Kim Nuyts en Alexander Schwinn

Ontwikkelen van mash-up's met open API's

EBAY API & MICROSOFT VIRTUAL EARTH API

De bekendste API's die het vaakst bij het 'mashen' worden gebruikt, zijn afkomstig van Google, Amazon, Yahoo!, AOL, eBay en Microsoft. We willen aantonen hoe gemakkelijk mash-up's kunnen worden ontwikkeld. Dit doen we aan de hand van een ASP.NET 2.0 (C#)-applicatie die de eBay API combineert met Microsoft Virtual Earth om objecten die op eBay worden aangeboden grafisch op een landkaart weer te geven.

De combinatie van deze twee API's biedt heel wat voordelen. Zo kun je bijvoorbeeld op zoek gaan naar objecten die in jouw buurt te koop worden aangeboden, waardoor je op de verzendkosten kunt besparen. Als je op zoek bent naar een nieuwe auto, kun je bijvoorbeeld starten met de aanbiedingen die zich in uw buurt bevinden. Op die manier kunnen alle auto's die op eBay worden aangeboden, aan de gewenste eigenschappen voldoen en in de nabije omgeving te vinden zijn, op een stadsplan worden weergegeven. Op basis van die gegevens kan dan een route worden uitgestippeld. Vooral voor auto's, vastgoed of objecten die moeilijk te verzenden zijn, is een lokale zoekopdracht uitermate geschikt. Hieronder worden de beide interfaces kort voorgesteld. Daarna vertellen we wat je moet doen om de API's te kunnen gebruiken.

De API van eBay

eBay biedt in het kader van het eBay-ontwikkelprogramma al sinds 2000 een set webservices aan. Deze kan vanaf 2005 door iedere ontwikkelaar kosteloos worden gebruikt. Via de webservices van eBay kunnen ongeveer alle functies worden aangeroepen die op de websites van eBay beschikbaar zijn. Voorbeelden hiervan zijn veilingen en advertenties plaatsen - AddItem, lopende veilingen bekijken - GetItem, en zoekopdrachten uitvoeren - GetSearchResults. De interface is internationaal waardoor via de eBay API alle internationale sites zoals ebay.com, ebay.be, ebay.nl, en ebay.de op een eenduidige manier kunnen worden gebruikt.

Registratie in het eBay-ontwikkelprogramma

Wie de eBay-API wil gebruiken en in de test- en productieomgeving van het eBay-platform wil werken, moet zich eerst voor het ontwikkelprogramma van eBay inschrijven (gratis) en vervolgens een toegangssleutel samenstellen. Hiervoor gaat u als volgt te werk:

- Schrijf je in voor het ontwikkelprogramma van eBay in op: http://developer.ebay.com
- Na de inschrijving ontvangt u een e-mail. Via de link die in de e-mail wordt vermeld, kunnen nieuwe sleutels voor de eBay Sandbox worden aangemaakt. De Sandbox (http://sandbox.ebay.com) is de testomgeving van eBay, waar je naar hartenlust kunt experimenteren (een voorbeeld is AddItem – iets aanbieden zonder daarvoor vergoedingen aan eBay te moeten betalen).

Om toegang te krijgen tot de eBay-API moet je over drie sleutels beschikken:

1. De applicatie-ID (*App-Id*): iedere applicatie die de eBay-API aanroept, krijgt een applicatie-ID toegewezen.
2. De developer-ID (*Dev-Id*): iedere ontwikkelaar die zich heeft ingeschreven voor het ontwikkelprogramma van eBay, krijgt een developer-ID toegewezen.
3. Certificaat (*Cert-Id*): naast de *App-Id* en de *Dev-Id* heb je een certificaat nodig om aan de API van eBay te werken.

