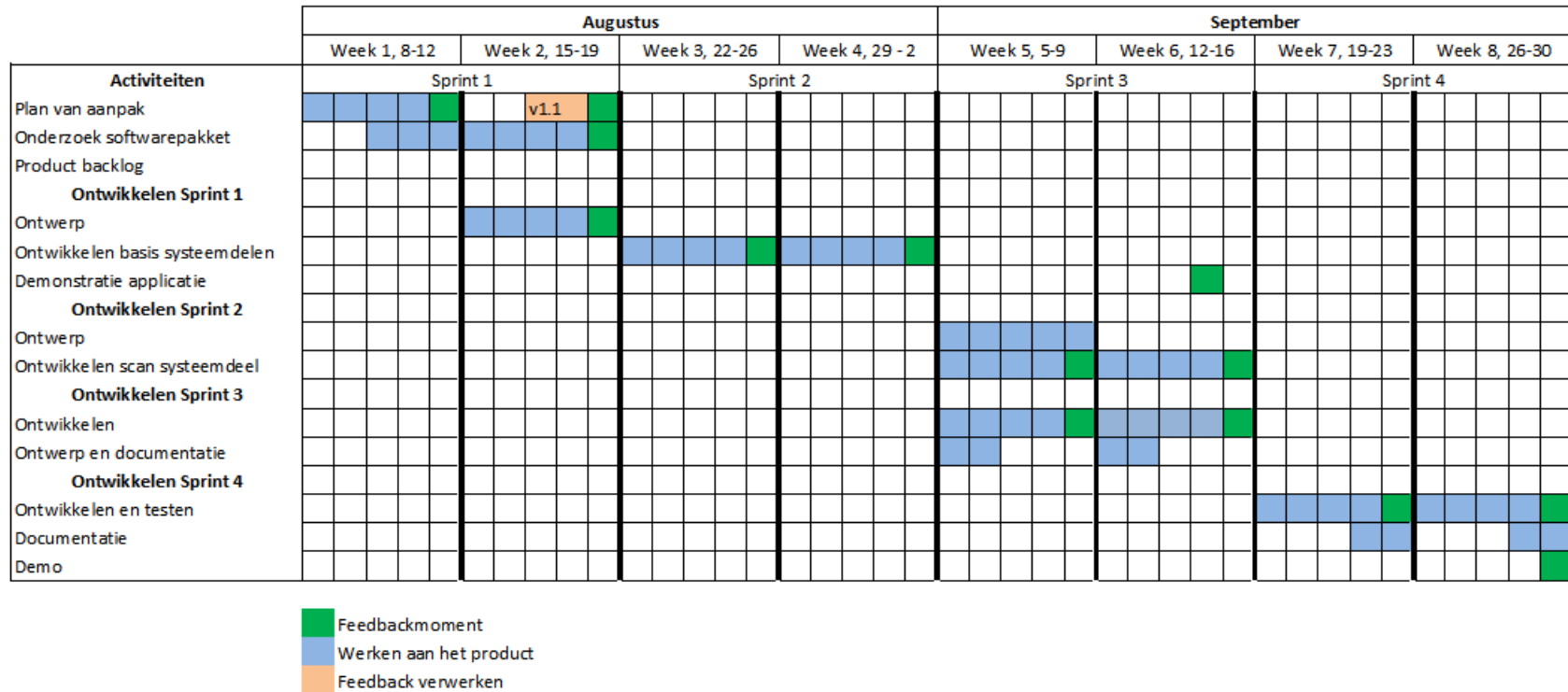
Testrapport

In dit rapport komen de resultaten van de uitgevoerde testen.

Afstudeerverslag

Het afstudeerverslag geeft een inzicht in de omgang en diepgang van mijn verrichte werkzaamheden.

