

Windows Live Gadgets

EEN PERSOONLIJK DASHBOARD IN TIJDEN VAN INFORMATIEOVERVLOED

Windows Live heeft als doel informatiebronnen samen te brengen in één enkel overzicht; zowel voor thuis- als zakelijke gebruikers. Voor veel gebruikers betekent dit de mogelijkheid in één oogopslag informatie tot zich te nemen, zoals het laatste nieuws, het weer en tv-programma's. Naast de thuisgebruikers biedt Live.com ook veel voordelen voor bedrijven en hun medewerkers. In dit artikel bouwen we een gadget voor Windows Live die CRM-informatie weergeeft.

In 2005 onthulde Microsoft het nieuwe Windows Live-platform. Het doel van dit platform is om een geheel van persoonlijke internetdiensten en software aan te bieden. Binnen de Windows Live family kunnen onder andere de volgende diensten worden onderscheiden: Live.com, Windows Live Mail, Windows Live Messenger, Windows Live Spaces en Windows Live Safety Center. In totaal zo'n twintig verschillende diensten. De dienst waaraan we in dit artikel aandacht besteden is Live.com en we richten ons op de zakelijke gebruiker die we bedrijfsinformatie willen tonen: een pagina met alle belangrijke informatie van verschillende systemen en waarschuwingen. Deze 'dashboards' kunnen bijvoorbeeld de startpagina vormen wanneer men de werkdag begint. Wanneer Live.com geïntegreerd wordt met applicaties zoals Sharepoint en Microsoft Dynamics CRM krijgen medewerkers een overzicht van openstaande activiteiten, klantgegevens, contactmomenten, nieuwe documenten, enzovoort. In plaats van een grote hoeveelheid pagina's door te moeten nemen, staat alles in één enkel overzicht. Dit artikel laat zien hoe eenvoudig het is om businessapplicaties te integreren in de Live.com-omgeving door middel van gadgets, gebaseerd op het Atlas/AJAX-framework.

Gadgets

Het nieuwe Live.com-platform biedt de mogelijkheid voor het maken van gadgets, vergelijkbaar met de gadgets die je kunt ontwikkelen voor Vista Sidebar en Windows Sideshow. Deze gadgets zijn mini-applicaties die informatie kunnen tonen. De gadgets dienen volgens een bepaalde standaard ontwikkeld te worden, zodat ze toegevoegd kunnen worden aan de Live.com-site. De site zal vervolgens alle gadgets laden die geplaatst zijn op de gepersonaliseerde webpagina. Later in dit artikel wordt uitgelegd waaraan bovengenoemde standaard moet voldoen.

Communicatie

Voor dit artikel wordt gebruikgemaakt van gadgets die actief zijn op <http://localhost>, zodat het eenvoudiger is de gadgets te debuggen. Eenmaal gereed voor productie kunnen gadgets beschikbaar worden gesteld door ze te uploaden naar de repository van Live.com (<http://gallery.live.com>), of de gadgets te plaatsen op een andere internetsite die te benaderen is door de Live.com proxy, bijvoorbeeld <http://www.bedrijfsnaam.nl/gadget.xml>.

Eerste vereisten

Voordat gadgets op de Live.com site gebruikt kunnen worden, moet je de site toevoegen als trusted site. Dit is nodig om ervoor te zorgen dat de gadgets voldoende rechten hebben tot het uitvoeren van de scripts. Het gaat hierbij om de site: <http://live.com>. Naast Live.com heeft Microsoft ook de site Start.com in het leven geroepen. Ook op Start.com kan gebruik worden gemaakt van dezelfde gadgets. Wanneer Start.com wordt gebruikt in plaats van Live.com, dan zal je deze

site moeten toevoegen als trusted site: <http://gadgets.start.com>. Het is alleen noodzakelijk om bovenstaande sites toe te voegen aan de lijst met 'Trusted Sites' wanneer gadgets worden ontwikkeld op localhost. Eindgebruikers van gadgets hoeven dit niet te doen