De sleutels kunnen op ieder moment in het afgeschermde gedeelte van http://developer.ebay.com bekeken worden (onder *View Keys*). Naast de sleutels heb je ook nog een eBay-gebruikerstoken nodig, waarmee een lid van eBay geïdentificeerd kan worden via de eBay-API. Door gebruik te maken van dit tokensysteem hoeven er nooit eBay-gebruikersnamen en -wachtwoorden in een applicatie te worden opgeslagen. Gebruikerstokens kunnen automatisch door de eindgebruiker of handmatig met behulp van de token-tool (http://developer.ebay.com/tokentool) worden aangemaakt. Je kunt alleen een token gebruiken als je over een eBay-account beschikt. Op de Sandbox kunnen eBay-accounts voor leden worden aangemaakt met behulp van een specifieke tool (http://developer.ebay.com/DevZone/sandboxuser.asp). Opmerking: eBay-gebruikersaccounts, aangemaakt op de publieke websites van eBay, zijn niet geldig op de Sandbox-omgeving. Wie de eBay-webservices wil testen, kan gebruikmaken van de testtool (developer.ebay.com/DevZone/build-test/test-tool/). Kies eerst Sandbox en voer dan de *App-Id*, de *Dev-Id*, de *Cert-Id* en het token in. Vervolgens kiest u een call-template (bijvoorbeeld *GetUser*). Deze request wordt dan naar de volgende url's verzonden: https://api.sandbox.ebay.com/ws/api.dll voor de Sandbox en https://api.ebay.com/ws/api.dll voor de productieomgeving. Wanneer je de productieomgeving van eBay wilt gebruiken, moet je specifieke productiesleutels aanmaken. Daarvoor is wel eerst een applicatieregistratie (*Self Certification*) nodig. Na het beantwoorden van een aantal vragen over de applicatie, ontvang je de productiesleutels en kun je tot 10.000 webservice-requests per maand uitvoeren. Als dat niet voldoende is, kun je nog altijd via een ander formulier een *Standard Certification* aanvragen. Indien de technische support van eBay die aanvraag goedkeurt, kunnen tot 1,5 miljoen webservice-requests per dag worden gemaakt. Op http://developer.ebay.com/support/ vind je alles over de eBay-API, zoals documentatie, forums, support, SDK's en een knowledge base.

De eerste toepassing met de SDK van eBay voor .NET

De eBay-SDK voor .NET bevat een groot aantal libraries en voorbeelden (ASP.NET, C# en VB.NET) waarmee Windows-ontwikke-

laars zich gemakkelijk op een OO-manier een toegang kunnen verschaffen tot de eBay-API. De meest recente versie en documentatie van de SDK voor .NET vind je op http://developer.ebay.com/developercenter/windows/sdk. Wie nog niet over Microsoft Visual Studio beschikt, kan bij Microsoft de gratis versie 'Visual Web Developer Express Edition' downloaden. Na de installatie van de eBay-SDK voor .NET gaan we eerst een eenvoudig voorbeeld implementeren, dat op eBay naar actieve veilingen op zoek gaat. Hiervoor maken we eerst een nieuwe ASP.NET-website aan 'eBay-API Sample'(File | New | Web Site), die gebruikmaakt van C#. Vervolgens voegen we de nodige DLL's toe uit de SDK-installatie: *eBay.Service.dll*, *eBayPictureService.dll* en *Interop.MSXML2.dll*. Daarvoor maak je gebruik van Solution Explorer | rechtsklik op het Webproject | Add Reference. Houd er wel rekening mee dat de SDK van eBay gebruikmaakt van de .NET http-compressietechnologie. Daarvoor heb je het J# Redistributable Package (*vjslib*) nodig, dat op http://msdn2.microsoft.com/en-us/vjsharp/bb188598.aspx kan worden gedownload. In ons voorbeeld werd eerst een nieuwe Web Form toegevoegd; GetSearchResults.aspx. In GetSearchResults.aspx wordt voor de zoekopdracht een tekstveld *<asp:TextBox>* gebruikt, dat de ID txtQuery krijgt toegewezen. Daarna wordt een knop *<asp:Button>* met de ID btnSearch toegevoegd. Tenslotte worden nog twee labels (*asp:Label*) toegevoegd, waarmee het aantal zoekresultaten (lblResult) en eventuele fouten (lblError) worden weergegeven. De lay-out wordt gerealiseerd met behulp van een cascading style sheet die aan het project wordt toegevoegd. In codevoorbeeld 1 is het volledige resultaat van de *GetSearchResults.aspx*-pagina te zien.

```
<%@ Page="" Language="C#" AutoEventWireup="true"
CodeFile="GetSearchResults.aspx.cs" Inherits="GetSearchResults" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head>
    <title>eBay - GetSearchResults sample</title>
    <link rel="stylesheet" type="text/css" href="css/mashup.css"/>
  </head>
<body>
  <form id="form1" runat="server">
    <div id="header">
      <h1>eBay - GetSearchResults Sample</h1>
    </div>
    <!-- start query field-->
    <div id="finder">
      <table cellpadding="5" cellspacing="5" border="0"
           class="finder" width="500px">
        <tr>
          <td>
            <asp:TextBox ID="txtQuery" runat="server"
                 Width="200px"></asp:TextBox>
          </td>
          <td align="right">
            <asp:Button ID="btnSearch" runat="server"
                 OnClick="btnSearch_Click" Text="zoeken" />
          </td>
        </tr>
      </table>
    </div>
    <!-- end query field-->
    <!-- start results-->
    <div id="eBaySearchTotalSample">
      <asp:Label ID="lblResult" runat="server"></asp:Label>
    </div>
    <div id="eBaySearchErrorSample" style="left: 0px; top: 160px">
      <asp:Label ID="lblError" runat="server"
        ForeColor="#FF0000"></asp:Label>
    </div>
    <div id="eBaySearchResultSample"></div>
      <!-- end results-->
  </form>
</body>
</html>
```