Planning



	Oktober								November											
	Week 9, 3-7		Week 10, 10-14		Week 11, 17-21		Week 12, 24-28		Week 13, 31-4		Week 14, 7-11		Week 15, 14-18		Week 16, 21-25		Week 17, 28-2			
Activiteiten	Sprint 5				Sprint 6				Sprint 7				Sprint 8							
Ontwikkelen Sprint 5																				
Ontwikkelen en testen																				
Documentatie																				
Ontwikkelen Sprint 6																				
Ontwikkelen en testen																				
Documentatie																				
Ontwikkelen Sprint 7																				
Ontwikkelen en testen																				
Documentatie																				
Ontwikkelen Sprint 8																				
Ontwikkelen en testen																				
Documentatie																				
Uitloop																				

Onderzoek naar een site crawler

Naam: Othmar Hawker

Datum: 15-11-16

Opdrachtgever: Protoneers

Inhoudsopgave

Samenvatting	10
Inleiding	11
Probleemstelling	11
Doelstelling	11
Methode	12
Wat is een site crawler?	13
Welke site crawlers zijn er gevonden?	15
Longlist	15
Shortlist	15
Welke site crawler ga ik gebruiken?.....	17
Conclusie	18
Bijlage	19
Literatuurlijst.....	20

Samenvatting

Een site crawler is een bot die een website op een methodische en automatiseerde manier doorbladert. Site crawlers zijn ook gebruikers van het internet en ze zijn 48.5% daarvan. Voor het project EM24ME te realiseren hebben ze een specifieke site crawler nodig en deze moet de volgende eisen hebben:

- Moet de hele website doorzoeken inclusief resources zoals HTML, CSS, scripts en bestanden.
- Moet in de programmeertaal PHP zijn.
- Moet configureerbaar zijn.
- Moet rekening houden met de robot.txt
- Moet de http response en http request zien en aanpassen.
- Moet voldoende documentatie hebben.

Ik heb totaal vijftien site crawlers gevonden en na het selectietraject was er een over. Ik heb voor PHP-Spider gekozen, want het is gemakkelijk te gebruiken, biedt voldoende informatie, is volgens Github heel populair en eenvoudig te configureren.

Inleiding

EM24ME (klant van Protoneers) wil een door middel van een website de MKB ondernemers meer grip geven op hun online infrastructuur. Dit wordt bereikt met een site crawler. Het is tijdrovend om een te maken en er zijn verschillende opensource site crawlers beschikbaar op het internet. De kiezen van een juiste site crawler is de aanleiding voor dit onderzoek.

Probleemstelling

Er zijn verschillende opensource site crawlers beschikbaar en ik wil de beste gebruiken voor deze opdracht. De hoofdvraag luidt als volgt:

Welke opensource site crawler kan Protoneers gebruiken om de EM24ME webapplicatie te maken en wat moet de tool allemaal doen?

Doelstelling

Het doel van dit onderzoek is op een systematische wijze een opensource site crawler softwarepakket kiezen voor het maken van de EM24ME webapplicatie. Bij het kiezen moet ik achterhalen wat de gekozen site crawlers eraan moeten voldoen.

Om antwoord te geven op de hoofdvraag moeten er eerste een aantal deelvragen beantwoord worden;

1. Wat is een site crawler?
2. Aan welke eisen moet de site crawler voldoen?
3. Welke site crawlers zijn er gevonden?
4. Welke site crawler ga ik gebruiken?

Methode

Om de deelvragen te beantwoorden word er per deelvraag een methode vastgesteld om deze te kunnen beantwoorden.

Wat is een site crawler?

Om deze vraag te beantwoorden, wordt er een kort literatuuronderzoek gedaan. Hierin wordt er gezocht naar een definitie.

Aan welke eisen moet de site crawler voldoen?

Om deze vraag te beantwoorden, worden er interviews gehouden met de opdrachtgever en mijn afstudeerbegeleider.

Welke site crawlers zijn er gevonden?

