





De speeltuin op het web

Auteur Brett Heesakkers	Opleiding ICT Media Design	Instelling Fontys Hogescholen	Klas IMD 8.0	Studentnummer 2049544	Datum 14-01-2010
-----------------------------------	--------------------------------------	---	------------------------	---------------------------------	----------------------------



Voorwoord

Graag wil ik Sense Interactive bedanken voor de leuke tijd die ik heb gehad tijdens het werken aan deze opdracht. De vriendelijke sfeer was fijn om in te werken.

Ook wil ik Jasper Ancher, Henrik van Duijnhoven, Tom van Lamoen, Paul van der Avoort, Eelco Veldhuizen en Joost Timmermans bedanken voor hun hulp wanneer ik dat nodig had.

Verder bedank ik Tosca, Daan en Colin voor gezelligheid tijdens de stage periode.

Mijn dank gaat ook uit naar mijn stagedocente Lucienne Wijgengangs voor de feedback en hulp tijdens de afstudeerperiode. Ook wil ik Sjoerd Joosten, Jan Stoker en Joyce van der Sagen bedanken voor de feedback op mijn scriptie.



Inhoudsopgave

Samenvatting	4	4.4 Netvibes	18
Summary	5	Wat is Netvibes?	18
Verklarende woordenlijst	6	De Netvibes API	18
Referenties	6	Beperkingen	19
Definities en afkortingen	6	Het design	19
Definities	6	Conclusie Netvibes	19
Afkortingen	6	5. Koppeling met Playground	20
1. Inleiding	7	6. Conclusie(s) en aanbeveling(en)	22
2. Het bedrijf	8	Evaluatie	23
Sense Interactive	8	Literatuurlijst	25
Organisatie	8	Bijlagen	
3. De opdracht	9	I. Plan van Aanpak	
De beginsituatie	9	II. Opdracht Sense Interactive	
Doel van het project	9	III. De koppeling met Playground 2.0	
Opdracht	9	IV. Zend_Service_Twitter methodes	
Werkwijze	9	V. cURL	
Onderzoeksmethodes	10	VI. OAuth	
4. Onderzoek	11	VII. PrototypeJS	
4.1 Vooronderzoek	11		
Wat is Playground?	11		
Hoe werkt het?	11		
Playground 2.0	12		
Conclusie Playground onderzoek	13		
4.2 Gekozen diensten	14		
4.3 Twitter	15		
Het idee achter Twitter	15		
De Twitter API	15		
Beperkingen	15		
Het design	16		
Conclusie Twitter	16		

Samenvatting

Deze opdracht is gemaakt voor Sense Interactive. Dit is een bedrijf dat websites ontwikkelt met behulp van hun eigen ontwikkel omgeving Playground. Playground biedt de klanten van Sense Interactive de mogelijkheid om hun website(s), nieuwsbrieven en databases te beheren. Ook biedt Playground voor de programmeurs de mogelijkheid om gebruik te maken van eigen ontworpen Playground tags. Met behulp van deze tags kan een programmeur gebruik maken van templates.

Sense Interactive wil dat Playground ook beschikt over diensten zoals Flickr, Twitter, Netvibes etc. Deze diensten worden aan Playground 2.0 gekoppeld. Deze versie van Playground gaat vele nieuwe mogelijkheden brengen voor de klanten en de programmeurs van Sense Interactive. Hieronder vallen dan ook de diensten die gekoppeld worden.

Een dienst zoals Twitter, heeft zijn techniek beschikbaar gesteld door middel van een Application Programming Interface (API). Om Playground 2.0 van Twitter te voorzien moet dus zijn API gekoppeld worden. Voor deze opdracht zijn dat Twitter en Netvibes geworden.

Om deze opdracht succesvol af te ronden is er eerst een onderzoek gedaan naar Playground. Door mee te draaien in de productie van Sense Interactive is de kennis van Playground vergroot. Verder is gekeken naar de 2.0 versie van Playground. Deze maakt gebruik van het Zend Framework. Dit zijn verschillende PHP: Hypertext Preprocessor (PHP) methodes die gebruikt konden worden bij het programmeren. Hieruit is naar boven gekomen dat Zend beschikt over een aantal Twitter methodes. De aanvraag van de gegevens van Twitter worden door deze methodes afgehandeld.

Vervolgens werd naar de inhoud van deze twee diensten en de werking van hun API gekeken. Met behulp van Zend was voor Twitter de aanvraag afgerond. Het resultaat was echter nog niet in het goede formaat. Het was een object en het moet een Extensible Markup Language (XML) formaat zijn. Dit komt omdat Playground met XML werkt. Nadat het formaat goed was kon Playground de gegevens uitlezen.

Netvibes heeft zijn eigen PHP scripts ter beschikking gesteld. Hiermee kunnen de Netvibes gegevens opgehaald worden. Ook kunnen met behulp van deze scripts widgets opgebouwd worden. Het aansturen van de widgets gebeurt met JavaScript. Deze scripts worden ook door Netvibes aangeboden.

Nadat de werking van de twee API's duidelijk was zijn de PHP scripts voor Playground gemaakt. Het is aan Sense Interactive om de scripts te integreren in de nieuwe versie van Playground voordat deze gelanceerd wordt.

Doordat deze twee API's constant in ontwikkeling zijn is het nodig om zo nu en dan de API documentatie door te nemen en eventueel functionaliteit van de scripts aan te passen.

Nu de Twitter en Netvibes API's gekoppeld zijn aan Playground 2.0 kunnen de klanten ook gebruik maken van deze diensten op hun eigen website. Hiermee laat Playground 2.0 werkelijk zien dat alles wat een klant nodig heeft op één plaats te vinden is.

Summary

This assignment has been made for Sense Interactive. This is a company that develops Internet sites using Playground, their own developing environment. Playground offers the customers of Sense Interactive the possibility to manage their Internet sites, newsletters and databases. Playground also offers the possibility for the programmers to use some of their own devised Playground tags. Using these tags a programmer can make use of templates.

Sense Interactive wants Playground to also be able to use services such as Flickr, Twitter, Netvibes etc. These services are to be linked to Playground 2.0. This version of Playground will bring a lot of new possibilities for the customers and the programmers of Sense Interactive. The linked services will also be a part of this.

A service such as Twitter, has made its technique available by means of an Application Programming Interface (API). Thus to provide Playground 2.0 with Twitter, its API must be linked. For this assignment these are Twitter and Netvibes.

To finish this task successfully a study into Playground has been done first. By working on common tasks inside Sense Interactive the knowledge of Playground is increased. There has also been a study to the 2.0 version of Playground. This makes use of the Zend Framework. These are several PHP: Hypertext Preprocessor (PHP) methods which could be used for programming. From this the knowledge came that Zend has a number of Twitter methods. The request for the data of Twitter is handled by these methods.

Then the contents of these two services and the functioning of their API was looked at. Using Zend the request for Twitter had been finished. The result was however not yet in the right format. It was an object and it must be an Extensible Markup Language (XML) format. This is because Playground works with XML. After the format was rightly formed Playground could read the data.

Netvibes has his own PHP script made available. With these scripts the Netvibes data can be picked up. Widgets can also be built using these script. Controlling these widgets happens with Java script. These scripts are also provided by Netvibes.

After the functionality of the two APIs were clear the PHP scripts for Playground were made. It is up to Sense Interactive to link these scripts in the new version of Playground before its launched.

Because these two APIs are constantly in development it is necessary to go through the API documentation now and then and possibly add functionality to the scripts.

Now that the Twitter and Netvibes APIs are linked to Playground 2.0 the customers can also make use of these services on their own Internet site. With this Playground 2.0 really shows that everything what a customer needs can be found in one place.