Codevoorbeeld 1. De GetSearchResults.aspx-pagina

Om een zoekopdracht naar de eBay-API te kunnen verzenden, passen we GetSearchResults.aspx.cs aan. Als eerste worden de nodige references naar de eBay-SDK toegevoegd; zie hiervoor *using* bovenaan codevoorbeeld 3. Om het voorbeeld zo eenvoudig mogelijk te houden, gebeuren alle acties in de Click-eventhandler van de knop (btnSearch). Bij het samenstellen van de webservice-request moet je altijd een onderscheid maken tussen een algemeen en een call-specifiek gedeelte. In het algemene gedeelte vind je bijvoorbeeld informatie over het versienummer van de API, het token van de gebruiker en de taal waarin de foutmeldingen moeten worden teruggeven. Het call-specifieke gedeelte bevat dan weer invoerparameters zoals een zoekterm of een eBay-rubriek waarin moet worden gezocht. Bovendien moet de header worden gedefinieerd. Die bevat informatie zoals de url waarop de webservice-request wordt uitgevoerd (Sandbox- of productieomgeving), de toegangssleutel (*App-Id*, *Dev-Id* en *Cert-Id*) en een site met het overeenstemmende eBay-land; zie codevoorbeeld 2.

Eerst wordt een nieuw ApiContext-object aangemaakt. Hierin worden de nodige authenticatie- en autorisatiegegevens verwerkt zoals de drie toegangssleutels (voor Sandbox of productie) en het eBay-gebruikerstoken, de eBay-API-url (eveneens Sandbox of productie), de eBay-site die geconsulteerd wordt, en de huidige versie van de eBay-API (momenteel is dit 517). Let wel, als je van de testomgeving (Sandbox) naar de productieomgeving wil overstappen, moet je zowel de url als de drie toegangssleutels en het gebruikerstoken aanpassen. Het call-specifieke gedeelte van de *GetSearchResults*-call bestaat - in het voorbeeld dat hier wordt beschreven - alleen uit een zoekterm, die uit de textbox werd overgenomen. Met behulp van een eenvoudige foreach-lus worden de resultaten, die door eBay worden teruggestuurd, uitgelezen. Hierbij worden de velden *Title* (titel van de aanbieding), *ViewItemURL* (url naar de aanbieding op eBay), *BidCount* (aantal biedingen die al werden uitgebracht op deze veiling) en *CurrentPrice* (huidige prijs) gegeven, zie codevoorbeeld 3.

```
//set eBay ApiContext
eBay.Service.Core.Sdk.ApiContext objContext = new eBay.Service.Core.Sdk.
ApiContext();

// set the dev,app,cert information
objContext.ApiCredential.ApiAccount.Developer = "YOUR_DEVID";
objContext.ApiCredential.ApiAccount.Application = "YOUR_APPID";
objContext.ApiCredential.ApiAccount.Certificate = "YOUR_CERTID";

// set the AuthToken
objContext.ApiCredential.eBayToken = "YOUR_TOKEN";

// set the base SOAP URL
// use https://api.sandbox.ebay.com/wsapi for Sandbox calls
// use https://api.ebay.com/wsapi for production calls
objContext.SoapApiServerUrl = "https://api.ebay.com/wsapi";

// set the Site of the Context
objContext.Site = eBay.Service.Core.Soap.SiteCodeType.Belgium_Dutch;

//set WSDL Version
objContext.Version = "517";
```