Implementatie

De vervolgstap is het daadwerkelijk implementeren van de gadget. Hiervoor zijn drie elementen vereist die de basis van de gadget vormen:

- **Manifest** - Een manifest bevat de metadata voor de gadget zoals de titel en de omschrijving, maar bevat tevens verwijzingen naar externe bestanden die nodig zijn voor het correct uitvoeren van de gadget.
- **Javascript** - De javascript-bestanden bevatten de functionaliteit voor de gadget. Het is vereist om de javascript-class(es) te registreren en uit te voeren binnen het ATLAS-framework dat draait op de Live.com-omgeving.
- **Stylesheet** - Om de vormgeving van de gadget te bepalen, kan gebruik worden gemaakt van CSS-stylesheets.

In de volgende paragrafen implementeren we de bovenstaande elementen voor de gadget. De functie van de gadget is het tonen van openstaande activiteiten uit een Microsoft Dynamics CRM-applicatie.

Manifest

Het manifest van een gadget is het startpunt van de ontwikkeling. Het manifest is een XML-bestand dat de gadget definieert. Het bevat een aantal parameters zoals de naam, omschrijving en links naar externe bestanden die toebehoren aan de gadget. De opmaak van het manifest heeft erg veel weg van de definitie van een RSS-feed, met een aantal toevoegingen.

Javascript

Functionaliteit kan aan de gadget toegevoegd worden door gebruik te maken van een javascript-bestand. Dit javascript-bestand dient opgezet te worden volgens de Live.com gadget-standaard. De eerste stap is het registreren van een namespace voor de gadget aan het begin van het javascript-bestand. Deze namespace zal overall binnen de gadget worden gebruikt. Als voorbeeld is gekozen voor de namespace 'Avanade'. Nadat de namespace is bepaald, dient de 'class' van de gadget gedefinieerd te worden, volgens de standaard AJAX-manier: 'Avanade.CRM'. Deze wordt geïnitieerd door middel van de function-methode. Alle methodes van de gadget zullen toegevoegd worden aan deze class. De eerste regel binnen de class is het initialiseren van de basis class, hetgeen te vergelijken is met het overerven van een abstracte class of een interface. Twee andere methoden die geïmplementeerd moeten worden zijn initialize en dispose. Ook deze worden geïnitieerd met de function-call en overriden hun basismethode. De initializemethode is te vergelijken

```
<?xml version="1.0"?>
<rss version="2.0" xmlns:binding="http://www.start.com">
  <channel>
    <title>My CRM activities</title>
    <link></link>
    <description>
      Shows the user's current Microsoft Dynamics CRM 3.0 activities
    </description>
    <language>en-us</language>
    <pubDate>Sat, 25 Mar 2006 01:00:00 GMT</pubDate>
    <binding:type>Avanade.CRM</binding:type>
    <icons>
      <icon height="32" width="32">
        http://localhost:8082/gadgetsamples/avanade/note_view.png
      </icon>
    </icons>
    <item>
      <link>
        http://localhost:8082/gadgetsamples/avanade/crm.js
      </link>
    </item>
    <item>
      <link binding:type="css">
        http://localhost:8082/gadgetsamples/avanade/crm.css
      </link>
    </item>
  </channel>
</rss>
```

Codevoorbeeld 1.

```
registerNamespace("Avanade");

Avanade.CRM = function(p_elSource, p_objArgs, p_namespace)
{
  Avanade.CRM.initializeBase(this, arguments);

  this.initialize = function(p_objScope)
  {
    Avanade.CRM.getBaseMethod(this, "initialize",
      "Web.Bindings.Base").call(this, p_objScope);
  }

  this.dispose = function(p_blnUnload)
  {
    Avanade.CRM.getBaseMethod(this, dispose",
      "Web.Bindings.Base").call(this, p_blnUnload);
  }
}

Avanade.CRM.registerClass("Avanade.CRM", "Web.Bindings.Base");
```