Om deze vraag te beantwoorden, ga ik op opensource repositories naar site crawlers zoeken.

Welke site crawler ga ik gebruiken?

Om deze vraag te beantwoorden ga ik alle site crawlers in de shortlist uitproberen en de beste kiezen.

Wat is een site crawler?

Volgens I. Zeifman (2015, december 9) is dit jaar de eerste keer in vier jaren dat de bots de minderheid op het internet zijn. Er waren 51.5% mensen op het internet en de rest zijn bots, waarvan de meeste slechte bedoelingen hebben. Deze bots zijn web crawlers, site crawlers en web scraper. Voorbeelden van web crawlers zijn:

- Zoekmachines, ze zoeken voor web pagina's, afbeeldingen en andere bestandstypen. Deze informatie wordt in een database opgeslagen.
- Website archivering, het nemen van periodieke opnames van een website en deze opslaan voor de toekomst.
- Data miners, het nemen van gestructureerde informatie van website en deze later analyseren.

Voor dit project hebben we een site crawler nodig. Een site crawler is een bot die een website op een methodische en automatiseerde manier doorbladert. Site crawlers zijn een van de belangrijkste component van een search engine. Ze bladeren door alle beschikbare pagina's van een website en bouwen een kaart van de website.

Aan welke eisen moet de site crawler voldoen?

Er zijn verschillende site crawlers softwarepakket op GitHub. GitHub is een hostingwebsite voor opensourceprojecten. De eisen die ik van de interviews hebben gekregen zijn:

Criterium	Reden
De site crawler moet door de hele website zoeken inclusief resources. De site crawler moet op basis van links door de website doorzoeken.	De site crawler moet de HTML pagina's, CSS scripts, javascript en ook bestanden doorzoeken, want die bestanden en scripts worden waargenomen.
De programmeertaal van de site crawler moet PHP zijn.	De framework van het systeem is in PHP geschreven en als de site crawler ook in PHP is, maakt het eenvoudig om de site crawler daarin te gebruiken.
Het moet gemakkelijk zijn om de site crawler te configureren.	Er moet instellen welke resources (HTML, CSS of bestanden) de crawler gaat doorzoeken, ook of de site crawler door de subdomains gaat zoeken en of de crawler links gaat filtreren.
Het moet mogelijk zijn om de HTTP-request en de HTTP-response aan te passen.	Door de HTTP-request aan te passen kan de site crawler op bepaalde webpagina's in te loggen.
De site crawler moet voldoende documentatie aangeven en wat zegt de GitHub metrics over de populariteit van de site crawler?	Voldoende documentatie maakt het duidelijk wat de site crawler doet en hoe het gebruikt moet worden. Met behulp van de Github metrics weten we hoe populair of vertrouwd de site crawler is. De metrics zijn: aantal stars, aantal watches en aantal forks.

Tabel 1: Eisen van de site crawler

Voor het selectietraject is er vijftien site crawlers gevonden. Deze is gebaseerd op tips, onderzoek via internetbronnen en GitHub.

Welke site crawlers zijn er gevonden?

Longlist

In de bijlage is er een lijst van de meest bekende site crawlers op GitHub. Er is een zoekopdracht op GitHub met de volgende zoektermen: “site crawler”, “web spider” en “web crawler”. Er is totaal vijftien bruikbare site crawlers gevonden. De lijst van site crawlers is in de bijlage te zien.

Vervolgens wordt er een selectie om deze lijst tot een shortlist te verkleinen gedaan. De knockout criteriums van deze lijst zijn: de site crawler moet PHP zijn en de site crawler moet voldoende documentatie bevatten. Hierna zijn er vijf site crawler overgebleven.

Shortlist

Om een site crawler uit de shortlist te kiezen ga ik alle vijf site crawlers uitproberen. Daarna kan ik bepalen welke site crawler het meest voor deze opdracht geschikt is. Eerst geef ik een korte beschrijving van de site crawler en daarna geef ik mijn bevindingen over de site crawler.