Verklarende woordenlijst

Referenties

Verwijzing	Betekenis	Auteur
PvA	Plan van Aanpak	Brett Heesakkers

Definities en afkortingen

Definities

Definitie	Betekenis
Bibliotheek	Een verzameling code met functies / methodes.
Class	Een sjabloon voor objecten.
Framework	Een framework is een geheel van softwarecomponenten dat gebruikt kan worden bij het programmeren van applicaties.
JavaScript	Een programmeer taal om internetpagina's interactief te maken en om webapplicaties te ontwikkelen.
Methode	Een functie dat zich in een class bevindt.
Netvibes	Een omgeving waarin de gebruiker zijn eigen webbeleving creëert.
PG-tag	Een tag speciaal ontworpen voor het Playground framework.
Script	Een bestand met programmeertaal.

tag	Een benaming voor een stukje code wat de start van een de code weergeeft. HTML heeft bijvoorbeeld de tag: <table>. De browser leest deze tag en weet dat de programmeur hiermee een tabel bedoelt. Hierna laat de browser vervolgens een tabel op het scherm zien.
Template	Een stuk code dat opnieuw gebruikt kan worden maar waar wel verschillende inhoudende aan gegeven kunnen worden.
Twitter	Een sociaal netwerk waar gebruikers een kort bericht achter kunnen laten.

Afkortingen

Afkorting	Betekenis
AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
CMS	Content Management System
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
JSON	JavaScript Object Notation
PHP	PHP: Hypertext Preprocessor
REST	Representational State Transfer
XML	Extensible Markup Language

1. Inleiding

Alles wat je maar nodig hebt in één omgeving! Zo dacht Jasper Ancher, oprichter van Sense Interactive een jong bedrijf gevestigd in Uden, erover toen hij begon met het bouwen van Playground. Een gemakkelijk te beheren website voor klanten die eenvoudig te bouwen is door de programmeurs.

Het zou nog mooier zijn wanneer er gebruik gemaakt kan worden van API diensten binnen Playground. Hiermee kunnen de klanten van Sense Interactive, binnen Playground op hun dashboard, diensten als Twitter gebruiken. Het aanbieden van de gelegenheid om gebruik te maken van deze API diensten zorgt ervoor dat Playground meer mogelijkheden biedt voor haar klanten.

Momenteel beschikt Playground niet over deze diensten. Bovendien heeft Playground ook nog geen dashboard. Deze nieuwe technieken worden in Playground 2.0 geïntroduceerd. Playground 2.0 is op dit moment in ontwikkeling en zal eind februari worden gelanceerd. Om deze nieuwe versie te voorzien van online diensten is er een opdracht ontstaan om twee van deze diensten te koppelen. Deze diensten worden uiteindelijk widgets die op het dashboard kunnen worden gesleept.

Om deze opdracht tot uitvoering te brengen is er eerst een onderzoek gestart naar de werking van Playground en de twee gekozen diensten.

De opdracht is op te delen in de volgende hoofdstukken. In hoofdstuk twee komt informatie over het bedrijf aan bod, hoofdstuk drie bevat de details van de opdracht, in hoofdstuk vier wordt het onderzoek besproken en in hoofdstuk vijf het resultaat van de koppeling met Playground 2.0. Tot slot komt in hoofdstuk zes de conclusie(s) en aanbeveling(en).



2. Het bedrijf

Sense Interactive

Sense Interactive is een innovatief multimediabureau met een eigenwijze kijk op websiteontwikkeling. Altijd op zoek naar nieuwe technologieën, nieuwe functionaliteiten en ideeën. Topcreativiteit om de klanten perfecte oplossingen te bieden die de verwachtingen van websitebezoekers overtreft.

Sense Interactive ontwerpt en realiseert websites, zowel standaardoplossingen als maatwerk. Allemaal gebaseerd op het door hen zelf ontwikkelde Playground. Sense Interactive werkt indirect voor reclamestudio's en reclamebureaus, drukkerijen, pr-bureaus, IT-bedrijven en marketingcommunicatie adviseurs via het eigen reseller-netwerk.

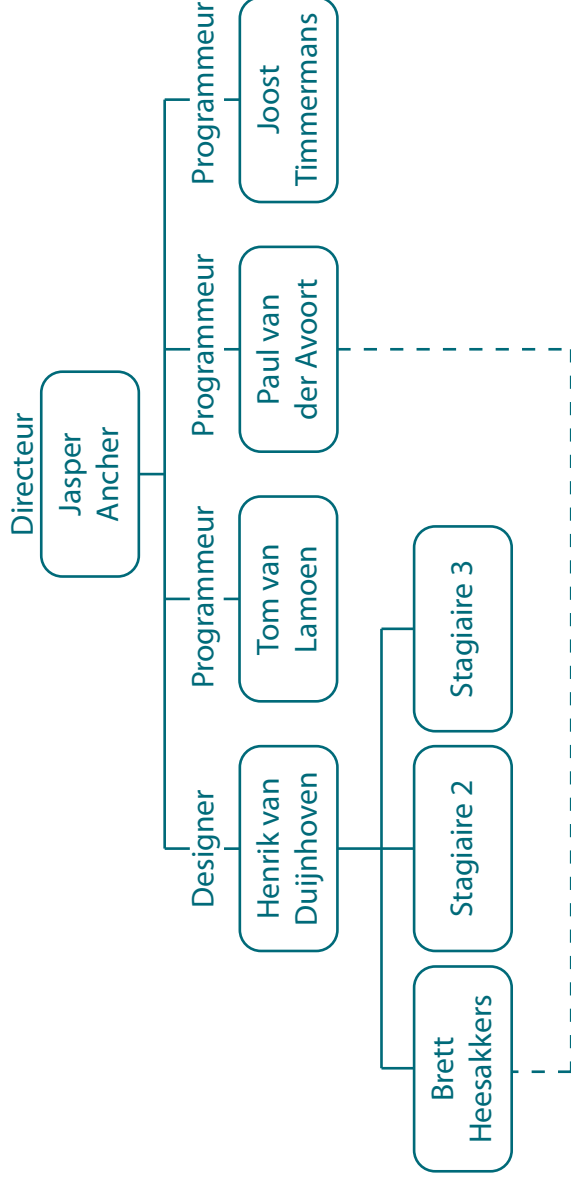
Sense Interactive is opgericht door Jasper Ancher met als missie snel en betaalbaar websites te ontwikkelen die volledig voldoen aan de laatste ontwikkelingen.

De visie van Sense Interactive is om de klanten te voorzien van de beste oplossing voor hun websites en online diensten met behulp van de nieuwste technieken, bestaande online diensten en maatwerk oplossingen samengebracht in één omgeving, Playground.

Organisatie

Bij Sense Interactive wordt gewerkt in een team. Er wordt gezamenlijk gewerkt aan projecten waarin Jasper Ancher en Henrik van Duijnhoven de taken verdelen aan de andere medewerkers.

Sense Interactive



De stagiairs en afstudeerders worden begeleid door Henrik van Duijnhoven voor wat betreft het algemene proces. Voor dit project wordt de afstudeerder begeleid door Paul van der Avoort bij de technische aspecten.

3. De opdracht

De beginsituatie

Het huidige Playground beschikt nog niet over online diensten zoals Twitter, Flickr, Netvibes etc. Vele van deze diensten hebben een API vrijgegeven zodat ontwikkelaars deze diensten kunnen gebruiken in hun eigen project. Sense Interactive is bezig met het ontwikkelen van een nieuwe versie van Playground wat een ideaal moment biedt om deze diensten te integreren.

