

I. Plan van Aanpak

Documentnaam:	Plan van Aanpak
Uitgave:	Definitief
Datum afronding:	11 november 2009
Plaats:	Uden
Auteur:	Heesakkers, Brett
Organisatie	Sense Interactive
Opdrachtgever:	Henrik van Duijnhoven
Versie:	0.6

Versiebeheer

Versie	Status	Opmerking
0.1	Eerste opzet PvA	In de eerste week van de afstudeerstage is het PvA opgezet.
0.2	Uitwerken PvA	Na afkeuring van het PID document is er besloten om een PvA te maken.
0.3	Revisie na feedback Paul van der Avoort	Het PvA is op een aantal punten aangepast en verbeterd na de feedback van Paul van der Avoort. Een medewerker bij Sense Interactive
0.4	Nacontrole indeling hoofdstukken	Nadat Luciënne feedback had gegeven over de hoofdstukindeling is het PvA opnieuw doorgenomen.
0.5	Feedback Luciënne verwerkt	Nadat het PvA volledig was, was het doorgestuurd en is er feedback op terug gekomen.
0.6	Laatste verbeteringen in het PvA met feedback van Paul van der Avoort	Met de feedback van Luciënne is het PvA aangepast en heeft Paul het document nog een keer bekeken voor een laatste feedback.

Inhoudsopgave

Inleiding	4
Definities en afkortingen	5
Hoofdstuk 1 :: Achtergronden	6
Hoofdstuk 2 :: Projectopdracht	7
Inleiding	7
Opdracht	7
Beginsituatie	7
Doelstelling	7
Hoofdstuk 3 :: Projectactiviteiten	8
Initiële fase	8
Onderzoeksfase	8
Realisatiefase	8
Test en evaluatiefase	9
Afrondingsfase	9
Overdrachtsfase	9
Slotfase	9
Hoofdstuk 4 :: Projectgrenzen	10
Afbakening	10
Randvoorwaarde	10
Hoofdstuk 5 :: De producten	11
Opsomming	11
Hoofdstuk 6 :: Kwaliteit	12
Waarborging	12
Controles	12
Terugkoppelen	12
Normen, techniek en methodes	13
Hoofdstuk 7 :: Projectorganisatie	14
Inleiding	14
Informatiematrix	15
Archivering	15
Hoofdstuk 8 :: Planning	16
Fase planning	17
Mijlpalen	18
Hoofdstuk 9 :: Kosten en baten	19
Kosten	19
Baten	19
Hoofdstuk 10 :: Risico's	20
Intern en extern	20
Risicoanalyse	20

Inleiding

Als afstudeerstudent ben ik bij Sense Interactive terecht gekomen om aan een innovatieve opdracht te werken. De inhoud van dit project wordt verder in het document beschreven.

Voordat er aan het project gewerkt kan worden moet er eerst duidelijk beschreven worden wat de opdracht inhoud en hoe de opdracht wordt aangepakt. Hiervoor wordt dit document geschreven.

Om te beginnen wordt de achtergrond van het bedrijf beschreven. Vervolgens wordt de opdracht duidelijk omschreven, de doelstelling wordt geformuleerd, de projectfases en projectgrenzen worden opgesteld, de uiteindelijke producten worden opgesomd, de kwaliteit wordt besproken, de projectorganisatie wordt beschreven, de planning wordt gemaakt en tot slot komen de kosten en baten en de risicoanalyse aan bod.

Definities en afkortingen

Definities

Definitie	Betekenis
Events	Gebeurtenissen die plaatsvinden in de programmeer omgeving. Zoals een muisklik of als de browser klaar is met het laden van gegevens.
JavaScript	JavaScript is een objectgeoriënteerde programmeertaal die veel gebruikt wordt om internetpagina's interactief te maken en webapplicaties te ontwikkelen.
Native XML	
Netvibes	Netvibes (gelanceerd op 15 september 2005) is een webapplicatie van Franse komaf die het mogelijk maakt een persoonlijke startpagina te creëren.
Playground	Door Sense Interactive ontwikkeld framework om website te ontwikkelen ten behoeven van content management en online diensten koppeling.
Twitter	Twitter is een internetdienst waarbij gebruikers korte berichtjes publiceren.
Oauth	OAuth is een open standaard voor het autoriseren van API aanroepen.
PGX Framework	Dit is de engine van Playground. Met behulp van dit framework kan gebruik gemaakt worden van alle mogelijkheden die Playground biedt.

Afkortingen

Afkorting	Betekenis
AJAX	Asynchronous Javascript And XML
API	Application Programming Interface
CMS	Content Management Systeem
HTML	Hyper Text Markup Language
JSON	JavaScript Object Notation
PHP	PHP: Hypertext Preprocessor
XML	Extensible Markup Language

Hoofdstuk 1 :: Achtergronden

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 de door hun 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.

Playground

Playground is een framework waarop websites gemaakt kunnen worden. Het beschikt over een CMS(Content Management Systeem) voor klanten en een eenvoudige programmeer structuur voor de werknemers van Sense Interactive.

Omdat Playground niet optimaal volgens de visie van Sense Interactive gebouwd is wordt er gewerkt aan een nieuwe versie van Playground. Deze nieuwe versie, genaamd Playground 2.0, zal gebruik maken van een verbeterde interface, meerdere online diensten, een native XML(Extensible Markup Language) database en een communicatielaag die zorgt voor de afhandeling van events.

Online diensten

In deze wereld van snelle ontwikkelingen zijn er steeds meer online diensten beschikbaar. Diensten zoals Twitter, GoogleMaps en Youtube hebben hun API (Application Programming Interface) beschikbaar gemaakt zodat ontwikkelaars deze in hun programma's kunnen gebruiken.

Om bij de koplopers van het aanbeiden van websiteontwikkeling te behoren, zal het nieuwe Playground 2.0 hier intensief gebruik van gaan maken. Dit is de hoofdreden waarom de opdrachtgever deze afstudeeropdracht heeft.

Het project

Bij dit project is de ICT en Design afdeling betrokken. De ICT afdeling ondersteund bij de koppeling van de diensten in Playground en de Design afdeling helpt bij het visualiseren van de dienst binnen in Playground.

Het Playground 2.0 project is in een ontwikkel fase. Hierdoor werken deze twee projecten nauw met elkaar samen.

Hoofdstuk 2 :: Projectopdracht

Inleiding

Sense Interactive is bezig met de ontwikkeling van een nieuwe versie van Playground. In deze versie willen ze de nieuwste API diensten koppelen zodat Playground een zo groot mogelijk aanbod heeft. Twitter en Netvibes zijn twee van deze diensten die ze willen koppelen. Hiervoor is een project door een afstudeerder gestart met als vraag:

- Hoe kan Playground gebruik maken van de online diensten Twitter en Netvibes?

Om deze vraag te beantwoorden zijn een aantal deelvragen ontstaan.