Codevoorbeeld 2.

met een constructor. Hierin worden initialisaties uitgevoerd en kunnen eventuele knoppen met functies toegevoegd worden aan de gadget. De dispose-methode zal gebruikt worden als destructor en zorgt voor het opruimen en vrijgeven van resources. Het is belangrijk hiervan gebruik te maken om memory-leaks te voorkomen. De laatste stap voor het maken van de basisopbouw van de gadget is het registreren van de complete class. Dit wordt gedaan met behulp van de registerclass-functie. De classnaam die hierin wordt gespecificeerd dient ook gebruikt te worden in het manifestbestand. Het resultaat van de bovengenoemde stappen is te zien in codevoorbeeld 2. Nu de basis gelegd is voor het javascript-bestand, kunnen we aan de gadget specifieke functionaliteit toevoegen. Nieuwe methodes kunnen binnen de class gedefinieerd worden, zoals dat met normale javascript-methodes wordt gedaan. Het aanroepen van deze methodes zal gedaan worden vanuit de initialize-methode, of vanuit controls die toegevoegd zijn aan de gadget vanuit de initialize-methode

Opslaan en laden van instellingen

Het gadget-framework op de Live.com-pagina biedt ontwikkelaars de mogelijkheid standaardfunctionaliteit van de site te gebruiken. In codevoorbeeld 3 is te zien dat de Avanade.CRM-class een drietal parameters meekrijgt van de Live.com-proxy, te weten p_elSource, p_objArgs en p_namespace. Voor het laden en opslaan van instellingen is p_objArgs belangrijk.

Voor het laden van de instellingen kan gebruikgemaakt worden van

```
// Create data object and initialize it with our current configuration
var data = new Object();

data.type = parseInt(m_module.GetBindingType(), 10);
data.binding = (data.type == Start.Const.BindingType.Internal) ?
  Start.Util.ModuleToString(p_objArgs.module.GetBinding()) :
  p_objArgs.module.GetBinding();

data.params = p_objArgs.module.GetBindingParams();
data.isExpanded = true;
data.view = m_module.GetView();

// Add any data that we want to store
if (!data.params)
  data.params = new Object();

m_crmServerName = document.all["tbCrmAddress"].value;
data.params.m_crmServerName = m_crmServerName;

// Refresh the data for this gadget
Start.BookmarkService.UpdateBookmark(p_objArgs.module.GetId(),
  p_objArgs.module.GetTitle(), p_objArgs.module.GetLink(),
  Start.Const.MyPagesFolderIdStr, Start.Util.ObjectToString(data));
```

Codevoorbeeld 3.

de variabelen die aangemaakt zijn tijdens het opslaan van de instellingen. In codevoorbeeld 3 is een nieuw 'data'-object aangemaakt, waarbij parameters zijn aangemaakt door het aanroepen van de functie:

```
p_objArgs.module.GetBindingParams();
```

Vervolgens kunnen we aan dit object de parameters toevoegen die opgeslagen dienen te worden:

```
data.params.m_crmServerName = m_crmServerName;
```

Ten slotte worden deze waarden opgeslagen door gebruik te maken van een functie die beschikbaar wordt gesteld door het framework. Het 'data'-object wordt hiervoor geserializeerd naar een string en deze wordt meegegeven aan de functie:

```
Start.BookmarkService.UpdateBookmark(...);
```

Wanneer een volgende keer de gadget aangeroepen wordt vanaf Live.com, kan de variabele m_crmServerName opgevraagd worden uit p_objArgs.m_crmServerName:

```
m_crmServerName = p_objArgs.m_crmServerName;
```

Het is wel zo dat verschillende gadgets op dezelfde pagina in theorie de beschikking kunnen hebben over de instellingen van andere gadgets op de pagina. Het framework biedt weliswaar geen supported mogelijkheden om vanuit één gadget andere gadgets te benaderen, het is altijd belangrijk om gevoelige informatie zoals inloggegevens niet in deze instellingen op te slaan. In de gadget wordt dan ook gebruikgemaakt van Windows Authenticatie om Microsoft CRM te benaderen.