Doel van het project

In 20 weken de nieuwe versie van Playground voorzien van online diensten zodat de klanten van Sense Interactive hier gebruik van kunnen maken.

Opdracht

De opdracht bestaat uit een onderzoek naar online diensten en het integreren van twee van deze diensten in de nieuwe versie van Playground.

Werkwijze

Om het doel van dit project te behalen is er een werkwijze bepaald.

Als eerst wordt er een Plan van Aanpak gemaakt (zie PvA bijlage I). Dit helpt bij het maken van een overzicht van het project en het beschrijven van de opdracht en de betrokken partijen. In het Plan van Aanpak komt ook de planning zodat het project makkelijker te managen is en er duidelijk wordt wanneer alles afgerond is.

Tijdens het maken van het Plan van Aanpak worden er een aantal opdrachten in het productieproces van Sense Interactive ontwikkeld. Door mee te draaien met de productie, en dus websites te ontwikkelen voor klanten van Sense Interactive, worden de eerste grondstenen gelegd voor het werken met Playground. Dit dient als een kennismaking met de Playground interface.

Vervolgens wordt gekeken naar Playground. Hierin wordt beschreven wat Playground inhoud en wordt kort de functionaliteit uitgelegd.

Hierna wordt onderzocht welke online diensten er zijn, wat deze diensten doen en welke gekozen worden voor de integratie met de nieuwe versie van Playground.

Dan komen de twee gekozen diensten aan bod. Er wordt uitgelegd wat deze diensten inhouden en hoe hun API's werken.

Hierop volgt de koppeling van de API's aan Playground 2.0.

Tot slot wordt er een conclusie en aanbeveling geschreven.

3. De opdracht

Onderzoeksmethodes

Voor dit onderzoek wordt voornamelijk gebruik gemaakt van desk research.

Met behulp van deze methode wordt de diensten onder de loep genomen en wordt de ondersteunende kennis van de verschillende programmeertalen opgehaald. Hierbij zal het Internet een grote rol spelen.

Bij het onderzoek van Playground wordt veel gebruik gemaakt van het doornemen van de programmeercode. Hier wordt dan voornamelijk de PHP programmeercode bekeken. Zo wordt duidelijk wat de structuur en de werking van Playground inhoud.

Een andere methode die veel gebruikt wordt is Trial and Error. Tijdens het bouwen van de Playground API's zullen er veel tests plaatsvinden. Nadat er een stuk code geschreven is moet gekeken worden of het stuk het gewenste resultaat oplevert. Om te kunnen testen worden de bestanden op de server van Sense Interactive gemaakt. Deze server beschikt over de mogelijkheid om PHP code te lezen en verwerken.

Nadat het bestand is opgeslagen wordt in een browser, dit kan Internet Explorer, Chrome, Firefox of Safari zijn, de pagina geopend. Nu kan gekeken worden of de code werkt door het resultaat in de browser te tonen. Is het resultaat niet zoals gewenst dan wordt de code aangepast en begint de test opnieuw. Dit is een veel gebruikte methode om te kijken of wat gemaakt is ook werkt.

4. Onderzoek

Om Playground te voorzien van online diensten is er een onderzoek naar deze diensten ontstaan. Voordat deze diensten onderzocht worden is er eerst een vooronderzoek geweest naar. In dit onderzoek komt Playground aan bod. De kennis die hieruit op is gedaan wordt gebruikt bij het onderzoek van de twee diensten.

4.1 Vooronderzoek

Wat is Playground?

Omdat het aantal beschikbare online diensten steeds groter wordt is het voor de gebruikers van deze diensten moeilijk om alles overzichtelijk te houden. Als een gebruiker dan ook nog een eigen website heeft die hij of zij moet beheren is het overzicht zoek. Waarom kan het niet op een plaats samenkomen? Om dit op te lossen is Sense Interactive begonnen met het ontwikkelen van een omgeving waar al deze online diensten samenkomen. Deze omgeving is gedoopt met de naam Playground.

De klanten van Sense Interactive kunnen met Playground op een eenvoudige manier hun website beheren. Ook is er de mogelijkheid om te beschikken over een database module. Met deze module kan de klant de database op een overzichtelijke manier bekijken en wijzigen.

Playground is meer dan alleen een Content Management System (CMS) voor klanten. Het biedt niet alleen een omgeving om websites en databases te beheren zoals de meeste CMS omgevingen. Playground heeft namelijk ook de mogelijkheid om gebruik te maken van de beschikbare gekoppelde online diensten. Hierdoor biedt Playground de klant meer mogelijkheden om de gebruikers van hun website te voorzien van meer dan alleen informatie.

Hoe werkt het?

Playground is een ontwikkelomgeving met een eigen PHP framework. Door gebruik te maken van eigen ontworpen tags kan het framework met de HTML en PHP pagina's communiceren. Door middel van een parser, dit is een PHP script dat een HTML pagina leest en bepaalde woorden vangt, worden deze tags, de PG-tags, omgezet in bruikbare code wat door de browser gelezen kan worden.

Voorbeeld PG-tag: '`<pgx:core.html />`'

De pagina met deze tag wordt door de parser gehaald. De parser weet in dit geval dat de pagina '`pgx.config.xml`' in de folder structuur '`/pgxframework/core/html/`' aangeroepen moet worden. Vervolgens weet de parser aan de hand van het XML bestand welke versie folder hij moet openen om zo '`html.php`' aan te roepen.

Om bijvoorbeeld de functionaliteit van Twitter aan te roepen wordt gebruik gemaakt van '`<pgx:local.twitter />`'. Dit roept direct de '`twitter.php`' pagina aan.

Doordat Playground een eigen framework heeft kan er, met behulp van PG-tags, gewerkt worden met templates. Een template is een los staande HTML pagina met zo weinig mogelijk HTML tags. Op deze manier kan de website opgebouwd worden zonder dat er inhoud, of een grote hoeveelheid aan HTML, in de '`index.html`' toegevoegd hoeft te worden. Ook kan een template op maat gemaakt worden zodat deze aan de wensen van de klant voldoet.

Op deze manier is het voor de programmeur makkelijker te werken en blijft de inhoud van de code gescheiden.

4. Onderzoek

Om de Playground interface dynamisch en grafisch aantrekkelijk te maken wordt er gebruik gemaakt van PrototypeJS. Dit is een JavaScript framework wat veel functionaliteit bevat voor het maken van dynamische webapplicaties. Met behulp van PrototypeJS kan bijvoorbeeld eenvoudig gebruik gemaakt worden van Asynchronous Javascript And XML (AJAX) functionaliteit.

Met behulp van AJAX kan JavaScript een pagina laden zonder dat de browser opnieuw hoeft te laden. Hierdoor hoeft maar een gedeelte van de pagina opnieuw geladen te worden, dit gebeurt meestal in een HTML div tag. Een div staat voor een container op een website waar inhoud in geplaatst kan worden.

Playground 2.0

Doordat Playground niet aan de visie van Sense Interactive voldoet is er besloten om een nieuwe versie te gaan ontwikkelen met de naam Playground 2.0.

In Playground 2.0 wil Sense Interactive de verschillende online diensten op zichzelf laten staan. Ze zijn als objecten gekoppeld aan Playground zodat deze diensten via Playground aan te roepen zijn.

Nieuw in Playground 2.0 is dat het gebruik maakt van het Zend Framework.

Het Zend Framework is een open-source framework, ontwikkeld in object-georiënteerde PHP5 code door Zend Technologies. Het framework is gemaakt met de gedachte om het gebruik ervan zo simpel mogelijk te houden.