- Wat is Playground?
- Wat zijn Twitter en Netvibes voor diensten?
- Hoe kunnen Twitter en Netvibes gekoppeld worden met Playground?
- Op welke manier kan de gebruiker veilig bij deze diensten inloggen?

Opdracht

De opdracht bestaat uit het onderzoeken en integreren van de online diensten Twitter en Netvibes in Playground 2.0. Hierdoor wordt Playground voorzien van de nieuwste online diensten. Om dit voor elkaar te krijgen is er onderzoek vereist naar de werking van Playground.

Het tweede onderzoeksdoel is Twitter en Netvibes. Dit onderzoek zorgt ervoor dat de integratie met Playground zo goed mogelijk verloopt.

Om meer inzicht te krijgen in de werking van Playground en om zo ook het onderzoek te ondersteunen wordt er een tijd meegewerkt aan het ontwikkelen van websites via de Playground architectuur.

Beginsituatie

Het huidige Playground beschikt over online diensten. Deze diensten worden aan het PGX Framework gekoppeld en zijn geen op zichzelf staande libraries die aangeroepen kunnen worden. De kracht van Playground is de mogelijkheid om de allernieuwste Internet technologieën samen te brengen in één omgeving. Twee van die applicaties zijn Twitter en Netvibes.

Zij hebben hun online dienst vrijgegeven in de vorm van een API, Playground zal hiervan gebruik gaan maken.

Doelstelling

In 20 weken Playground te voorzien van de online diensten Twitter en Netvibes zodat de klanten van Sense Interactive hier gebruik van kunnen maken en eventueel nieuwe klanten enthousiast te maken voor Playground.

Hoofdstuk 3 :: Projectactiviteiten

Initiëlefase

Om een overzicht van het project te krijgen wordt in deze fase het Plan van Aanpak gemaakt.

Onderzoeksfase

Deze fase bestaat uit het onderzoek naar [Playground](#), Playground 2.0, [Twitter](#), [Netvibes](#) en [OAuth](#).

Bij het onderzoek naar Playground wordt er onderzocht wat Playground inhoud. Ook wordt er gekeken naar welke programmeer technieken en architectuur er zoal gebruikt worden.

Vervolgens wordt Twitter onderzocht. Er wordt gekeken naar wat Twitter inhoud en wat voor meerwaarde het voor Sense Interactive heeft. Daarna wordt gekeken hoe Twitter aan Playground gekoppeld kan worden en wat daar allemaal voor nodig is.

Om gebruik te kunnen maken van Twitter is het nodig om de gebruiker in te laten loggen. Hiermee haalt hij of zij de Twitter gegevens op en kan de status van zijn of haar Twitter geüpdate worden. Dit wordt gedaan met OAuth. Om hiervan gebruik te maken wordt onderzocht wat het inhoud en hoe er gebruik van gemaakt wordt.

Na de koppeling van Twitter zal Netvibes worden onderzocht. Er wordt wederom gekeken wat het inhoud en wat voor meerwaarde het voor Sense Interactive heeft. Hierna wordt onderzocht hoe Netvibes gekoppeld kan worden.

Realisatiefase

Nadat het onderzoek van Playground en Twitter is afgerond wordt Twitter aan Playground gekoppeld. Hierbij wordt er gebruik gemaakt van PHP(PHP: Hypertext Preprocessor), [JavaScript](#), de [API](#) van Twitter en het [PGX Framework](#). Om de communicatie tussen deze technieken te voorzien wordt er gebruik gemaakt van onder andere AJAX(Asynchronous Javascript And XML), JSON(JavaScript Object Notation) en XML(Extensible Markup Language).

Ook wordt er gekeken naar de visualisatie van Twitter binnen Playground. De benodigde vormgeving wordt aangepast zodat het binnen de stijl van Playground blijft en het voor de gebruiker eenvoudig te begrijpen is.

In deze fase wordt, nadat het onderzoek van Netvibes is afgerond, Netvibes gekoppeld aan Playground. Hierbij wordt ook rekening gehouden met de evaluatie van Twitter zodat de ervaring van het koppelen van Twitter meegenomen kan worden tijdens deze fase. Ook hier wordt er gekeken naar de vormgeving van Netvibes binnen Playground.

Test en evaluatiefase

Nadat de Twitter koppeling is afgerond wordt er gekeken of alles goed functioneert binnen Playground. Vervolgens wordt er geëvalueerd wat goed en minder goed ging tijdens het proces. Ook wordt in deze fase de scriptie geschreven.

Deze fase wordt ook gebruikt na het koppelen van Netvibes. Er wordt gecontroleerd of Netvibes goed functioneert in Playground.

Afrondingfase

Deze fase dient voor het afronden van de scriptie en eventuele andere documenten.

Overdrachtsfase

In deze fase wordt het project overgedragen aan Sense Interactive. Ze ontvangen een kopie van de scriptie en benodigde documentatie voor het gebruik van Twitter en Netvibes in Playground.

De scriptie wordt ook aan de Fontys overhandigd ter evaluatie van het project.

Slotfase

Tussen 20 en 26 januari vindt de afstudeerzitting plaats waarin een presentatie wordt gegeven over het project en wordt de afstudeerder beoordeeld op zijn afstudeerstage.

Hoofdstuk 4 :: Projectgrenzen

Afbakening

De stage periode duurt 20 weken en zal om 15-1-2010 eindigen. Het project houdt zich bezig met het onderzoek van [Playground](#), [Twitter](#), [Netvibes](#) en [OAuth](#). Verder zal het integreren van Twitter en Netvibes onderdeel zijn van het project.

Na oplevering van de producten zullen er geen verdere werkzaamheden aan verricht worden, wat betreft onderhoud of herstel. Uiteraard wordt het project en de resultaten overgedragen aan het bedrijf.

Na 20 weken zal de afstudeerder geen verplichtingen meer hebben tegenover Sense Interactive wat betreft de stage.

Randvoorwaarde

Om het slagen van het project te garanderen zijn de volgende randvoorwaarde van toepassing.

- Er moet voldoende kennis van de benodigde programmeer talen aanwezig zijn.
- De benodigde software is aanwezig.
- Het project is in 20 weken af te ronden.
- De begeleiding vanuit school is aanwezig.

Hoofdstuk 5 :: De producten

Aan het eind van de afstudeerperiode worden een aantal producten opgeleverd. Een groot onderdeel hiervan is de beschrijving van het project en de projectaanpak in de vorm van een scriptie.

Hieronder is een opsomming te vinden van de producten die opgeleverd worden.

Opsomming

- Plan van Aanpak
- Blog (iedere vrijdag voor 15:00 met update project voortgang)
- Onderzoek
 - Onderzoek Playground
 - Onderzoek Twitter
 - Onderzoek Netvibes
 - Onderzoek OAuth
- Gekoppelde diensten aan Playground
 - Gekoppelde Twitter
 - Gekoppelde Netvibes
- Scriptie
- Eindpresentatie

Hoofdstuk 6 :: Kwaliteit

Waarborging

De bedrijfsbegeleiders(s) bekijken of de producten, Twitter en Netvibes, volgens de eisen van Sense Interactive zijn.