Codevoorbeeld. 2. De webservice-request

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

using eBay.Service;
using eBay.Service.Core.Soap;
using eBay.Service.Core.Sdk;
using eBay.Service.Call;
using eBay.Service.Util;

public partial class GetSearchResults : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {

  }
  protected void btnSearch_Click(object sender, EventArgs e)
  {
    try
    {
      lblResult.Text = "";
      lblError.Text = "";

      //set eBay ApiContext
      eBay.Service.Core.Sdk.ApiContext objContext = new
      eBay.Service.Core.Sdk.ApiContext();

      // set the dev,app,cert information
      objContext.ApiCredential.ApiAccount.Developer = "YOUR_DEVID";
      objContext.ApiCredential.ApiAccount.Application = "YOUR_APPID";
      objContext.ApiCredential.ApiAccount.Certificate = "YOUR_CERTID";

      // set the AuthToken
      objContext.ApiCredential.eBayToken = "YOUR_TOKEN";

      // set the base SOAP URL
      // use https://api.sandbox.ebay.com/wsapi for Sandbox calls
      // use https://api.ebay.com/wsapi for production calls
      objContext.SoapApiServerUrl = "https://api.ebay.com/wsapi";

      // set the Site of the Context
      objContext.Site = eBay.Service.Core.Soap.SiteCodeType.Belgium_Dutch;

      //set WSDL Version
      objContext.Version = "517";

      //set Log-File
      ApiLogManager objLog = new ApiLogManager();
      objLog.ApiLoggerList.Add(new eBay.Service.Util.
      FileLogger("GetSearchResults.log", true, true, true));
      objLog.EnableLogging = true;
      objContext.ApiLogManager = objLog;

      //GetSearchResults Call
      eBay.Service.Call.GetSearchResultsCall objCall = new
      GetSearchResultsCall(objContext);

      //Define query
      string strQuery = txtQuery.Text.ToString();

      // call GetSearchResults
      SearchResultItemTypeCollection objResults = objCall.
      GetSearchResults(strQuery);

      lblResult.Text = "Resultaten voor de zoekopdracht: <b>" +
      strQuery + " </b>";

