

Windows Live Messenger Activiteiten

PEER-TO-PEER-APPLICATIES ONTWIKKELEN MET DE WINDOWS LIVE MESSENGER

Windows Live Messenger biedt meer mogelijkheden dan alleen maar chatten. Met behulp van Live Messenger Activity en de Windows Live Messenger API's kun je allerlei aardige applicaties bouwen. In dit artikel beschrijft de auteur hoe je op een eenvoudige manier een peer-to-peer-applicatie bouwt.

Stel dat je een applicatie wilt ontwikkelen die real-time data uitwisselt tussen twee gebruikers. De gebruikers kunnen zich overal ter wereld bevinden en de data moeten zo snel mogelijk van gebruiker A naar gebruiker B en vice versa. Hoe doe je dit zonder dat je je zorgen wilt maken over het opzetten van betrouwbare connecties, geen gebruik wilt maken van onpraktische pollingtechnieken en hoe ga je om met firewalls? Een mogelijke oplossing is het ontwikkelen van een Live Messenger Activity. De Windows Live Messenger biedt een API waarmee softwareontwikkelaars op een eenvoudige manier peer-to-peer-applicaties kunnen bouwen.

Historie

Windows Live Messenger is een instant messaging-client die door Microsoft wordt ontwikkeld sinds 1999. De eerste versies van de Messenger boden eenvoudige instant messaging-functionaliteit en de mogelijkheid tot het beheren van zogenaamde contactlists. Met de release van MSN Messenger 6 in 2003 werden de activiteiten geïntroduceerd. In de daarop volgende versies is het aantal features flink uitgebreid. In het kader van het Windows Live-initiatief werd de MSN Messenger in 2006 hernoemd naar Windows Live Messenger. De huidige versie van de Windows Live Messenger is 8.1. Het Messenger-netwerk kent wereldwijd meer dan 150 miljoen geregistreerde gebruikers en is met name in Nederland populair. In 2005 is Microsoft gestart met het openstellen van verschillende Messenger API's voor ontwikkelaars. Microsoft biedt ontwikkelaars daarnaast de mogelijkheid hun software te publiceren, zodat de eigen ontwikkelde applicaties voor alle Messenger-gebruikers beschikbaar zijn. Er is inmiddels een levendige community van Messenger-ontwikkelaars en deze wordt actief ondersteund en gestimuleerd door Microsoft.

API-overzicht

De huidige versie van de Windows Live Messenger bevat een uitgebreid scala aan programmeerinterfaces. De Activity API is een COM-programmeerinterface die het mogelijk maakt om real-time data uit te wisselen tussen twee gebruikers. Tijdens het voeren van een conversatie kunnen twee gebruikers samen een activity starten. De activity verschijnt naast het conversatiescherm in het Activity-window.

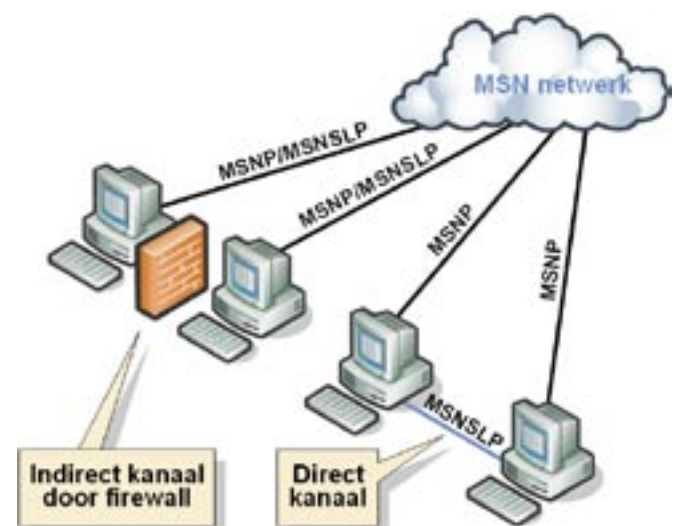
Naast de Activity API biedt Live Messenger een managed add-in API. Een add-in is een applicatie die automatisch taken uitvoert namens de gebruiker. Met een add-in kunnen bijvoorbeeld automatisch chatberichten worden beantwoord of de online status van de gebruiker worden gewijzigd.