Zend beschikt over een aantal handige methodes die gebruikt kunnen worden voor het programmeren van applicaties.

Zo heeft het framework ook een Twitter functionaliteit door gebruik te maken van de Twitter Representational State Transfer (REST) API.

Een API is een verzameling definities op basis waarvan een computerprogramma kan communiceren met een ander programma of onderdeel (meestal in de vorm van bibliotheken). Vaak vormen API's de scheiding tussen verschillende lagen van abstractie, zodat applicaties op een hoog niveau van abstractie kunnen werken en het minder abstracte werk uitbesteden aan andere API's. Hierdoor hoeft bijvoorbeeld een tekenprogramma niet te weten hoe het de printer moet aansturen, maar roept het daarvoor een gespecialiseerd stuk software aan in een bibliotheek, via een afdruk-API.

Door het aanroepen van de `Zend_Service_Twitter_Class` kan eenvoudig een Twitter object aangemaakt worden. Dit object kan op zijn beurt de verschillende methodes aanroepen om bijvoorbeeld de gebruiker zijn Tweets op te vragen.

De verbeteringen liggen ook in het gebruik van de Playground omgeving en de manier waarop de programmeur een website bouwt. Met Playground 2.0 kan de programmeur bijvoorbeeld eenvoudig ronde hoekjes in een website laten tonen zonder dat deze plaatjes hoeft te gebruiken. Hierdoor hoeft de programmeur geen Photoshop te gebruiken om de hoekjes uit te halen en deze op de juiste plaats te positioneren. Dit scheelt veel tijd en geld.

Playground 2.0 maakt meer gebruik van JavaScript dan zijn voorganger. Dit geeft de mogelijkheid om mooiere en betere website te ontwikkelen. Bovendien wordt er een dashboard met JavaScript ontwikkeld wat de mogelijkheid biedt om een eigen omgeving in te richten met widgets. Een widget is een kleine applicatie wat op een platform draait en ervoor zorgt dat de gebruikers externe gegevens op hun persoonlijke website kan bekijken.

Een nadeel van deze technieken is dat oudere browsers niet meegaan met deze ontwikkeling wat problemen kan geven voor Playground. Aan de andere kant kunnen de modernere browsers steeds sneller JavaScript lezen en uitvoeren wat ten goede komt van Playground 2.0.

4. Onderzoek

Ook wordt de interface van Playground 2.0 verbeterd. De huidige interface werkt wat ongemakkelijk en vergt een goede uitleg om te begrijpen hoe het werkt. De nieuwe interface is makkelijker te begrijpen en heeft een aantal nieuwe eigenschappen zoals het vergroten en verkleinen van het venster in Playground of bij het databaseoverzicht kan de bij de verschillende velden de grootte worden aangepast.

Playground 2.0 zal ook beschikken over een dashboard. De gebruiker kan met behulp van de Widget Inspector, dit is een venster met een overzicht van allerlei widgets, een widget toevoegen aan zijn of haar dashboard. Deze widget kan dan over de dashboard naar verschillende posities worden gesleept dat aangegeven wordt door Playground.

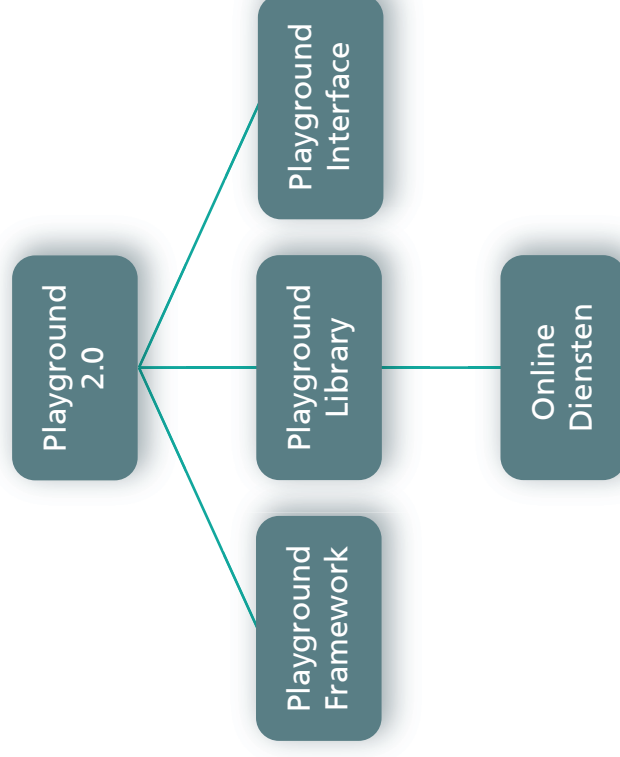
Het uiterlijk krijgt ook een nieuw lijke verf. De stijl van Playground wordt speels maar toch zakelijk. Dit wordt gedaan door gebruik te maken van een cartoon uiterlijk. Dit kan goed worden gezien bij de portal. Hier komen verschillende portal plaatjes afhankelijk van de weersomstandigheden.

Door de stijl van de portal op deze manier weer te geven ontstaat er voor de gebruiker het gevoel dat Playground leuk is om te gebruiken. Ook zal er hierdoor eerder weer gebruik van Playground gemaakt worden.

Conclusie Playground onderzoek

Ondanks dat de basis van Playground 2.0 een groot deel gelijk is aan Playground, is het een enorme verbetering in gebruikersgemak voor zowel gebruiker als programmeur. Met de nieuwe Playground kan nog eenvoudiger een website gebouwd worden en ziet deze website, door gebruik te maken van de laatste JavaScript technologieën, er nog beter uit.

De koppeling van externe API's geeft Playground 2.0 de mogelijkheid om gebruik te maken van deze diensten zonder zelf veel te programmeren. Bovendien hebben de klanten van Sense Interactive de mogelijkheid om ook gebruik te gaan maken van deze diensten in de dashboard van Playground 2.0. Dit geeft Playground een enorme meerwaarde.



4.2 Gekozen diensten

De keuze aan online diensten zijn talrijk. Een aantal van deze diensten zijn: Flickr, Twitter, Netvibes, picnik, Google Maps, Google Analytics, YouTube en LinkedIn. Omdat het project in een bepaalde tijd moet worden afgerond worden hier twee diensten van gekozen om te koppelen aan Playground.

Sense Interactive stelde voor om de diensten Twitter en Netvibes te kiezen voor dit project.

Twitter heeft een gemakkelijke dienst. De werking hiervan is eenvoudig wat het een ideale kandidaat maakt om als eerste te koppelen. De ervaringen die hieruit gehaald worden kan goed worden gebruikt voor de tweede dienst. Bovendien is Twitter enorm populair waardoor het een aanwinst is voor Playground.

Netvibes is een grotere online dienst waar veel programmeer code bij komt kijken. Het is dan ook een opdracht met een uitdaging. Dit komt vooral omdat Netvibes gebruik maakt van allerlei kleine applicaties. Deze applicaties kunnen ook als diensten worden gezien. Je hebt bijvoorbeeld een Weer applicatie. Dit laat het huidige weer zien met temperatuur, bewolking, etc.

Ook is Netvibes een enorm populaire dienst. Doordat deze dienst gebruik maakt van vele kleine applicaties kan, na de koppeling, Playground hier ook gebruik van maken. Hierdoor zal de functionaliteit van Playground enorm toenemen.