      Response.Write("<div id='eBaySearchResultSample'><table
      border='0' cellspacing='1' cellpadding='0' width='100%'>");
      foreach (SearchResultItemType objResult in objResults)
      {
        Response.Write("<tr bgcolor='#E6F0F0'><td><a href='");
        Response.Write(objResult.Item.ListingDetails.ViewItemURL.
        ToString());
        Response.Write("' target='_blank'>");
        Response.Write(objResult.Item.Title);
        Response.Write("</a>");
        Response.Write("<br>Huidig bod: <B> ");
        Response.Write(objResult.Item.SellingStatus.BidCount.
        ToString());
        Response.Write("</b> Huidige prijs: <b>");
        Response.Write(objResult.Item.SellingStatus.CurrentPrice.
        Value.ToString("c"));
        Response.Write("</b><br><br></td></tr>");
      }
      Response.Write("</table></div>");
    }
    catch (ApiException expAPI)
    {
      lblError.Text = expAPI.Message;
    }
    catch (SdkException expSDK)
    {
      lblError.Text = expSDK.Message;
    }
    catch (Exception exp)
    {
      lblError.Text = exp.Message;
    }
  }
}
```

Codevoorbeeld 3. Met behulp van een foreach-lus worden de resultaten, die door eBay worden teruggestuurd, uitgelezen.

De volledige documentatie over de eBay-webservices en de in- en uitvoerparameters van afzonderlijke calls vind je op http://developer.ebay.com/support/docs. Met behulp van de *ApiLogManager* kun je bovendien op een eenvoudige manier een logfile (*GetSearchResults.log*) aanmaken, die alle communicatie met de eBay-API netjes registreert. Nu is het tijd om de API van Microsoft Virtual Earth voor te stellen, voordat beide API's met elkaar worden 'gemasht'.

De API van Microsoft Virtual Earth

Met behulp van Microsoft Virtual Earth kan kaartinformatie in drie 2D-weergaven (landkaart, satellietbeeld en hybride) worden getoond. Ook biedt Virtual Earth voor verschillende Europese steden een vogelperspectief, waarmee je gedetailleerde stadsgezichten vanuit de vier windstreken kunt bekijken. Neem een kijkje op maps.live.com voor een uitgebreid overzicht van de functionaliteit. Sinds november 2006 is het bovendien mogelijk het materiaal van de kaarten driedimensionaal weer te geven. Hiervoor heb je wel een plug-in nodig, die gratis door Microsoft ter beschikking wordt gesteld. Momenteel bestaan er al 3D-versies van ongeveer vijftien grote steden in de VS. Tegen het einde van 2007 zijn er nog ongeveer 100 gepland. Daarnaast zijn er wereldwijd 3D-weergaven van landschappen beschikbaar. Hiermee kun je bijvoorbeeld een virtuele vlucht over de dalen van de Alpen of door de bergketens van de Himalaya maken. Virtual Earth wordt intussen al op een groot aantal platforms gebruikt voor de visualisering van informatie (zie dev.live.com/virtualearth/gallery/). Met behulp van de interactieve SDK's van Virtual Earth (dev.live.com/virtualearth/sdk/) kun je in slechts enkele minuten tijd een kaart in jouw website opnemen.
De API van Microsoft Virtual Earth is aan zijn vierde versie toe. In de interactieve SDK van Virtual Earth vind je ook een overzicht van alle beschikbare klassen.

Voorbeeld

Aan de hand van het volgende voorbeeld (VirtualEarth.aspx) wordt aangetoond hoe snel Virtual Earth in een website kan worden geïntegreerd. De API van Virtual Earth wordt als Javas-

```
function GetMap()
{
map = new VEMap('myMap');
map.LoadMap(new VELatLong(50.51, 4.21), 7 ,'r' ,false);
}
```

Codevoorbeeld 4.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
  <head>
    <title>Virtual Earth - Map sample</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
      <script src="http://dev.virtualearth.net/mapcontrol/v4/mapcon
      trol.js"></script>
      <script>
        var map = null;
        function GetMap()
        {
        map = new VEMap('myMap');
        map.LoadMap(new VELatLong(50.51, 4.21), 7 ,'r' ,false);
        }
      </script>
    </head>
  <body onload="GetMap();">
    <div id='myMap' style="position:relative; width:600px;
    height:400px;"></div>
  </body>
</html>
```

Codevoorbeeld 5.

```
//set CatID und sort function
objCall.CategoryID = "9801";
objCall.Order = SearchSortOrderCodeType.BestMatchSort;

//set price filter
objCall.PriceRangeFilter = new PriceRangeFilterType();
objCall.PriceRangeFilter.MaxPrice = new AmountType();
objCall.PriceRangeFilter.MinPrice = new AmountType();
objCall.PriceRangeFilter.MinPrice.Value = double.Parse(ddlMinPrice.Text.
ToString());
objCall.PriceRangeFilter.MaxPrice.Value = double.Parse(ddlMaxPrice.Text.
ToString());

//set distance
objCall.ProximitySearch = new ProximitySearchType();
objCall.ProximitySearch.PostalCode = txtPostalCode.Text.ToString();
objCall.ProximitySearch.MaxDistance = int.Parse(ddlRange.Text.ToSt
ring());

// define query
string strQuery = ddlMake.Text.ToString() + " " + txtModel.Text.ToSt
ring();
```