De scriptie wordt doorgenomen door de bedrijfsbegeleider(s) en docentbegeleider. Deze geven feedback over de hoofdstukken en de informatie die hier in is verwerkt.

De eindverantwoording voor de producten ligt bij de afstudeerder.

Controles

Om ervoor te zorgen dat de uiteindelijke producten goed afgerond worden, zullen er een aantal controles uitgevoerd worden.

- De Blog wordt iedere week door de docentbegeleider bekeken.
- Het PvA wordt gekeurd door de bedrijfs en docentbegeleider.
- Tijdens het integreren van Twitter en Netvibes vele controles uitgevoerd worden. Hierbij wordt gekeken naar browser ondersteuning voor IE7, IE8, Firefox, Chrome, Safari. Ook wordt de werking binnen en buiten de Playground architectuur gecontroleerd.
- De programmeer code wordt geëvalueerd.

Terugkoppelen

Na het integreren van een dienst vindt er een evaluatie met de opdrachtgever plaats over dit onderdeel. Hierbij wordt gekeken wat goed en fout ging en wat beter kon en of het onderdeel afgerond is. Ook geven de begeleiders hun feedback over het PvA en de scriptie.

Normen, techniek en methodes

Normen

- Elke vrijdag voor 15:00 wordt een Blog bijgehouden over de voortgang.
- Het beheer van Twitter en Netvibes in Playground gaat uit van een end-user.
- Het PvA wordt doorgenomen door de begeleiders waarop feedback teruggekoppeld wordt.

Techniek

- Twitter en Netvibes wordt gebruikt in het Playground framework.
- De volgende programmeer talen worden gebruikt in en rondom Playground:
 - PHP
 - JavaScript
 - XML
 - HTML

Methodes

Om het projectverloop te bevorderen worden er een aantal methodes toegepast.

- Documenten worden in een concept versie opgezet. Hierna worden deze documenten gecontroleerd waarna ze verbeterd worden, om uiteindelijk tot een definitieve versie te komen.
- Er wordt een PvA gemaakt om het project te beschrijven en ondersteunen.
- Om de kennis te vergroten worden er een aantal scenario's ontwikkeld zodat het uiteindelijke product het beste resultaat krijgt.
- Om de interface van de online diensten te ontwerpen wordt er gebruik gemaakt van paper prototyping.
- Om het leerproces te bevorderen worden er peer-reviews afgenomen en vinden er evaluaties plaats over de opgeleverde producten.

Hoofdstuk 7 :: Projectorganisatie

Inleiding

In dit hoofdstuk staan alle gesprekken, afspraken en documenten die gemaakt en opgeleverd worden, en welke personen deze kunnen verwachten.

Tijdens de afstudeerstage worden er een aantal documenten opgeleverd. Het belangrijkste document van de afstudeerstage is de scriptie. Deze zal door de student individueel worden gemaakt. Het maken van dit verslag gebeurt in 3 aparte stappen:

- Het opstellen van een initiële versie met een correcte opmaak van hoofdstukken.
- Er volgt een conceptversie ter evaluatie.
- Na verbetering volgt het definitieve eindverslag.

De afstudeerstage is opgedeeld in een aantal fases. Aan het eind van de eerste realisatie fase wordt er een evaluatie gehouden. Als dit goed is gegaan en er verder geen correcties doorgevoerd hoeven te worden kan gestart worden aan de volgende fase.

Opdrachtgever

De opdrachtgever is Sense Interactive in Uden.

Binnen het bedrijf is een contactpersoon aangewezen die het contact tussen de school en het bedrijf regelt.

De gegevens van het bedrijf en de contactpersoon staan hieronder.

Bedrijfsnaam	Sense Interactive
Adres	President Kennedylaan 159,5402KC Uden
Telefoon	0413-270101
E-mail	info@sense-i.nl

Contactpersoon	Henrik van Duijnhoven
E-mail	duijnhoven@sense-i.nl

Opdrachtnemer

De opdrachtnemer is de afstudeerder van de Fontys Hogescholen te Eindhoven.

Zijn gegevens zijn hieronder te vinden.

Naam	Brett Heesakkers
Adres	Venkellaar 26, 5467BS Veghel
Telefoon	06 23168632
E-mail	brett.heesakkers@gmail.com
Bedrijf / Organisatie	Fontys Hogescholen te Eindhoven

Informatiematrix

Documenten / afspraken met een exacte datum

Datum	Documenten/afspraken	Begunstigde	Manier	Feedback
27-08-2009	Uitzwaai dag	Brett Heesakkers	-	-
31-08-2009	Aanvang afstudeerstage	Brett Heesakkers	-	-
16-09-2009	Bedrijfsbezoek Luciënne Wijgergangs	Brett Heesakkers	Mondeling	Ja
23-10-2009	Tussenrapport	Brett Heesakkers / Docentbegeleider	Digitaal	-
03-11-2009	Terugkomdag	Brett Heesakkers	Mondeling	Ja
18-12-2009	Opleveren producten	Sense Interactive	-	Ja
14-01-2010	Inleveren scriptie	Fontys Hogescholen	Hardcopy	Nee

Documenten / afspraken zonder een exacte datum

Datum	Documenten/afspraken	Begunstigde	Manier	Feedback
Elke dag 8:30 – 17:00	Start werkdag	Brett Heesakkers	-	-
Ochtend	Vergadering	Brett Heesakkers / bedrijfsbegeleider		
Elke vrijdag voor 15:00	Blog werkzaamheden	Bedrijfsbegeleider / docentbegeleider	Digitaal	-
Na goedkeuring docent-begeleider	PID document / PvA	Docentbegeleider	Digitaal	Ja, e-mail

Archivering

Documenten staan online op www.getdropbox.com. Met behulp van deze website worden alle documenten online opgeslagen en worden ze automatisch opgeslagen op computers die Dropbox hebben geïnstalleerd. De gebruiker moet dan wel uitgenodigd zijn om de documenten te delen.

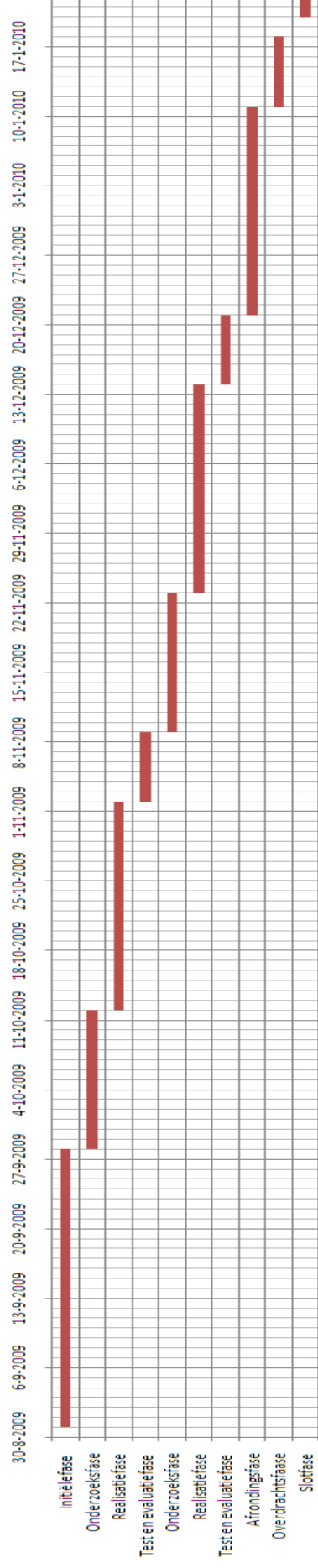
Hierdoor worden de documenten online opgeslagen en beschikken de computers die deelnemen aan de folder over de laatste versies van de documenten.