Door de eenvoud van Twitter en zijn populariteit is het een goede dienst om als eerste te koppelen. Netvibes zal een grotere uitdaging zijn maar levert Playground veel functionaliteit op. Het koppelen van Netvibes als tweede dienst is daarom een goede keus.

twitter



netvibes
(re)mix the web

4. Onderzoek

4.3 Twitter

Twitter is de eerste van de twee diensten die gekoppeld dient te worden. Om dit voor elkaar te krijgen zal eerst gekeken moeten worden wat Twitter voor een dienst is. Vervolgens wordt de API van Twitter bekeken. Hieruit zal naar boven komen wat er nodig is om Twitter te koppelen. Daarna worden de beperkingen van de API genoemd. Het design van Twitter komt vervolgens aan bod. Tot slot zal er een conclusie gevormd worden van het onderzoek.

Het idee achter Twitter

Jack Dorsey had het simpele idee om te willen weten wat zijn vrienden zoal aan het doen waren. Hiermee is het idee van Twitter ontstaan. Door middel van een kort bericht van maximaal 140 characters te sturen, via mobiel, instant messaging of het Internet, naar een online server van Twitter, worden de lezers van de gebruiker van Twitter op de hoogte gesteld met wat de gebruiker aan het doen is.

Twitter kan op deze manier ook gebruikt worden als een helpdesk waar klanten vragen kunnen stellen en direct antwoord kunnen ontvangen van de expert.

De eenvoud is wat Twitter zo populair maakt bij zijn gebruikers. Hierdoor zijn er ook vele soorten gebruikers zoals bedrijven die met hun klanten wil communiceren of beroemdheden die met hun fans willen praten.

Elk kort bericht, wat een gebruiker van Twitter heeft geplaatst, wordt een Tweet genoemd. Nadat de gebruiker een tekst heeft ingevoerd en op de update knop heeft geklikt, ontstaat er een nieuwe Tweet.

De Twitter API

Om Twitter beschikbaar te maken voor ontwikkelaars is er een API ontwikkeld. Met behulp van deze API kunnen programmeurs applicaties, websites, widgets en andere projecten ontwikkelen die interactie met Twitter hebben.

Om dit voor elkaar te krijgen maakt Twitter gebruik van de Twitter REST API. REST is een manier om webservices te creëren op basis van de bestaande en eenvoudige bouwstenen van het internet. Met behulp van REST kunnen gegevens, met een eenvoudige HTML aanroeping, opgevraagd en gemanipuleerd worden.

Als voorbeeld wordt er een computerzaak genomen die zijn producten beschikbaar wil stellen met behulp van REST. Door het aanroepen van de URL <http://www.computerzaak.nl/rest/moederborden?formaat=xml> (fictieve URL) wordt er een overzicht van zijn moederborden opgevraagd in de vorm van een XML. Nu kan een ontwikkelaar deze manier gebruiken om gegevens van de computerzaak op te halen en te gebruiken in zijn eigen applicatie, website etc.

Met behulp van de methodes in de Twitter REST API kan de kern van de Twitter data benaderd worden. De ontwikkelaar kan op deze manier alles, wat Twitter ter beschikking heeft gesteld via REST, benaderen en gebruiken in zijn eigen project.

Playground 2.0 maakt gebruik van het Zend Framework waarin gebruik gemaakt kan worden van verschillende Twitter methodes. Wat deze methodes voornamelijk inhouden is de aanvraag naar de Twitter REST API. Hiermee wordt er een object aangemaakt met de resultaten. Door gebruik te maken van deze Zend methodes hoeft de hele Twitter REST API aanvraag niet geprogrammeerd te worden.

Beperkingen

Twitter heeft een aantal beperkingen waar rekening mee gehouden moet worden.

Twitter heeft een limiet aan aanvragen per uur. Een gebruiker mag maar 150 aanvragen per uur doen via een API. Hierna zal Twitter een fout melding geven.

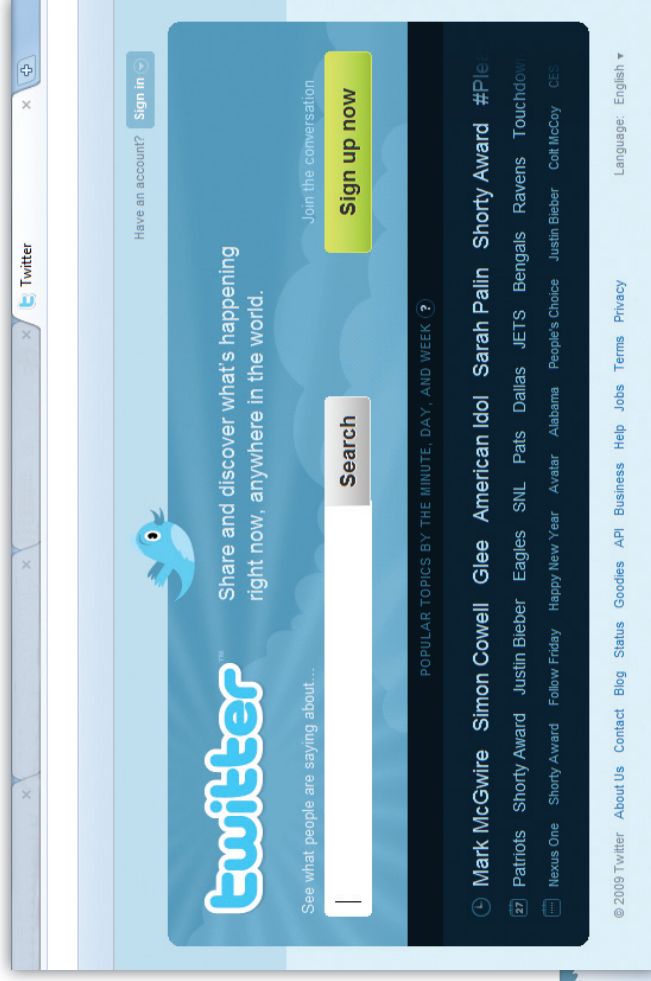
4. Onderzoek

Als er te veel gebruik gemaakt wordt van de API, meer dan 20.000 aanvragen per uur, kan het zijn dat Twitter het IP-adres blokt. Als er dus veel gebruikers zijn die gebruik willen maken van Twitter kan dit een probleem opleveren.

Omdat er gebruik gemaakt wordt van het Zend Framework blijft de functionaliteit afhankelijk van de ontwikkeling van Zend. Twitter is constant bezig om zijn functionaliteit van de API te vergroten. Hierop moet Zend vervolgens zijn Framework gaan bijwerken wat tijd kost. Playground zal dan een nieuwe versie van Zend moeten gaan gebruiken en het PHP script moeten uitbreiden met de nieuwe functionaliteiten van Twitter. Dit kost uiteraard tijd en geld voor Sense Interactive.

Het design

Het design van Twitter is gebaseerd op eenvoud. Er wordt gebruikt gemaakt van eenvoudige vormen en lichte kleuren.



Conclusie Twitter

Twitter heeft een API vrijgegeven die eenvoudig gebruikt kan worden door ontwikkelaars. De REST methode die ze hebben ontwikkeld zorgt voor eenvoudige weergave en gebruik van gegevens. Doordat Playground 2.0 gebruik maakt van het Zend Framework zal het koppelen van Twitter goed verlopen.

Twitter beschikt over veel functionaliteit wat constant wordt uitgebreid. Hierdoor zal ook het Zend Framework en de Playground Twitter API uitgebreid moeten worden als Playground over de laatste mogelijkheden wil beschikken. Het Zend Framework beschikt over de basis mogelijkheden van Twitter en het zal daarom niet nodig zijn om in korte tijd de Playground Twitter API aan te hoeven passen. Bovendien is het aan te raden om de eenvoud van Twitter ook binnen Playground aan te houden. De basis van de API, het ophalen van de Tweets van een gebruiker, het toevoegen van een Tweet en het reageren op een Tweet, is dan het enigste wat nodig is voor de Playground Twitter API.