Codevoorbeeld 6. De query bestaat uit het gezochte automerk en de naam van het model

cript-klasse beschikbaar gesteld, die op de volgende manier in de header van de ASP.NET- of html-pagina wordt geïntegreerd:

```
<script src="http://dev.virtualearth.net/mapcontrol/v4/
mapcontrol.js"></script>
```

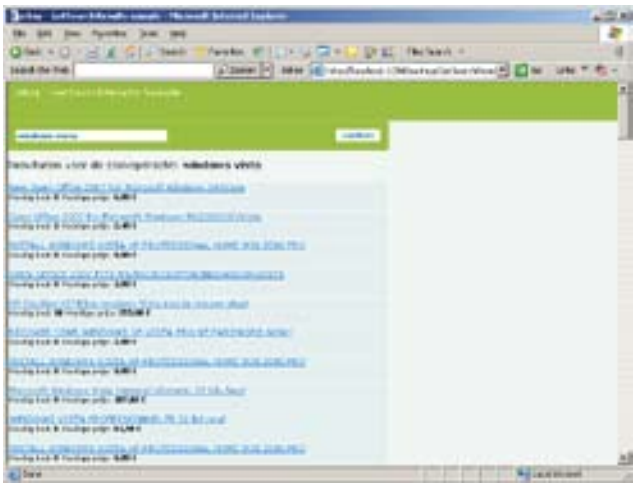
Om de kaart weer te geven, moet je met behulp van GetMap() een nieuwe kaart samenstellen en laden. Met behulp van *VELatLong* worden de coördinaten (in ons voorbeeld voor België), het zoomniveau, de stijl van de kaarten en het gedrag (statisch of dynamisch) vastgelegd; zie codevoorbeeld 4.
In de body wordt dan de Javascript-functie GetMap() bij het laden aangeroepen.

```
<body onload="GetMap();">
```

Vervolgens wordt de kaart op de pagina gepositioneerd door middel van een *div*-element met een resolutie van 600x400 pixels.

```
<div id='myMap' style="position:relative; width:600px; height:400px;
"></div>
```

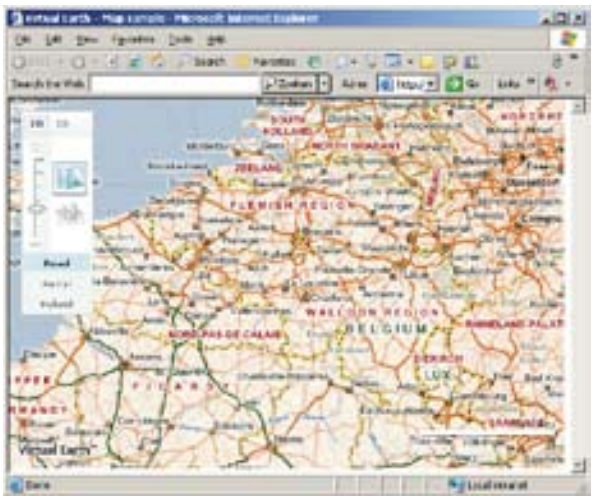
In codevoorbeeld 5 is het volledige voorbeeld te bekijken.



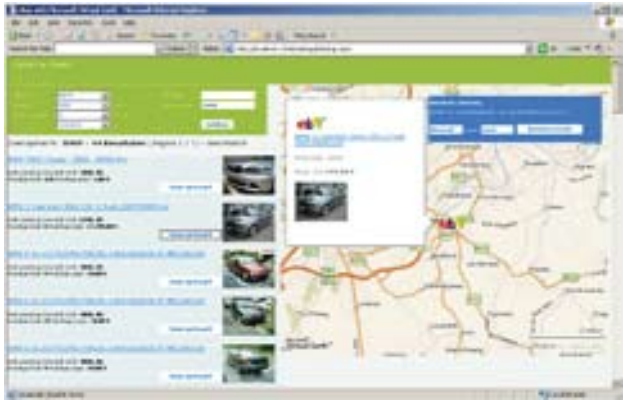
Afbeelding 1. GetSearchResults.aspx in actie