Verder is het mogelijk chatbots te ontwikkelen voor Live Messenger. Een chatbot is een applicatie die via Live Messenger communiceert met de gebruiker, nadat deze is toegevoegd aan de contactlijst. Een bot kan opdrachten uitvoeren en conversaties voeren. Voor het ontwikkelen

van chatbots bestaan drie software development kits die zijn ontwikkeld door de externe partijen IM Interactive, Incesoft en Colloquis. Colloquis is recentelijk overgenomen door Microsoft. Ten slotte bevat Windows Live Messenger een COM API die vrijwel alle functies van de client-user-interface beschikbaar stelt aan ontwikkelaars. Met de API kun je zaken als het beheren van contactlists en het starten van IM-conversaties integreren in eigen applicaties.

Activity-framework

Als twee gebruikers een activity willen uitvoeren, wordt de activity gestart op beide machines. De initiële communicatie vindt plaats via het MSN-netwerk, waarna het framework een directe socket-connectie probeert te openen tussen beide partijen. Als dat niet lukt, vanwege bijvoorbeeld firewall-instellingen, wordt een indirect kanaal geopend dat via het MSN-netwerk loopt. Een indirect kanaal bestaat uit een socket-connectie met het MSN-netwerk. Wanneer het opzetten van een indirect kanaal mislukt, maakt Messenger gebruik van http-pollingtechnieken. Tijdens het draaien van de activity is het mogelijk dat een kanaal wijzigt van een direct naar indirect kanaal of vice versa. Het netwerkprotocol dat wordt gebruikt voor activity-communicatie is MSNSLP (een vorm van het Session Initiation Protocol) over MSNP (Mobile Status Notification Protocol). Beide protocollen zijn ontwikkeld door Microsoft en zijn niet openbaar. Zie afbeelding 1 voor een grafische weergave van directe en indirecte kanalen. De kracht van de Activity API is dat het opzetten en beheren van netwerkconnecties volledig is afgeschermd. De ontwikkelaar kan zich nu volledig focussen op het implementeren van applicatiefunctieiteit.



Afbeelding 1. Activity-kanalen



Afbeelding 2. Activity API

```
<?xml version="1.0"?>
<Entry>
  <EntryID>7</EntryID>
  <Name>My First Activity</Name>
  <Description>Een Activity API demo</Description>
  <URL>http://localhost/MyFirstActivity.htm</URL>
  <Height>500</Height>
  <Width>500</Width>
  <MinUsers>2</MinUsers>
  <MaxUsers>2</MaxUsers>
  ...
</Entry>
```

Codevoorbeeld 1.