Stylesheets

Standaard hebben alle gadgets op Live.com en Start.com automatisch de stijl van de websites. Het is mogelijk de look-and-feel van zelfontwikkelde gadgets aan te passen door middel van Stylesheets. Deze stylesheets worden geïmplementeerd volgens de CSS-standaard. Iedere entry in de CSS dient te beginnen met de gedefinieerde namespace gevolgd door een underscore en de naam van het gadget-object. In het geval van dit voorbeeld zal een div-definitie er uitzien als in codevoorbeeld 4. Dit is dus iets anders dan div.Avanade_CRM. In dit laatste geval zal de div een stylesheet class-naam moeten hebben, genaamd 'Avanade_CRM'. De gadgets op Live.com krijgen echter allemaal hun eigen namespace waarin de verschillende elementen opereren. In dit geval is die namespace 'Avanade.CRM' en iedere div daarbinnen maakt gebruik van bovenstaande style-definitie. Het gebruik van classnamen zoals hierboven beschreven is uiteraard wel mogelijk, maar dan moet nog steeds de namespace

```
.Avanade_CRM div
{
  margin: 2px;
  position: relative;
}
Codevoorbeeld 4.
```

ervoor worden gezet (.Avanade_CRM div.customname). Dit allemaal met het doel te voorkomen dat verschillende gadgets op één pagina met elkaar interverniëren. Door de namespace toe te voegen wordt dit voorkomen.

Resultaat

Om het resultaat van bovenstaande acties te zien, worden de bestanden naar een webservice gekopieerd die contact kan maken met een Microsoft CRM-applicatie. Beide zijn aanwezig op een intern netwerk. In Internet Explorer wordt de site www.Live.com geladen waar vervolgens wordt ingelogd met een Windows Live-ID (voorheen Passport-ID genaamd). De persoonlijke pagina kan nu aangepast worden. Door te kiezen voor 'add stuff' en vervolgens voor 'Advanced Options' zijn op verschillende manieren gadgets toe te voegen:

1. Via het zoeken naar een feed
2. Via een URL
3. Door een OPML-bestand te importeren.

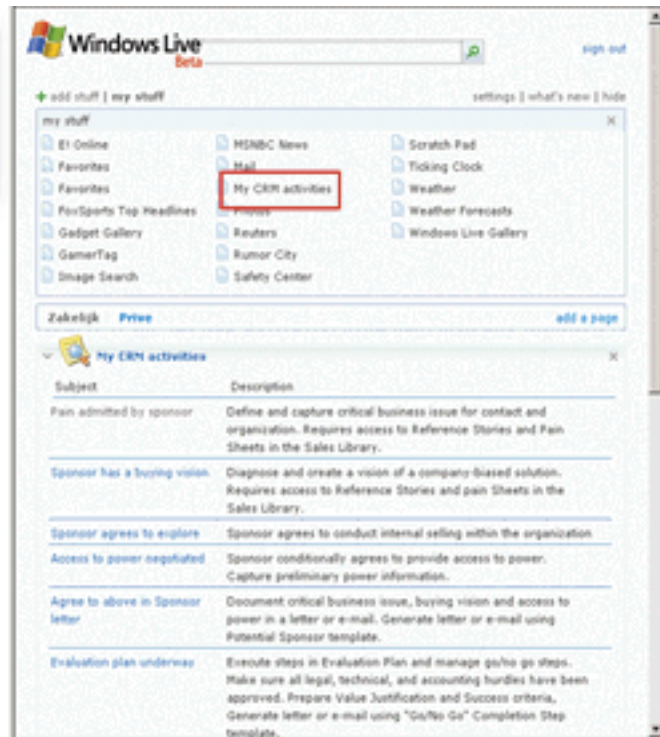
Er wordt een URL opgegeven naar het XML-manifestbestand van onze lokale gadget en vervolgens wordt de gadget toegevoegd aan de persoonlijke Live.com-pagina. Nadat de gadget op de pagina geladen is, wordt deze geconfigureerd, zodat deze kan communiceren met de Microsoft CRM-applicatie. Het resultaat is zichtbaar in afbeelding 1, een overzicht van de CRM-activiteiten op het persoonlijke dashboard. Wanneer een bedrijf voor verschillende interne applicaties gadgets maakt, is het voor de medewerkers mogelijk deze zelf toe te voegen aan hun persoonlijke Live.com-pagina. Het resultaat zal een pagina zijn waarop iedere medewerker de informatie kan vinden die hij of zij het belangrijkste vindt.