PHP-Crawl

PHP-Crawl is een framework voor het doorzoeken van websites in de PHP taal. Deze framework zoekt door een website en het passeert alle informatie(pagina's, links, bestanden enz.) naar de gebruiker van de framework. PHP-Crawl is gratis en opensource.

Deze site crawler is heel configureerbaar, het is mogelijk te kiezen welke resources het doorzoekt en hoe het de resources opslaat. Wat er bij deze site crawler ontbreekt, is dat het is niet mogelijk om een HTTP-request te wijzigen of inzien.

PHP-Spider

PHP-Spider is een configureerbare en uitbreidbare PHP web crawler. De kenmerken van deze web crawler is:

- Het is mogelijk je eigen URI filter die op XPath¹ is gebaseerd toepassen. XPath is een querytaal voor het adresseren van onderdelen van XML-documenten.
- Event handlers vooraf en na het ophalen van de URI.
- Bevat Guzzle. Guzzle is een PHP HTTP-client voor het sturen van HTTP-request. Hiermee is het mogelijk om de HTTP-headers te wijzigen en inzien.

Deze site crawler maakt het heel simpel om het httpverkeer te wijzigen en te aanpassen. Dit vind ik heel belangrijk. Ook is bijna alle functionaliteiten van deze site crawler configureerbaar. Het biedt ook voldoende documentatie om je eigen functionaliteiten erbij te implementeren.

Deze site crawler heeft voldoende documentatie en het voldoet met alle opgestelde criteriums. Ik heb voorkeur voor deze site crawler.

¹ Bron - <https://www.w3.org/TR/xpath/>

Crawl Anywhere

Crawl Anywhere is voornamelijk een web crawler. Crawl Anywhere bevat:

- Een documentverwerking pipeline die alle informatie (websitenaam, land, url, encoding enz) over de inkomende documenten bevat.
- Bevat Apache Solr, een zoekmachine met een tag cloud analyser.
- Bevat ook een front-end applicatie voor de zoekmachine.

De eerste indruk die ik bij deze site crawler kreeg is dat het meer op een zoekmachine lijkt. Crawl Anywhere is een groot softwarepakket, het bevat: web administratie, crawler, documentverwerking pipeline, Solr zoekmachine en zoekmachine front-end. De crawler kan alleen de inkomende links analyseren en niet de uitgaande links.

FCC/Crawler

Crawler is een kale site crawler die snel en effectief een index van alle bestanden en pagina's (bouwt ook de relatie tussen pagina's) op een bepaalde website kan bouwen.

Deze site crawler heeft bijna geen documentatie en voelt niet compleet. Er is ook geen mogelijkheid om HTTP-request te wijzigen. Deze site crawler zou ik niet gebruiken.

Spatie/crawler

Crawler kan alle links op een website doorzoeken. Functionaliteiten van deze crawler zijn:

- Bepalen van alle links op een webpagina.
- Normaliseren van links. Een relatieve link zoals /contactpagina naar een absoluut link (<http://voorbeeld.com/contactpagina>) overmaken.
- Bepalen of een link doorgezocht moet worden.
- Een webpagina doorzoeken en het proces observeren.
- Kan de HTTP-request en HTTP-response inzien.

Deze site crawler kan alle links en resources op een website doorzoeken, biedt voldoende documentatie en kan de HTTP-request en HTTP-response inzien. Als laatste is er geen gemakkelijk manier nieuwe functionaliteiten bij deze site crawler toe te voegen.

Deze site crawler heeft voldoende documentatie en doet alles wat ik wil. Ik heb voorkeur voor deze site crawler.

Welke site crawler ga ik gebruiken?

Uit de shortlist heb ik voorkeur voor PHP-Spider en Spatie/crawler. Ze zijn zeer vergelijkbaar met elkaar, maar volgens GitHub is PHP-spider populairder dan Spatie/crawler en heeft meer configuratiemogelijkheden dan Spatie/crawler.