Omdat de Twitter API een aantal beperkingen heeft moet hier rekening mee gehouden worden. De aanvraag mag maar 150 keer per uur gebeuren. Het is dan aan te raden om een vertraging in de Playground Twitter API te bouwen. Hiermee wordt de gebruiker beperkt in het ophalen van de laatste Tweets. Dit betekent dat ze niet direct kunnen zien wat iemand heeft geplaatst maar eerst een tijdje moeten wachten voordat de nieuwste Tweets worden opgehaald.

Twitter heeft van zichzelf de uitstraling van eenvoud en daarom moet dit ook worden doorgevoerd in Playground. Om dit voor elkaar te krijgen worden de meest gebruikte functies van Twitter in de applicatie opgenomen. Hierdoor zullen er een aantal geprogrammeerde methodes afvallen en niet in het design worden opgenomen.

4. Onderzoek

Zo heeft Twitter een overzicht van alle Tweets van de gebruiker en zijn vrienden en heeft Twitter ook een overzicht van alle favoriete Tweets. Deze favoriete Tweets zijn handig om te hebben maar is niet nodig om de kern van Twitter te laten zien. Als alle functionaliteiten van Twitter ook binnen Playground gebruikt zullen worden zal dit betekenen dat er tabjes of links moeten komen. Dit zorgt ervoor dat de Twitter applicatie onoverzichtelijk of te groot wordt.

Het voordeel van dit design is dat de informatiehoeveelheid beperkt blijft tot het meest relevante. De gebruiker zal in Playground beschikken over de twintig laatste Tweets van zichzelf en zijn of haar vrienden. Ook heeft de gebruiker de mogelijkheid om een Tweet toe te voegen of te beantwoorden en zijn of haar vrienden of volgers te bekijken.

Om gebruik te maken van de andere functies van Twitter moet de gebruiker inloggen bij Twitter zelf. De Twitter API zal echter wel beschikken over de methodes zodat Sense Interactive deze later alsnog kan gebruiken voor Playground of voor websites van klanten.

4.4 Netvibes

De tweede dienst die gekoppeld wordt is Netvibes. Hier zal ook eerst worden gekeken wat de dienst inhoudt. Vervolgens wordt de API bekeken en komt het design aan bod. Tot slot komt er een conclusie.

Wat is Netvibes?

Netvibes is een gratis website die in staat is om een eigen beleving van het web te creëren. Alles wat een gebruiker belangrijk vindt in de online wereld zoals nieuws, blogs, het weer, email, zoekengine, video's, foto's etc. kan op een eigen persoonlijke pagina worden samengebracht. Deze pagina wordt vervolgens opgebouwd uit widgets.

De gebruiker heeft ook de mogelijkheid om een publieke pagina aan te maken die toegankelijk is voor vrienden. Op deze manier kan de gebruiker zijn meest populaire widgets met zijn of haar vrienden delen. Deze vrienden kunnen dan met één klik de widget aan hun eigen pagina toevoegen.

De Netvibes API

Netvibes maakt zijn widgets beschikbaar als Universal Widget API (UWA). Met behulp van deze API is het eenvoudig om een widget te maken voor omgevingen zoals Netvibes, iGoogle, Vista, Apple Dashboard, Live.com, Opera, etc.

Verder heeft Netvibes nog twee API's waar gebruik van gemaakt kan worden.

De REST API en de Ecosystem API.

De REST API is diegene waar Playground in geïnteresseerd is. Deze wordt gebruikt voor het opvragen van de inhoud van een Netvibes pagina. Hiermee kan Netvibes binnen Playground gebruikt worden.

Het Ecosystem API is voor het ophalen van data van Netvibes Ecosystem. Dit is een database van Widgets. Verder onderzoek naar deze API is niet nodig aangezien hier de interesse niet ligt.

De mogelijkheden van de Netvibes REST API zijn als volgt:

- opvragen van alle tabs en widgets van de gebruikers Universe
- opvragen van enkele tabs en hun widgets
- opvragen van één tab en zijn widgets
- opvragen van één widget
- opvragen van een gebruikers publieke tijdlijn
- zoeken in een gebruikers publieke tijdlijn

Een aanvraag aan Netvibes gaat via de URL: <http://rest.netvibes.com>. Op deze manier worden gegevens in de formaten XML of JavaScript Object Notation (JSON) opgevraagd. Voor de tijdlijn zijn ook de formaten RSS en Atom beschikbaar. De standaard formaat is XML.

Om gebruik te maken van de UWA heeft Netvibes PHP en JavaScripts scripts vrijgegeven. Met behulp van de PHP scripts kunnen gegevens van een widget worden opgehaald en kunnen deze gegevens gebruikt worden om een widget op te bouwen. Met de JavaScript scripts wordt de kernfunctionaliteit van elke widget aangestuurd.

De UWA kan gebruikt worden in een inline frame (iframe). Dit is een webpagina in een webpagina. Een iframe heeft zijn eigen omgeving en staat daarmee los van de pagina waarin deze iframe zich bevindt.

Met deze modus kan de widget ingeladen worden op een website. Alle benodigde scripts worden ingeladen en de widget is klaar voor gebruik.

4. Onderzoek

Het gebruik van een UWA op een pagina in inline modus wordt door Netvibes afgeraden vanwege veiligheidsredenen. Met inline mode wordt bedoeld dat de widget in de omgeving van een pagina wordt getoond. Hiermee kunnen alle acties door de gebruiker op één pagina worden afgehandeld.

Beperkingen

Sense Interactive had graag dat de Netvibes widgets inline getoond worden. Deze manier geeft echter problemen met de JavaScripts die Netvibes voor zijn widgets heeft geschreven.

Doordat Netvibes geen ondersteuning biedt voor inline modus is het moeilijker voor Playground om de eigenschappen van de widget te bepalen. Een gebruiker kan bijvoorbeeld de titel van de widget veranderen waarna Playground deze op moet slaan. Omdat de widgets niet in inline modus kunnen worden geladen moet er een andere oplossing gevonden worden voor het afvangen van deze wijziging.

Netvibes geeft ook geen overzicht van widgets weer. Om toch een overzicht van widgets te krijgen moet eerst de Netvibes Universe (de Netvibes pagina) van een gebruiker opgehaald worden om zo de widgets UWA Skeleton id op te vragen. Dit is een unieke naam voor de widget bestaande uit letters en cijfers. Met behulp van deze id kan de widget opgebouwd worden. Om op deze manier de widgets op te halen moet er een publieke Netvibes pagina aangemaakt worden met widgets.

Het design

Elke widget heeft zijn eigen stylesheets en daarmee zijn eigen ontwerp. Playground heeft een widget in zijn eigen omgeving te slepen waarmee de widget een kader met een titel krijgt. De widget heeft dan zijn eigen design waardoor dit niet door Sense Interactive ontworpen hoeft te worden.