Tijd om te 'mashen': de Local Car Finder

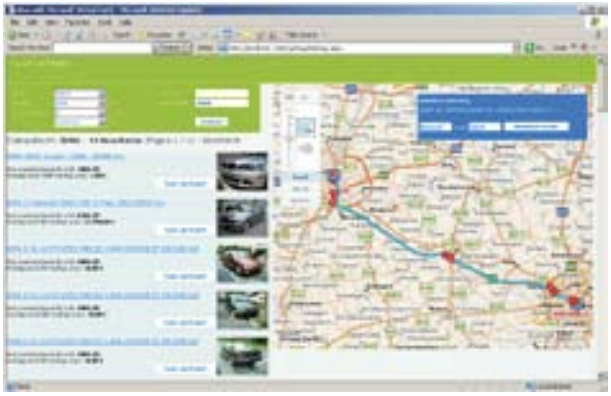
De beide voorbeelden die werden beschreven, worden nu uitgebreid en met elkaar gecombineerd. Hierbij wordt eerst de zoekfunctie uitgebreid met extra filtercriteria zoals automerk, model, afstand en prijs. Het bestand GetSearchResults.aspx wordt omgedoopt tot Mashup.aspx en daar wordt de zoekfunctie uitgebreid. Voor de afzonderlijke filtercriteria worden dropdownmenu's en tekstvelden aangemaakt. De waarden in de dropdownmenu's worden in dit voorbeeld handmatig ingevuld, maar je kunt de waarden ook via de eBay-API opvragen. De geselecteerde filtercriteria worden bij de *GetSearchResults*-call dynamisch meegegeven. In de eerste plaats wordt de eBay-rubriek 'Auto's' met de eBay-rubriek-ID 9801 toegevoegd, zodat er in de 'Car Finder' alleen maar auto's worden weergegeven. Een overzicht van alle rubriek-ID's vind je op http://listings.benl.ebay.be (klik op "Rubrieknummers weergeven"). De gevonden aanbiedingen worden volgens het BestMatch-principe weergegeven. Hierbij worden de eBay-aanbiedingen niet op basis van de resterende tijd van de veiling gesorteerd, maar op basis van hun relevantie. Deze BestMatch-relevantie wordt grotendeels bepaald door de historische activiteit van kopers op eBay. Vervolgens worden het prijsfilter (*PriceRangeFilter*) voor de minimum- en maximumprijs en de zoekomgeving (*ProximitySearch*) ingesteld. De query wordt uitgebreid en bestaat nu uit het gezochte automerk en de naam van het model, zie codevoorbeeld 6.