Ik heb voor PHP-spider gekozen, want het biedt voldoende documentatie, is volgens GitHub heel populair, bevatte mogelijkheid om het HTTP verkeer te wijzigen en is gemakkelijk om wijzigingen in te brengen. Als laatste heeft PHP-spider meer configuratiemogelijkheden dan Spatie/crawler.

Conclusie

Een site crawler is een tool die een hele website kan doorzoeken. Wij willen een heel specifieke site crawler gebruiken die de volgende eisen beheerst:

- Kan het HTTP verkeer analyseren en wijzigen.
- Kan de hele website doorzoeken.
- Moet gemakkelijk zijn om functionaliteiten erbij te voegen.

Als laatst is er voor PHP-spider gekozen, want het biedt voldoende documentatie, is volgens GitHub heel populair, bevatte mogelijkheid om het HTTP verkeer te wijzigen en is gemakkelijk om wijzigingen in te brengen.

Bijlage

Criteria/Site crawler	mmerian/PHPCrawl	mvdbo's/php-spider	Spider	Scrapy	Beautiful Soup	Apache Nutch	crawler4j	StormCrawler	Crawl Anywhere	FCC/Crawler	spatie/crawler	symfony/dom-crawler	PHP-Crawler	hightman/pspider
Is het PHP taal?	✓	✓	✓	✗	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓
Zoekt de crawler een hele website inclusief resources?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
Is het makkelijk te configureren?	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✗
Kunnen we de http request en reponse wijzigen?	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗	✓	✗	✗	✗
Wordt het softwarepakket onderhouden, heeft voldoende documentatie en is het populair?	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✗

Tabel 2: Longlist opensource site crawlers

✓ Site crawler voldoet aan de gestelde criteria.

✗ Site crawler voldoet niet aan de gestelde criteria.

Literatuurlijst

Zeifman, I. (2015, December 9). 2015 Bot Traffic Report: Humans Take Back the Web, Bad Bots Not Giving Any Ground. Opgehaald 24 oktober 2016, van <https://www.incapsula.com/blog/bot-traffic-report-2015.html>

Bos, M. V. (2015, October 10). Mvdbos/php-spider. Opgehaald 29 august 2016, van <https://github.com/mvdbos/php-spider>

Mérian, M. (2015, October 5). Mmerian/phpcrawl. Opgehaald 05 september 2016, van <https://github.com/mmerian/phpcrawl>

Scrapy | A Fast and Powerful Scraping and Web Crawling Framework. (n.d.). Opgehaald 5 september 2016, van <https://scrapy.org/>

Richardson, L. (n.d.). Beautiful Soup. Opgehaald 5 september 2016, van <https://www.crummy.com/software/BeautifulSoup/>

D. (n.d.). Highly extensible, highly scalable Web crawler. Opgehaald 5 september 2016, van <http://nutch.apache.org/>

Y. (2016, August 10). Yasserg/crawler4j. Opgehaald 5 september 2016, van <https://github.com/yasserg/crawler4j>

D. (2016, August 31). DigitalPebble/storm-crawler. Opgehaald 5 september 2016, van <https://github.com/DigitalPebble/storm-crawler>

FCC. (2012, June 05). FCC/Crawler. Opgehaald 5 september 2016, van <https://github.com/FCC/Crawler>

Symfony. (2016, August 10). Symfony/dom-crawler. Opgehaald 5 september 2016, van <https://github.com/symfony/dom-crawler>

Herten, F. V. (2016, August 18). Spatie/crawler. Opgehaald 05 september 2016, van <https://github.com/spatie/Crawler>

Hunfeld , U. (n.d.). PHPCrawl webcrawler library/framework. Opgehaald 5 september 2016, van <http://phpcrawl.cuab.de/>