Conclusie Netvibes

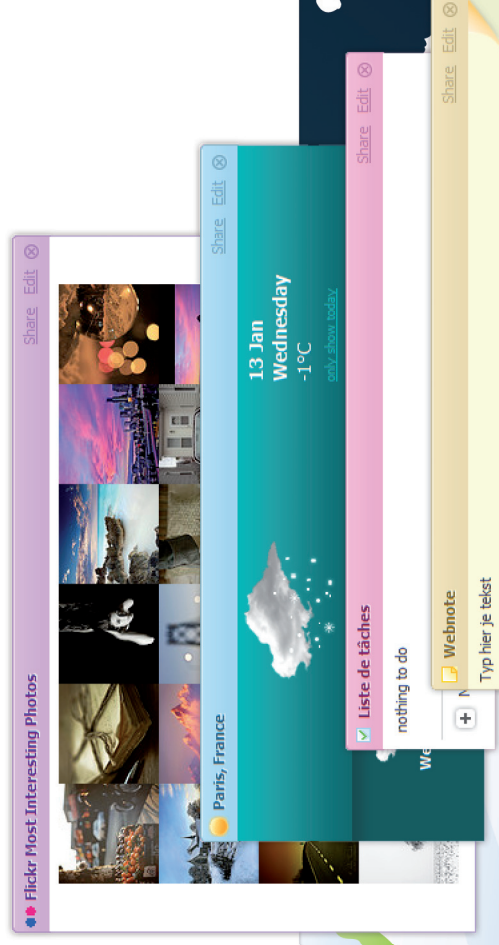
Netvibes beschikt over tal van widgets. Het is daarom niet zeker te zeggen of alle widgets in Playground 2.0 zullen werken. Hierdoor zal er gekeken moeten worden of de widget goed werkt voordat deze in de live versie van Playground 2.0 gebruikt gaat worden.

Om een widget te tonen in Playground zal er gebruik gemaakt moeten worden van de PHP scripts die door Netvibes zijn ontwikkeld. Ook zal de JavaScript van Netvibes gebruikt moeten worden voor het aansturen van zijn widgets.

Het gebruik van Netvibes zijn UWA zal met een iframe moeten. Dit zal conflicten voorkomen zodat de widget goed zal werken.

Ook zal er een overzicht met Skeleton id's gemaakt moeten worden. Met deze id's kunnen vervolgens de widgets gebouwd worden. Om deze id's te generen is een aanvraag naar de Netvibes REST API nodig.

Als de gebruiker van Playground een widget heeft gekozen zal zijn Skeleton id opgeslagen moeten worden in een database. Bij een volgend bezoek wordt dan de id opgevraagd zodat de juiste widget voor de gebruiker zichtbaar wordt. De afgevangen wijzigingen moeten ook in een database worden opgeslagen. Dit zorgt ervoor dat de gebruiker zijn wijzigingen van een widget bewaard blijven.



5. Koppeling met Playground

Nadat Playground, Twitter en Netvibes zijn onderzocht kan aan de koppeling worden begonnen. Voor deze koppeling is ook kennis vereist van een aantal programmeertalen. Zo moet er kennis zijn van: PHP, JavaScript, XML, Hyper Text Markup Language (HTML) en Cascading Style Sheets (CSS). Deze programmeertalen maken het mogelijk om een applicatie of website te maken.

Omdat deze programmeertalen niet bij iedereen bekend zijn volgt een korte uitleg wat deze talen inhouden.

PHP zorgt voor de verwerking van de gebruikersinvoer aan de server kant. Gegevens versturen naar de server of gegevens uit een database ophalen en laten tonen aan de gebruiker zijn een aantal van de mogelijkheden van PHP.

JavaScript is een programmeertaal wat aan de gebruikerskant wordt uitgevoerd. Hiermee kan bijvoorbeeld animatie gebruikt worden in de website. Uitschuivende menu's en het direct controleren van de invoer van een gebruiker zijn een aantal voorbeelden. Tegenwoordig kan ook met behulp van JavaScript gegevens worden verstuurd naar de server. PHP kan deze aanvraag dan afhandelen.

XML is een manier om gegevens te laten zien in de vorm van parent en child. XML is voor zowel mens als machine gemakkelijk te lezen en te begrijpen. Een voorbeeld van een XML bestand:

```
<xml>
<boeken>
<boek>
  <titel>De Wereld Rond</titel>
  <beschrijving>Dit boek gaat over...</beschrijving>
</boek>
...
</boeken>
</xml>
```

Verder zorgen HTML en CSS voor de weergaven van de website aan de gebruiker. HTML zorgt in dit geval voor de lay-out en inhoud van de website. CSS zorgt weer voor de opmaak van de website.

Voor de Playground Twitter API wordt gebruik gemaakt van PHP en HTML. Het PHP script zorgt ervoor dat de Tweets worden getoond. Dit script werkt vervolgens met het Zend Framework samen om de Twitter gegevens op te halen. Playground werkt met XML om gegevens te tonen. Hierdoor moet het resultaat van de Twitter gegevens, die met Zend opgehaald worden, omgezet worden naar een XML formaat.

Voordat deze gegevens opgehaald kunnen worden is er eerst een gebruikers account aangemaakt bij Twitter.com. Ook werd hier een OAuth sleutel aangevraagd. Dit werd gedaan om de API te voorzien van een OAuth inlog methode. Met OAuth kan de gebruiker inloggen bij zijn eigen online dienst, in dit geval is dat Twitter, zonder dat de gegevens bij een derde partij bekend hoeven te zijn.

Om het gevoel van veiligheid van de gebruiker te bevorderen wordt het PHP script met een OAuth methode uitgebreid.

Omdat het aantal aanvragen naar Twitter maximaal 150 per uur is, is er een vertraging functionaliteit aan het script toegevoegd. Dit zorgt ervoor dat er een nieuwe aanvraag één keer per minuut wordt toegestaan. Wordt er eerder een aanvraag gedaan, dan worden de eerdere opgeslagen resultaten getoond.

5. Koppeling met Playground

Het design van Twitter heeft veel weg van het design van Playground. De eenvoud komt daarom ook terug in het dashboard. De gebruiker heeft op het dashboard beperkte mogelijkheden. Hij of zij kan Tweets bekijken en een nieuwe Tweet toevoegen en op een Tweet antwoorden. Doordat de mogelijkheden van Twitter op het dashboard beperkt blijven is het voor de klanten van Sense Interactive overzichtelijk en lijdt het niet te veel af van de werkzaamheden die ze met Playground moeten verrichten.

De Playground Netvibes API maakt gebruik van Netvibes PHP scripts die ze samen hebben gevoegd met als naam Exposition Library. Deze scripts zijn in staat om gegevens van een gebruiker zijn pagina op te halen. Vervolgens kan ook met deze scripts en de gegevens die opgehaald zijn een widget gebouwd worden.

Doordat een widget in een iframe opgebouwd wordt staan de JavaScripts, waar deze widget gebruik van maakt, los van de JavaScripts die Playground gebruikt. Dit zorgt ervoor dat er geen conflicten ontstaan en dat de widget in zijn eigen veilige omgeving getoond wordt.

Meer details van de koppeling is terug te vinden in de bijlage:

III. De koppeling met Playground 2.0.



playground

6. Conclusie(s) en aanbeveling(en)

Met de Twitter API is Playground 2.0 voorzien van de functionaliteiten die Twitter zijn gebruikers biedt. Het Zend Framework zorgt voor een goede basis waardoor uitbreidingen, door toekomstige verbeteringen en of toevoegingen van Twitter, eenvoudig aan de API toe te voegen zijn. Een vereiste hiervan is dat het Zend Framework ook up to date wordt gehouden zodat Zend ook over de laatste technieken beschikt.

Wel is aan te raden om geen gebruik te gaan maken van alle mogelijkheden van Twitter op de Playground 2.0 dashboard. Dit zorgt ervoor dat het gebruik van Twitter minimaal is zodat de aanvragen naar de Twitter REST API zo laag mogelijk blijft. Hiermee wordt voorkomen dat Playground 2.0 op de zwarte lijst komt of zelfs geblokkeerd wordt bij Twitter. Bovendien is te veel informatie in een widget verwarrend. De eenvoud is wat het een widget tot een widget maakt.