Opmerkingen

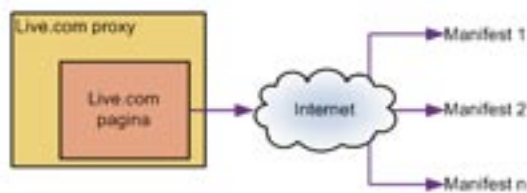
Het feit dat gadgets publiekelijk toegankelijk zijn, lijkt wellicht gevaarlijk voor businessgevoelige informatie, aangezien de businessapplicaties normaal gesproken niet publiekelijk te benaderen zijn. Het is echter niet zo erg als het lijkt; de gadgets gebruiken de proxy van Live.com alleen om instellingen op te slaan of op te halen. De businessinformatie daarentegen kan via javascript door de browser worden opgehaald, wat resulteert in directe communicatie tussen de gadget en de businessapplicatie binnen het intranet. Dit is schematisch weergegeven in afbeelding 2 en 3. Om de communicatie extra te beveiligen kan SSL gebruikt worden tussen de browser en de server. Microsoft heeft aangegeven dat omstreeks september 2006 een aantal wijzigingen wordt doorgevoerd aan het gadget-framework, zodat het ook mogelijk is om Windows Live-gadgets te hosten in Windows Vista's Sidebar.



Afbeelding 3. Uitvoeren van de gadgets



Afbeelding 1. CRM-activiteiten op het persoonlijke dashboard.



Afbeelding 2. Laden van Live.com gadgets

Gepersonaliseerde content op de werkplek

Met gadgets van Live.com is het mogelijk een persoonlijk dashboard te maken dat ook in de behoeften van een businessgebruiker kan voorzien. Je kunt een applicatie, die slechts via het intranet te benaderen is, opnemen op een dashboardpagina als Live.com. De voordelen zijn evident: ten eerste wordt bedrijfsgevoelige informatie niet aan de buitenwereld blootgesteld, doordat het lokale javascript op de client de verbinding legt met deze intranetapplicatie. Ten tweede kunnen businessgebruikers zelf hun pagina samenstellen op basis van eigen eisen en wensen. Hierdoor wordt het aantal openstaande applicaties verminderd en behoudt men een goed overzicht van de lopende bedrijfsprocessen. Naast de getoonde oplossing als CRM-activiteiten kan uiteraard worden gedacht aan business-scorecards, BizTalk's BAM, enzovoort. De mogelijkheden zijn eindeloos. In een tijd waarin mensen steeds meer informatie te verwerken krijgen, is gepersonaliseerde content op de werkplek steeds belangrijker. Live.com biedt hierin een goede oplossing.

Rob Bakkers en Sebastian Hek zijn beiden werkzaam als consultant bij Avanade Nederland. (www.avanade.nl) Ze hebben veel ervaring opgedaan met Microsoft Server-producten, waaronder Microsoft CRM. Rob is ook MVP voor Microsoft CRM.

Referenties

- <http://www.live.nl>
- <http://dev.live.com>
- <http://www.start.com>
- <http://microsoftgadgets.com>
- <http://www.start.com/developer/>