Hoofdstuk 8 :: Planning

De planning bestaat uit het onderzoek naar [Playground](#), [Twitter](#), [Netvibes](#) en [OAuth](#) en de integratie van Twitter en Netvibes in Playground. Ook is hier de tijd te zien die besteed wordt aan de scriptie.

In de planning is voldoende tijd voor het integratie proces ingedeeld om eventuele uitloop op te vangen.

Fase planning¹



¹ Zie bijlagen: Planning_fase.xlsx

Mijlpalen

Week	Mijlpaal
Week 1	
Week 2	PID document
Week 3	
Week 4	
Week 5	Onderzoek Playground + Twitter + Oauth
Week 6	
Week 7	
Week 8	
Week 9	
Week 10	Twitter
Week 11	Evaluatie Twitter
Week 12	Onderzoek Netvibes
Week 13	
Week 14	
Week 15	
Week 16	
Week 17	Netvibes
Week 18	
Week 19	
Week 20	Scriptie

Hoofdstuk 9 :: Kosten en baten

Het project is voor Sense Interactive intern. Hierdoor heeft het geen andere kosten buiten de afstudeerder om.

Kosten

De kosten voor dit project is voor Sense Interactive de stage vergoeding van € 180,- netto per maand met een totaal van € 900,- voor vijf maanden.

Het gebruik van een laptop.

Baten

Aan het einde van het project beschikt Sense Interactive over Twitter en Netvibes binnen hun eigen Playground. Hiermee heeft de klant de optie om ook deze diensten in hun website te gebruiken.

De afstudeerder behaalt zijn diploma, hij doet veel werkervaring op en het project helpt in de persoonlijke ontwikkeling.

Hoofdstuk 10 :: Risico's

Intern en extern

Intern

- Benodigde kennis ontbreekt.
- Begeleider afwezig voor langere tijd.
- Benodigde software ontbreekt.
- Playground 2.0 heeft een grote verandering ondergaan.

Extern

- Privé omstandigheden.
- Ziekte .

Risicoanalyse

Intern

Benodigde kennis ontbreekt.

Dit zal geen probleem vormen aangezien er voldoende kennis binnen het bedrijf is.

Begeleider afwezig voor langere tijd.

Omdat er meerdere personeelsleden als begeleider kunnen functioneren zal dit niet zo'n groot probleem vormen. Een ander personeelslid neemt dan zijn taak over.

Benodigde software ontbreekt.

Dit kan een probleem vormen. Het kost immers tijd om nieuwe software aan te schaffen of gratis software dat geschikt is te vinden.

Playground 2.0 heeft een grote verandering ondergaan.

Grote wijzigingen in de structuur van Playground 2.0 kan invloed hebben op het verloop van dit project.

Extern

Privé omstandigheden

Dit hoeft geen negatief effect te hebben op het werk. Het ligt geheel aan wat de omstandigheden zijn. Mocht het ernstig zijn dan zal dit wel een probleem veroorzaken met de te behalen afstudeerstage dagen. De afstudeerstage zal dan wat langer door moeten gaan.

Ziekte

Dit kan altijd voorkomen. De gemiste dagen zullen dan ingehaald worden.

II. Opdracht Sense Interactive

1. Beschrijf de probleemanalyse.

Sense Interactive heeft een eigen product ontwikkeld (Playground) voor het beheer van online diensten, voornamelijk websites. Playground maakt veel gebruik van koppelingen met bestaande functionaliteiten van andere partijen zoals Google (route beschrijving, statistieken), YouTube (film), Picnick (fotobewerking), enz. Deze partijen bieden hun diensten middels API's aan. Omdat deze diensten een steeds grotere rol spelen in de online wereld dient Playground hier ook actief op in te springen, dus het steeds up-to-date blijven en nieuwe diensten voor klanten beschikbaar maken via Playground. Hiervoor dienen nieuwe diensten via Playground beschikbaar gemaakt te worden.

2. Beschrijf de afstudeerstageopdracht.

Het koppelen van een aangeboden API aan Playground van een nader te bepalen dienst. Hierbij valt te denken aan Twitter, Flickr (foto database), Photoshop Express(online foto bewerking) of Linked In (social networking).

3. Wat is de onderzoeksvraag?

Wat is Playground en hoe wordt een API aan Playground gekoppeld?

4. Beschrijf de deelprojecten?

- Onderzoeken wat Playground is en hoe Playground werkt.
- Bekijken welke API aan Playground gekoppeld moet worden en hoe dit in zijn werk gaat.
- API aan Playground koppelen.

5. Wat is de doelstelling van de afstudeerstageopdracht?

Via Playground een nieuwe dienst voor de klanten beschikbaar maken door binnen vijf maanden een API aan Playground te koppelen.

6. Vakgebieden:

- ICT wordt vereist om de API aan Playground te koppelen.
- Bij Design wordt gekeken naar het gebruikersgemak van de API en hoe het wordt afgebeeld.

7. Faciliteiten op afstudeerstagebedrijf:

Werkplek, computer met de benodigde software, ondersteuning door huidige personeel, literatuur en uiteraard uitleg over Playground en de technische opbouw hiervan.

8. Persoonlijke ontwikkeling

Ik ga tijdens deze stage dieper in op het vakgebied waar ik later mijn beroep van wil maken en krijg met deze stage dan ook een goed beeld hoe mijn dagelijkse werkweek eruit ziet. Met de stageopdracht krijg ik meer inzicht in het gebruik en de werking van API's.

III. De koppeling met Playground 2.0

In dit document wordt gekeken hoe de twee API's, Twitter en Netvibes, aan Playground gekoppeld worden. Omdat dit document veel technische aspecten op het gebied van ICT bevat, wordt met behulp van voorbeelden uitgelegd wat er nodig is om deze twee API's te koppelen.

Twitter

Twitter wordt onderdeel van Playground 2.0. Hierdoor is Twitter beschikbaar in de Playground omgeving en hebben de klanten de mogelijkheid om hun website van Twitter te voorzien. Twitter is enorm populair waar veel mensen gebruik van maken. Om met de tijd mee te gaan moet Playground ook hiervan gebruik gaan maken.