Omdat de Netvibes widgets door externe programmeurs worden gemaakt is het lastiger om deze binnen Playground te krijgen. Om dit voor elkaar te krijgen zal er naar elke widget gekeken moeten worden of deze volledig binnen Playground kan werken.

De widgets van Netvibes worden geladen met een unieke id. Deze id wordt via de REST API van Netvibes geladen. Deze actie is echter alleen nodig als de gebruiker een nieuwe widget wil toevoegen. Als Netvibes problemen heeft met de REST API kunnen er geen widgets geladen worden waardoor de gebruiker geen nieuwe widgets kan toevoegen. Hier moet rekening mee gehouden worden.

Ook zullen de niet werkende widgets uit het overzicht, van de resultaten van de REST aanvraag, uitgefilterd moeten worden. Anders zou het mogelijk zijn dat een gebruiker van Playground een niet werkende widget op zijn dashboard sleept. Hierdoor ontstaat verwarring wat niet ten goede komt voor het gebruik van Playground.

Ook is de Netvibes UWA constant in ontwikkeling wat betekent dat dit ook in de gaten gehouden dient te worden. Als er een grote verandering komt in de Netvibes UWA kan dit nadelig zijn voor Playground. Het is daarom verstandig om Playground de mogelijkheid te geven om het gebruik van de Netvibes widgets stop te zetten. Hierdoor zal een gebruiker niet in staat zijn om een niet of slecht werkende widget op zijn dashboard te gebruiken. Door gebruik te maken van een melding op het scherm kunnen de gebruikers op de hoogte worden gesteld. Dit zal veel verwarring voorkomen en ziet er professioneler uit dan widgets die niet werken.

Begin

In het begin stelde Lucienne voor om een PID document op te stellen. Hiermee heb ik veel problemen gehad voor wat betreft de invulling van het document. Ik kreeg het niet voor elkaar om de juiste informatie weer te geven waardoor ik het document vaak terug kreeg van Lucienne ter verbetering.

Omdat het me niet lukte om het PID document af te ronden heb ik besloten om een Plan van Aanpak te maken. Ik had eerder een Plan van Aanpak ook in mijn oriënterende stage gemaakt wat goedgekeurd was en heb deze dan ook als leiddraad gebruikt.

Het Plan van Aanpak is voor mij een goede beslissing geweest omdat het veel beter in elkaar stak als het PID document.

Onderzoek

Het onderzoek naar Twitter is goed gegaan. Er was veel informatie beschikbaar en de ontwikkelaars van Twitter en het Zend Framework hadden voor goede documentatie gezorgd.

Ik had alleen niet verwacht dat ik veel tijd kwijt was om ook de PHP object georiënteerde manier te moeten bestuderen. Ook moest ik veel tijd steken in het bekijken van JavaScript. Ik had enige ervaring met deze programmeertalen wat mij ook naar deze opdracht toe stuurde. Echter had ik niet verwacht dat het zo'n grote uitdaging was.

Het Netvibes onderzoek verliep moeizaam. De documentatie was beperkt en ik wist niet goed waarmee ik moest beginnen. Jasper had iets gelezen over Netvibes zijn Exposition Server. Hiermee zou het mogelijk zijn om widgets op te bouwen. Met deze informatie kon ik verder met het onderzoek.

Sense Interactive wilde Netvibes het liefst inline hebben waardoor ik geen iframes kon gebruiken. Met deze manier ontstonden er veel problemen met de JavaScript van Playground en Netvibes. Hierdoor is besloten om gebruik te gaan maken van iframes.

Integratie

Nadat ik had gewerkt aan het bouwen van de Playground Twitter API, werd duidelijk dat het niet geheel volgens het Playground principe was. Hierdoor heb ik samen met Paul van der Avoort een opzet gemaakt. Met deze opzet heb ik volgens het principe van Playground met aanvraag en reactie een voorbeeld site opgezet. Hier kan Sense Interactive alle mogelijkheden van Twitter bekijken en is ook meteen bekend wat de aanvraag is.

Veel van de methodes die ik al had gebouwd kon ik hergebruiken hierin waardoor het zeker geen verloren tijd was geweest.

Netvibes gaf veel problemen bij het integreren. Hierbij waren voornamelijk JavaScript conflicten. Ik heb in eerste instantie hier omheen proberen te werken; wat niet lukte. Hierna is besloten met Sense Interactive dat het met iframes opgelost moest worden. In eerste opzicht moesten de widgets in één iframe geladen worden. Dit is echter niet gelukt omdat het weer een conflict gaf. Dit keer met de div waar de widget ingeladen wordt. De widgets overschreven elkaar steeds.

Achteraf was het door mij niet goed begrepen. Iedere widget moest in zijn eigen iframe geladen worden. Ik kwam bij de oplossing nadat ik het probleem goed had bekeken en de documentatie nog eens goed doorlas op de iframe methode.

Deze oplossing werkte goed en de widgets konden geladen worden. Hierna kwam het volgende probleem. De Netvibes REST API deed het niet meer. Ik heb hierop Netvibes een mail gestuurd waarop ik een antwoord kreeg waarmee ik niet verder kon.

Evaluatie

Om toch te laten zien dat de API werkt heb ik een UWA id hard in de code moeten zetten. Hier had ik er maar één van omdat ik er niet van uit ging dat ik andere UWA id's nodig zou hebben. De Netvibes REST API deed het immers.

Ondanks dit probleem heb ik de to do widget van Netvibes werkend gekregen. Voor demonstratie doeleinde zal deze widget gebruikt worden.

Om te demonstrenen hoe de twee diensten in Playground gaan werken heb ik een test omgeving opgezet. Deze omgeving is de laatste versie van Playground 2.0 die ik van Jasper Ancher mag gebruiken voor mijn demo. Samen met Tom Lamoen heb ik deze omgeving opgezet. Hierin wordt de Playground 2.0 dashboard en de werking van de API's gesimuleerd.

MochaUI

<http://mochaui.com/>

PrototypeJS

<http://www.prototypejs.org/>

Zend

<http://framework.zend.com/about/overview/>

http://nl.wikipedia.org/wiki/Zend_Framework

MooTools

<http://mootools.net/>

Wikipedia

<http://nl.wikipedia.org/wiki/Hoofdpagina>

AJAX

<http://www.zdnet.nl/itprofessional/55859/ajax-veelbelovende-webtechnologie-of-een-hype-/>

REST

<http://www.xfront.com/REST-Web-Services.html>

<http://www.whitehorses.nl/whitebooks/2005/webservices-rest> http://en.wikipedia.org/wiki/Representational_State_Transfer#Concept

Twitter

<http://www.ibm.com/developerworks/xml/library/x-twitterREST/>

<http://www.contentgirls.nl/2009/02/twitter-populair-wat-nu/>

OAuth

<http://www.software-innovators.nl/2009/05/19/twitter-veilig-met-oauth/>

<http://hueniverse.com/oauth/>

<http://white-box.us/2009/10/03/zend-twitter-and-oauth-made-easy/>

cURL

<http://en.wikipedia.org/wiki/CURL>

XML

http://nl.wikipedia.org/wiki/Extensible_Markup_Language

Document Object Model (DOM)

http://nl.wikipedia.org/wiki/Document_Object_Model

Bijlagen

- I. Plan van Aanpak
- II. Opdracht Sense Interactive
- III. De koppeling met Playground 2.0
- IV. Zend_Service_Twitter methodes
- V. cURL
- VI. OAuth
- VII. PrototypeJS