Afbeelding 2. Microsoft Virtual Earth in actie – België centraal



Afbeelding 3. Het resultaat van onze mash-up



Afbeelding 4. Kleine uitbreiding: een routeplanner

De resultatenlijst wordt met twee elementen uitgebreid. Een foto van de auto in kwestie, die door de API wordt teruggezonden, en de button 'Toon op kaart'. Met die button worden de waarden (postcode, titel, url van de veiling op eBay.be, prijs en foto) van het overeenstemmende voertuig ingevoerd, zodat die gegevens later op de landkaart kunnen worden getoond, zie codevoorbeeld 7.
De postcode is nodig om de vindplaats van het voertuig in Virtual Earth bij benadering te visualiseren. Met behulp van een concrete straatnaam zou je de zoekopdracht nog kunnen verfijnen, maar aangezien de aanbiedingen op eBay niet altijd straatnamen vermelden, beperken we ons hier tot de postcodes. Het voorbeeld van de kaart van België in Virtual Earth wordt nu aan het bestand *mashup.aspx* toegevoegd en uitgebreid. Eerst wordt *MapControl* van Virtual Earth geïntegreerd, waarbij de control door middel van een * * rechts naast de lijst met de zoekresultaten van eBay wordt gepositioneerd. Nu moet er nog een aantal aanpassingen aan de Javascript-code worden uitgevoerd om de informatie bij de afzonderlijke aanbiedingen op de kaart weer te geven. De button 'Toon op kaart' roept de Javascript-functie *FindLoc()* op; zie codevoorbeeld 8.
Met behulp van de *FindLocation*-methode wordt de correcte positie op de kaart gevonden. Hierbij wordt de postcode van de aanbieding op eBay met *BE* aangevuld. Nu zou er een pushpin met daarbij het eBay-logo op de kaart moeten worden weergegeven. De pushpin moet in het midden (*map.GetCenter()*) van de gevonden positie geplaatst worden, maar Virtual Earth plaatst altijd eerst de pushpin voor het naar de correcte positie gaat, waardoor de *AddPin*-functie door een *setTimeout* altijd met een korte vertraging wordt opgeroepen. Met de *AddPushPin*-methode wordt het gecreëerde *VEPushPin*-object overgenomen. In de constructor van het *VEPushPin*-object worden de ID, de coördinaten (of *map.GetCenter* in ons geval), het eBay-logo, de titel van de aanbieding en de informatie bij de aanbieding (zoals prijs, foto, enzovoort) weergegeven. De volledige brontekst van het bestand *mashup.aspx* vind je op www.microsoft.nl/netmagazine19.

De mash-up zou daarna nog kunnen worden uitgebreid met bijvoorbeeld een routeplanner, zodat je het traject kunt berekenen om de verschillende aangeboden voorwerpen te gaan bekijken.

```
<INPUT type='hidden' value='" + objResult.Item.PictureDetails.GalleryURL +
"'name='url'><INPUT id='find' type='button'
 value='Toon op kaart' name='find'
onclick='FindLoc(PostalCode.value, eBayTitle.value, eBayURL.value,
price.value,
url.value);'>
```

Codevoorbeeld 7. De resultatenlijst wordt met twee elementen uitgebreid

```
function FindLoc(PostalCode, eBayTitle, eBayURL, eBayPrice, purl)
{
var location = PostalCode +", BE";
//find location on map
map.FindLocation(location);
//add pin
setTimeout("addpin("+PostalCode+", '"+eBayTitle+"','"+eBayURL+"' ,
'"+eBayPrice+"', '"+purl+"')", 2000);
}

function addpin(PostalCode, eBayTitle, url, eBayPrice, purl)
{
var pin = new VEPushpin(pinID, map.GetCenter(),'images/eBay.
gif',null,'<img src=images/eBay.gif alt="Nu op eBay"><br><a href=' + url
 +' target=_blank>' + eBayTitle + '</a><br><br>Postcode: ' + PostalCode
 + '<br><br>Prijs:<b> ' + eBayPrice + '</b><br><br><a href='+ url +'
target=_blank><img src=' + purl + '></a>');
map.AddPushpin(pin);
pinID++;
}
```

Codevoorbeeld 8. De button 'Toon op kaart' roept de Javascript-functie FindLoc() op

Samenvatting

In dit artikel werd aangetoond hoe eenvoudig mash-up's kunnen worden ontwikkeld. In de toekomst zullen mash-up's een belangrijke rol spelen, aangezien het toenemende 'maatwerk' van webaanbiedingen en de overdracht van functies en content naar andere multimediakanalen zoals gsm's of televisietoestellen steeds meer combinatiemogelijkheden vereisen. De basis voor de verdere ontwikkeling van mash-up's wordt daarbij altijd gevormd door open en gratis toegankelijke interfaces. In de praktijk blijkt echter dat niet alle aanbieders dit soort interfaces in een eenvoudig toegankelijke vorm beschikbaar stellen.

Kim Nuyts is productmanager Local Development for eBay.be. Hij is bereikbaar op kim.nuyts@ebay.com

Alexander Schwinn is Manager Developers program for eBay.de. Hij is bereikbaar op alexander.schwinn@ebay.com

Referenties

eBay developer program: http://developer.ebay.com
eBay solutions directory: http://solutions.ebay.com
Microsoft MapPoint Developer Center: http://msdn.com/mappoint/
http://dev.live.com/virtualearth/sdk/Virtual-Earth-Interactive-SDK