Twitter kan op twee manier aan Playground gekoppeld worden. Er kan een eigen aanvraag geschreven worden wat de gegevens van Twitter ophaalt en verwerkt of er kan gebruik gemaakt worden van het Zend Framework. Hiermee kan dan via Zend gebruik gemaakt worden van de Twitter API. Zend beschikt over methodes die de Twitter aanvraag, zoals het ophalen van de Tweets van de gebruiker, afhandelt. Wat hierna gedaan moet worden is het verwerken van de gegevens zodat Playground ze kan uitlezen.

Voor dat er gebruik gemaakt kan worden van de Twitter API moet er eerst een Twitter account aangemaakt worden. Hiermee kan vervolgens via de API ingelogd worden bij Twitter om zo de Tweets op te halen.

Om gebruik te maken van OAuth moet er een consumerkey en consumersecret voor de applicatie aangevraagd worden. Hiermee kan Playground toegang krijgen tot de Twitter gegevens zonder dat de gebruiker van de API zijn gegevens bij Sense Interactive hoeft op te slaan. Deze twee keys worden aan het aangemaakt account gekoppeld.

De koppeling van Twitter zal met behulp van Zend worden gemaakt. Hiervoor is gekozen omdat het tijd bespaart en de aanvraag naar de Twitter REST API zeker goed zal verlopen.

Om de werking te testen wordt het inloggen direct in het Zend object meegegeven. In een later stadium moet het inloggen verlopen via OAuth. Met OAuth wordt de gebruiker naar een inlog pagina van de service provider gestuurd, in dit geval Twitter.com. Hierop moet de gebruiker dan inloggen en de applicatie, die gebruik wil maken van Twitter, toestemming geven.

Voordat Twitter in Playground gebruikt gaat worden zal er eerst in een test omgeving aangemaakt worden. Deze testomgeving staat in de folder Twitterpgx. De Twitter applicatie staat vervolgens in '_pgxframework/local/twitter/2.0/'

De testomgeving wordt aangemaakt zodat er ongestoord gewerkt kan worden aan de Twitter integratie in Playground. Wijzigingen in het Playground 2.0 project hebben hierdoor geen invloed op het Twitter project.

Omdat Playground en ook het Zend Framework object georiënteerd te werk gaat wordt er gekeken wat dit inhoud. Dit is benodigde kennis voor het onderzoek wat ook uitgezocht moest worden om verder te komen.

De PHP class

Om de Twitter functionaliteit in Playground te integreren moet er een class aangemaakt worden genaamd 'pgx_local_twitter' in het bestand 'twitter.php'.

```
class pgx_local_twitter implements pgx_interface {  
  
}
```

Er kan een class aangemaakt worden in PHP om op de OOP (Object Oriented Programming) manier te programmeren. Een class is een verzameling methodes bij elkaar gebonden. Er kan een object aangemaakt worden van een dergelijke class waarmee vervolgens de methodes in die class kunnen worden aangeroepen. OOP gaat dus uit van een object met gedefinieerde methodes. Een klein voorbeeldje om een uitleg van OOP te geven.

Er is een object **fiets**. Deze **fiets** heeft een aantal eigenschappen die aangeroepen kunnen worden. De fiets kan bijvoorbeeld zijn snelheid verhogen door de methode aan te roepen **verhoogSnelheid**. Nu wordt deze functie uitgevoerd en wordt de snelheid van **fiets** verhoogt. Omdat **fiets** een object is, is de methode **verhoogSnelheid** aan dat object gebonden. Als er nu een tweede **fiets** wordt gemaakt heeft deze zijn eigen **verhoogSnelheid** en daarmee dan ook een eigen snelheid.

Een object aanmaken betekent dat er een nieuwe instantie aangemaakt wordt van de class. In het voorbeeld wordt er dan een nieuwe **fiets** aangemaakt. Er kan bijvoorbeeld gezegd worden:

```
$mijnFiets = new fiets();
```

```
$janFiets = new fiets();
```

\$mijnFiets betekent in PHP dat er een variabele mijnFiets aangemaakt wordt. Een variabele is een benaming voor een gegeven. \$hallo = "Hallo ik ben Brett"; betekent dat de waarde "Hallo ik ben Brett" in de variabele \$hallo wordt gezet. Alles wat in dubbel of enkele quotes staat wordt een String genoemd. Dit representeert een tekst. Ook is er bijvoorbeeld een Int wat een Integer is. Dit is een numerieke waarde. De ; sluit de regel af. Dit is nodig voor PHP zodat hij weet dat na de ; er niks meer in de variabelen moet.

Nu kan er in het PHP script gebruik gemaakt worden van de variabele \$hallo. Dit is natuurlijk veel handiger dan constant de tekst "Hallo ik ben Brett" te moeten typen. Bovendien kunnen er eenvoudig een aantal bewerkingen met deze variabelen plaatsvinden, zoals het vergelijken van deze variabelen met tekst wat een gebruiker heeft ingevoerd.

Het '\$' teken zegt dus tegen PHP dat het een variabele is. In het geval van \$mijnFiets en \$janFiets wordt tegen PHP gezegd dat deze variabelen een object is van de class **fiets**. Deze variabelen worden dan een instantie van **fiets** genoemd.

Nu kan de methode verhoogSnelheid aangeroepen worden voor \$mijnFiets en voor \$janFiets. In PHP wordt dat gedaan met \$mijnFiets->verhoogSnelheid();

In \$mijnFiets is nu de snelheid verhoogt terwijl in \$janFiets deze nog hetzelfde is. Er zijn nu twee objecten aangemaakt met elk hun eigen waarde.

De PHP interface

Ook is er in de `pgx_local_twitter` class een implements aan toegevoegd. Een implements betekent dat deze class de structuur of te wel de methodes die in de interface zijn gedefinieerd moet overnemen. In dit geval moet de `pgx_local_twitter` class de interface `pgx_interface` gebruiken. Deze interface ziet er als volgt uit.

```
interface pgx_interface {  
  
    public function __execute();  
  
}
```

Hieruit wordt opgemaakt dat `pgx_local_twitter` een methode moet hebben genaamd `public function __execute()`.

Wordt deze methode niet overgenomen dan kan er geen instantie van de `pgx_local_twitter` class aangemaakt worden.

Twitter met PHP, Zend en HTML

Playground kan met de HTML tag ``<pgx:local.twitter />`` de class `pgx_local_twitter` aanroepen. Hierdoor hoeft de programmeur geen PHP kennis te hebben om toch gebruik te kunnen maken van de Twitter class.

Het is ook mogelijk om in JavaScript de class aan te roepen. Met behulp van `pgx('local.twitter')` wordt een AJAX aanvraag uitgevoerd waardoor er ook dynamisch van deze class gebruik gemaakt kan worden. Deze aanvraag zal nodig zijn om van Twitter gebruik te kunnen maken in de Playground 2.0 omgeving. Voor websites van klanten zal dit niet altijd nodig zijn. Hier zal meestal een aanvraag in de HTML pagina voldoende zijn. Om de werking van de Twitter Playground API te demonstreren in het dashboard zal gebruik gemaakt worden van de HTML aanvraag.

Om de gegevens van Twitter op te halen wordt er gebruik gemaakt van het Zend Framework. Hiermee kan een instantie aangemaakt worden van `Zend_Service_Twitter`. Dit object heeft dan de beschikking tot een aantal methodes die de gegevens van Twitter ophalen.

Zo kan bijvoorbeeld de methode `status->friendsTimeline` aangeroepen worden. Dit geeft vervolgens een resultaat terug in de vorm van een `Zend_Rest_Client_Result` object. Dit object bevat de Twitter gegevens van de gebruiker. Deze methode geeft alle berichten van de gebruiker terug die op zijn Twitter pagina te vinden zijn.

Nu de `pgx_local_twitter` class is aangemaakt worden hierin methodes geschreven die de functionaliteit van Zend met Playground combineren. Als eerst wordt de `__execute()` methode uitgewerkt, aangezien dit volgens de interface moet.

Als Playground dan de tag ``<pgx:local.twitter />`` leest wordt als eerst de `__execute()` methode uitgevoerd.

Deze methode kijkt of de variabele `action` is gezet. Deze variabele kan gezet worden door een `action` mee te geven in de Playground tag.

Er ontstaat dan bijvoorbeeld '`<pgx:local.twitter action="friendsTimeline" />`'. Als deze variabele `action` is gezet zal de `__execute` methode een gelijknamige methode, in dit geval `friendsTimeline`, uitvoeren.

De methode `friendsTimeline` zal op zijn beurt weer een Zend methode aanroepen en zo de Tweets van de gebruiker ophalen. De gegevens worden als een instantie van `Zend_Rest_Client_Result` terug gestuurd. Dit object bevat een aantal SimpleXML objecten. SimpleXML is een XML omgezet in een object. Op deze manier kan PHP makkelijk XML lezen en bewerken.

Playground en core.repeat

Omdat Playground gebruikt maakt van XML om gegevens uit te lezen moet het Zend object omgezet worden naar een XML. PHP heeft hier een ingebouwde methode voor genaamd '`asXML()`'. Het object is nu een XML string die aan Playground gegeven kan worden om uit te lezen. Door aan de variabele '`$this->xml`' de XML string waarde mee te geven kan Playground verder gaan om deze variabele uit te lezen.

Om dit voor elkaar te krijgen heeft Playground een class genaamd '`pgx_core_repeat`'. Met de tag '`<pgx:core.repeat`' kan deze class worden aangeroepen. Door een id mee te geven aan deze tag kan de XML die gelezen moet worden mee worden gegeven. Bijvoorbeeld '`<pgx:core.repeat id="{{twitter.xml}}"></pgx:core.repeat>`'. In de Playground tag '`<pgx:local.twitter`' moet in dit geval een id krijgen met de waarde `twitter`.

De tag wordt dan: '`<pgx:local.twitter id="twitter" action="friendsTimeline" />`'. Nu wordt de variabele '`$this->xml`' uitgelezen in de '`pgx_core_repeat`' class.

Als vervolgens tussen de openings tag '`<pgx:core.repeat id="{{twitter.xml}}">`' en de sluit tag '`</pgx:core.repeat>`' de naam van een XML node wordt geplaatst tussen `{ }` tekens, zoals `{status.text}`, wordt de tekst van de Tweet in de XML uitgelezen. Deze tekst bevindt zich dan, in de XML, in de tag '`<status><text></text></status>`'.

Met `core.repeat` worden de XML gegevens van Twitter uitgelezen en wordt het resultaat op het scherm getoond.

OAuth

Om in te loggen bij het Twitter account van de gebruiker is gekozen om gebruik te maken van OAuth. OAuth houdt in dat de gebruiker naar Twitter.com wordt gestuurd om vervolgens in te loggen, als hij / zij nog niet is ingelogd, en toestemming te geven aan de API die gebruik wil maken van zijn of haar Twitter gegevens. Op deze manier hoeft de gebruiker geen inlog gegevens aan een derde partij vrij te geven.

Om van Twitter gebruik te kunnen maken in Playground 2.0 mag de gebruiker echter niet naar een andere pagina worden gestuurd. Dit geeft anders problemen met de inlog sessies van Playground en is ook niet gewenst bij Sense Interactive. Om dit probleem op te lossen wordt er gebruik gemaakt van cURL. Dit is een manier van PHP om gegevens van een pagina op te halen en op te slaan in een variabele zonder naar die pagina te gaan.

De gebruiker geeft zijn inlognaam en wachtwoord waarna deze naar Twitter.com worden gestuurd om in te loggen en meteen toestemming te geven aan Playground 2.0 om gebruik te mogen maken van zijn of haar Twitter.

De OAuth access_token wordt opgeslagen in een sessie waarna de gebruiker van Twitter gebruik kan maken. In theorie kan deze token ook opgeslagen worden in een database. Hiervoor moeten nog test worden gedaan in een later stadium.

Foutafhandeling

Om fouten af te handelen wordt gebruik gemaakt van een methode genaamd 'twitter_switch'. Deze zorgt ervoor dat, als er iets fout gaat bij het ophalen van de Twitter gegevens, er een foutpagina wordt getoond. Doordat hiervan gebruik gemaakt wordt zullen de methodes om de Twitter gegevens op te halen via deze methode aangeroepen worden.

De Playground tag ziet er dan als volgt uit: '`<pgx:core.html source="/_templates/twitter/twitter_switch.html" action="twitter_switch" getPage="/_templates/twitter/accountLimit.html" />`'.

Playground laad dan de HTML pagina 'twitter_switch' in en stuurt daarbij de variabelen action met de waarde 'twitter_switch' en getPage met de waarde '/_templates/twitter/accountLimit.html' mee. In deze pagina is het volgende te vinden:

```
<pgx:local.twitter
  id="twitter"
  action="{action}"
  getPage="{getPage}"
  errorPage = "/_templates/twitter/error.html"
  loginPage = "/_templates/twitter/login_twitter_form.html"
  signatureMethod="HMAC-SHA1"
  callbackUrl="http://projects.mrxt.nl/Sense Interactive/Brett/PG2/pgi/2.0/twitter.html"
  consumerKey="zrsZaf4ul5AvnspvjJ6qng"
  consumerSecret="R8iTVx00FcD9MQ9gWxU2tBLyKsfOqmOqihmtOBB3l"
/>
<pgx:core.html source="{twitter.path}" />
```

De id waarde krijgt de naam twitter en de action wordt twitter_switch.

De '{getPage}' wordt door Playground omgezet naar accountLimit omdat deze is meegegeven in de '`<pgx:core.html`' tag. Verder zijn de gegevens die meegegeven worden, signatureMethod, callbackUrl, consumerKey en consumerSecret, nodig om gebruik te kunnen maken van de Twitter API. De errorPage is de locatie naar de error pagina en de loginPage is de locatie naar de login pagina.

Deze HTML pagina zorgt ervoor dat er een error pagina wordt getoond als de aanvraag niet succesvol is.

De methode 'twitter_switch' geeft een 'path' waarde terug die uitgelezen wordt in de 'core.html' tag van Playground. Deze laad in dit geval 'accountLimit.html' in.

In 'accountLimit.html' staat het volgende:

```
'<div>
    <pgx:core.repeat xml="{[twitter.xml]}">
        {remaining-hits}/{hourly-limit}
    </pgx:core.repeat>
</div>'
```

Dit wil zeggen dat Playground de 'core.repeat' class aanroept en de XML variabele gebruikt die door de 'twitter_switch' methode in de 'local.twitter' class is gezet.

De {remaining-hits} en {hourly-limit} worden door de 'core.repeat' vervangen door de waarde uit de XML.

Netvibes

De koppeling van Netvibes gebeurt door gebruik te maken van de Exposition Library van Netvibes. Dit zijn PHP scripts die gebruikt worden om widget gegevens op te halen en om een widget zijn HTML op te bouwen. Het gebruik van de widget zelf wordt afgehandeld doormiddel van JavaScript. Deze scripts worden door Netvibes en de ontwikkelaar van de widget verzorgt.

Een widget moet in een iframe geladen worden. Dit is niet alleen veilig, ook zorgt dit ervoor dat de JavaScripts niet met elkaar in conflict komen. Omdat de widgets in een iframe geladen worden moet er een eigen JavaScript code geschreven worden zodat de wijziging van een widget zijn gegevens kan worden afgevangen. Een template van dit script is in de Netvibes documentatie te vinden.

Deze invoer kan dan worden opgeslagen in een database zodat de gebruiker niet steeds opnieuw de gegevens hoeft in te voeren wanneer hij of zij Playground opnieuw bezoekt.

Als eerst is er een script van Netvibes nodig.

```
<script type="text/javascript"
src="http://www.netvibes.com/js/UWA/Utils/IFrameMessaging.js"></script>
```

Vervolgens moet er een eigen script gemaakt worden die de gegevens kan afhandelen. Dit script is in de bijlage terug te vinden. Als de gebruiker bijvoorbeeld de titel van een widget verandert, dan wordt een functie uit dit script aangeroepen en kan hierop gereageerd worden. Hier kan dan bijvoorbeeld met Playground een PHP script worden aangeroepen wat de gewijzigde gegevens opslaat.

Netvibes in Playground

Om Netvibes in Playground te gebruiken moet een eigen PHP script gemaakt worden die methodes van de Netvibes Exposition Library gebruikt om een widget te maken.

Dit PHP script wordt dan aangeroepen in een iframe en bouwt vervolgens de HTML pagina op met de widget. Om dit voor elkaar te krijgen is geen 'core.repeat' van Playground nodig. De widget heeft immers zijn eigen omgeving in het iframe. Wat wel gebruikt wordt is de '<pgx:core.html source="" />' tag of met JavaScript de code 'pgx('core.html',argumenten)'. In plaats van argumenten wordt dan een object meegegeven met onder andere de variabele source. De source wordt gevuld met de locatie naar een HTML pagina.

Ook worden Netvibes Playground tags gebruikt. Er is bijvoorbeeld de tag:

```
<pgx:local.netvibes.widget
  action="render_widget_container"
  widgetId="0001"
  uwa="4NY4n5gbqbyvfg5gbqbyvfg4x"
  container="/_templates/netvibes/widget_container.html"
  frame="/_templates/netvibes/widget_frame.html"
  proxy="http://projects.mrxt.nl/Sense
Interactive/Brett/Afstudeerstage/Dashboard/pgi/proxy.html"
  width="500"
  height="400"
  class="netvibes"
  title="My Title"
/>
```

Deze Playground tag zorgt ervoor dat er een widget geladen wordt. In dit geval is het een To do lijst widget.

De action “render_widget_container” laat de widget op het scherm zien.

De widgetId moet een unieke waarde hebben voor elke widget.

UWA is de unieke naam voor de widget. Hiermee wordt de widget opgebouwd.

De container is de HTML pagina waar de widget container wordt gemaakt. Door op deze manier de pagina aan te roepen kan deze ook op een andere locatie staan. Hetzelfde geldt voor frame. Met deze HTML pagina wordt de iframe opgebouwd.

De proxy wordt gebruikt om de proxy.html pagina te linken aan een widget. Dit is nodig om gebruik te kunnen maken van de afhandeling van de gegevens wijzigingen van een widget.

De width en height zijn de afmetingen van de iframe. De class is de naam van de stijl die aan de container van de widget mee gegeven wordt. Hiermee kan vervolgens in een stylesheet de container van de widgets worden opgemaakt.

De title is de titel die bij de container komt.

Deze Playground tag wordt vervangen door de HTML in de container pagina.

Om een Netvibes widget te maken moet de UWA opgehaald worden. Dit wordt gedaan met behulp van de Netvibes REST API. Deze API haalt een gebruiker zijn publieke Netvibes pagina op. Uit deze gegevens kunnen vervolgens de widgets opgehaald worden.

Playground 2.0 dashboard

De Playground 2.0 dashboard beschikt over een Widget Inspector. Hiermee kan de gebruiker kiezen uit een aantal widgets en deze op zijn of haar dashboard slepen. De Widget Inspector bestaat uit Google Maps, Google Analytics, Twitter, de Netvibes widgets etc.

Nadat de widget op de dashboard is geslepen, kan de gebruiker de plaats bepalen waar de Twitter widget zich moet tonen. Doormiddel van te slepen kan de widget verplaatst worden. Ook is er de mogelijkheid om de widget in te klappen zodat er alleen nog maar een titel balk overblijft met de widget naam.

De widgets worden ingeladen met JavaScript. Met AJAX wordt een widget zijn pagina op het dashboard geladen.

Als het dashboard is geladen kan de gebruiker hier alles vinden wat hij of zij nodig heeft.

IV. Zend_Service_Twitter methodes

- `account->verifyCredentials`
Controleert of de gegevens van de gebruiker kloppen.
- `account->rateLimitStatus`
Geeft het aantal aanvragen weer wat de gebruiker over heeft. Maximaal 150 aanvragen per uur.
- `status->friendsTimeline`
Laat de laatste twintig tweets van de gebruiker en zijn vrienden zien.
- `status->userTimeline`
Laat de laatste twintig tweets van de gebruiker zien.
- `status->publicTimeline`
Laat de laatste twintig publieke tweets die niet beschermd zijn zien. Deze word elke 60 seconden door Twitter vernieuwd.
- `status->replies`
Laat de laatste twintig tweets zien met een @gebruikersnaam erin voor de ingelogde gebruiker.
- `status->show`
Laat een enkele tweet zien.
- `status->update`
Maakt een nieuwe tweet.
- `status->destroy`
Verwijdert een tweet.
- `user->friends`
Laat de vrienden van de gebruiker zien.
- `user->followers`
Laat de volgers van de gebruiker zien.
- `user->show`
Laat een enkele gebruiker zien.
- `directMessage->messages`
Laat alle berichten van de gebruiker zien.
- `directMessage->sent`
Laat alle verzonden berichten zien.
- `directMessage->create`
Maakt een nieuw bericht.
- `directMessage->destroy`
Verwijdert een bericht.
- `friendship->create`
Maakt een nieuwe vriend aan.
- `friendship->destroy`
Verwijdert een vriend.
- `favorite->favorites`
Laat de laatste twintig favoriete tweets zien.
- `favorite->create`
Voegt een nieuwe favoriete tweet aan de favoriete lijst toe.
- `favorite->destroy`
Verwijdert een favoriete tweet uit de favoriete lijst.

V. cURL

Wat is CURL?

cURL is een pakket functies waarmee je makkelijk kan communiceren met allerlei servers. cURL ondersteunt o.a. http, https, ftp, telnet, HTTP POST, proxies, cookies en user+password authenticatie. Je kan het gebruiken om webpagina's in te lezen maar je kan er ook formulieren mee submitten.

cURL en Playground

cURL kan dus worden gebruikt om in te loggen bij een dienst die gebruik maakt van OAuth. Door POST waarde mee te geven wordt direct ingelogd bij de dienst en wordt er een terugkeer link gegenereerd met een geautoriseerde OAuth Request Token. Deze gegevens worden dan opgeslagen in een String variabelen die uitgelezen kan worden.

Met behulp van regular expressions wordt er gekeken of er geen error in de variabelen zit. Is dit niet het geval dan worden de oauth_token en oauth_verifier waarde er uit gehaald. Met deze waarde wordt vervolgens de Request Token omgezet in een Access Token.

Op deze manier kan de gebruiker met een formulier zijn gebruikersnaam en wachtwoord op Playground ingeven en worden deze waardes naar de OAuth pagina van de dienst gestuurd. Hierdoor hoeven er geen gegevens van de gebruiker bij Playground bekend te zijn om gebruik te kunnen maken van de dienst.

VI. OAuth

Wat is OAuth?

Kort gezegd is OAuth een manier om in te loggen bij een online dienst zoals Flickr terwijl je gebruik maakt van een andere website. Om OAuth uit te leggen is het schetsen van een situatie de beste manier.

Situatieschets

Als voorbeeld nemen we Jan die foto's op Flickr heeft staan en hij wil deze laten uitprinten als een fotoboek. De website maakuwfotoboek.nl heeft de mogelijkheid om een fotoboek te maken met foto's die op onder andere Flickr staan.

Jan gaat naar maakuwfotoboek.nl en geeft in de website aan dat hij een fotoboek wil maken met foto's die op zijn Flickr account staan. maakuwfotoboek.nl heeft een Flickr account aangemaakt en een OAuth Consumer Key en Consumer Secret aangevraagd. Met behulp van deze twee gegevens kan maakuwfotoboek.nl een Request Token aanvragen.

Zodra de Request Token opgehaald is wordt Jan naar Flickr gestuurd samen met de Request Token. OAuth vraagt om een geldige gebruiker en vraagt daarom om Jan in te laten loggen bij Flickr. Hij hoeft dus geen inloggegevens aan maakuwfotoboek.nl door te geven.

Vervolgens vraagt OAuth om maakuwfotoboek.nl toestemming te geven om Jan zijn foto's op te halen. Nadat Jan is ingelogd en toestemming heeft gegeven wordt hij terug gestuurd naar maakuwfotoboek.nl met de geautoriseerde Request Token.

Nu zal maakuwfotoboek.nl de Request Token omzetten in een Access Token. De Access Token wordt gebruikt om de foto's van Jan op te halen terwijl de Request token alleen wordt gebruikt om gebruikers toestemming op te halen. In de eerste aanvraag wordt de Request Token omgezet in een Access Token. In de tweede aanvraag worden de foto's opgehaald.

Nu krijgt Jan de foto's te zien op maakuwfotoboek.nl en kan hij selecteren welke hij graag uitgeprint wil hebben.

OAuth in Playground

Omdat OAuth werkt met het versturen van de gebruiker naar de website die een dienst ter beschikking stelt, moet er een oplossing gevonden worden om dit te omzeilen. Dit moet gedaan worden omdat de diensten die aan Playground zijn gekoppeld dynamisch in een venster gegenereerd worden. Bovendien is het onhandig voor een klant om steeds uit Playground te gaan om bij een van deze diensten in te loggen.

Conclusie

OAuth is een veilige manier van inloggen wat de gebruiker de zekerheid geeft dat zijn gegevens beschermd blijven en dat zijn gegevens maar op een plaats geregistreerd zijn. Er moet echter wel een oplossing gezocht worden om OAuth uit te voeren zonder dat de gebruiker uit Playground hoeft te gaan.

VII. PrototypeJS

Een belangrijk onderdeel van Playground is het Prototype framework. Dit framework is ontwikkeld als fundering voor AJAX ondersteuning. Het is opgebouwd als een class object georiënteerd framework met een uitgebreide AJAX ondersteuning en eenvoudige DOM manipulatie. Door gebruik te maken van PrototypeJS kan Sense Interactive gebruik maken van vele methodes wat veel tijd en geld bespaard.

PrototypeJS maakt ook gebruik van script.aculo.us. Dit is een verzameling van JavaScript bibliotheken om de gebruikersinterface te verrijken. Hiermee kan bijvoorbeeld een eenvoudig versleepbaar venster gemaakt worden.

AJAX

Waar Playground ook veel gebruik van maakt is AJAX. Met behulp van deze techniek kan er met dynamische webpagina's gewerkt worden. Dit wil zeggen dat de webpagina gegevens uit bijvoorbeeld een database ophaalt zonder dat de pagina opnieuw geladen hoeft te worden. De browser begint op deze manier veel te lijken op een desktop applicatie wat voor de gebruiker een grote verbetering is.

XML

Playground werkt met XML(Extensible Markup Language).

JSON

Nog een techniek waar Playground gebruik van maakt is JSON.

JSON staat voor JavaScript Object Notation. Dit is een gegevens uitwissel formaat wat weinig resources vraagt. Het is makkelijk voor mensen om te lezen en schrijven en makkelijk voor machines om te parsen en genereren.

JSON wordt in de vorm van een string, met naam, waarde paren, gebruikt. Op deze manier kan via JSON een object en/of een array verstuurd worden. Hierdoor kan JavaScript en PHP eenvoudig de data in de string omzetten naar een bruikbaar object of array.

Met behulp van AJAX kan je bijvoorbeeld een JSON string naar PHP sturen. Deze kan de string omzetten in bruikbare gegevens om bijvoorbeeld een database query te versturen. Vervolgens kan PHP deze data weer omzetten naar een JSON string en terugsturen naar de AJAX request. Vervolgens kan JavaScript deze gegevens weer makkelijk omzetten naar bijvoorbeeld een object en kan deze gebruikt worden in het script.