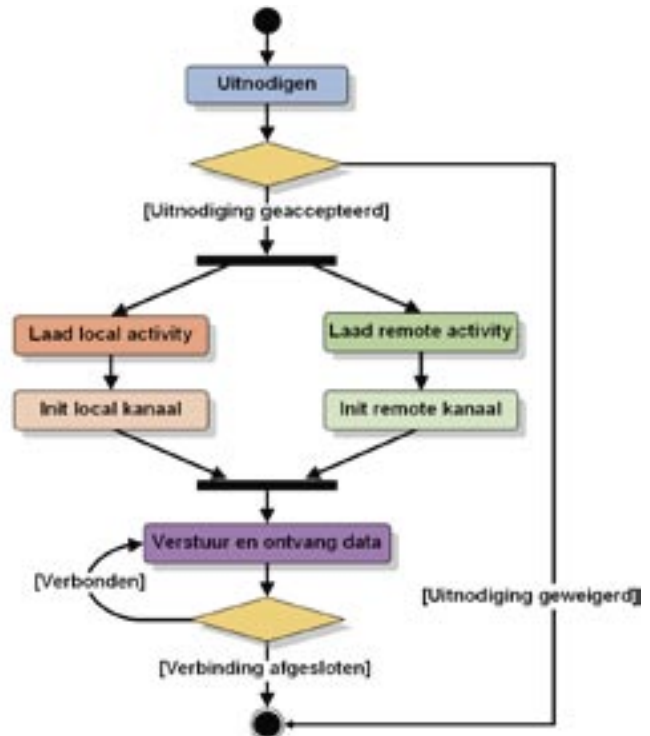
Een activity is een webpagina die draait in het Activity-window van de Messenger. De Messenger host een instantie van de Internet Explorer, en interactie met het Activity-framework vindt plaats via de Activity API. Een activity kan dezelfde functionaliteit verzorgen als een standaardwebpagina. Het gebruik van technologieën als Flash, ActiveX, ASP.NET- en Ajax-implementaties behoort tot de mogelijkheden. De configuratie van een activity wordt beschreven in een XML-document, MSGRP2P.xml genaamd. De configuratie bestaat uit een dertigtal eigenschappen en definieert zaken als naam, omschrijving, url van de Activity-pagina en verschillende permissie-instellingen.

Activity API

De Activity API is een eenvoudig objectmodel dat uit zeven COM-interfaces bestaat. Afbeelding 2 geeft het model grafisch weer. Toegang tot het objectmodel vindt plaats via de External-interface. De IChannel-interface vormt het hart van de Activity API. Deze interface definieert functies voor het initialiseren van een activity, het versturen en ontvangen van data en instant messages, en het registreren van fouten. De IMessengerUI-interface definieert verschillende methodes voor interactie met de messenger. De IUsers- en IUser-interfaces verschaffen gedetailleerde gebruikersinformatie. De Activity API is toegankelijk via het DHTML-objectmodel van de Internet Explorer. De property window.external biedt toegang tot het object dat de External-interface implementeert wanneer een activity in de Messenger wordt uitgevoerd.

Activity lifetime

Het state-diagram van een activity is weergegeven in afbeelding 3. Nadat een uitnodiging voor het starten van een activity is verstuurd en geaccepteerd, wordt de activity in beide Messenger-instanties geladen. Wanneer de activity is geladen, stuurt deze een initialisatiebericht naar de remote activity. Nadat de activity het initialisatiebericht van de remote activity heeft ontvangen, beginnen beide partijen met het versturen en ontvangen van data. Zodra een gebruiker uitloft of de activity afsluit, verstuurt de Activity API een afsluitbericht.



Afbeelding 3. Activity-state

Benodigheden

De minimale systeemeisen voor het uitvoeren van een activity is MSN Messenger 6.0 in combinatie met Internet Explorer 5.01. Voor het schrijven van een activity volstaat een eenvoudige tekst-editor als Notepad. Het is aan te raden om Visual Studio te gebruiken, zeker wanneer de activity met behulp van ASP.NET wordt ontwikkeld. Voor het opzetten van een ontwikkel- en testomgeving is nodig:

1. De Activity-applicatie bestaande uit een webpagina met alle bijbehorende plaatjes, stylesheets, enzovoort.
2. Een (lokale) webserver waar de activity over http bereikbaar is.
3. Twee computers waarop Messenger is geïnstalleerd.
4. De configuratie van de activity in de vorm van een MSGRP2P.xml-document in dezelfde folder als de Live Messenger client-executable (standaard %ProgramFiles%\MSN Messenger\) op beide machines. Het gebruik van een lokaal MSGRP2P.xml-document wordt alleen voor testdoeleinden gebruikt. Gepubliceerde activiteiten hebben geen lokale configuratie nodig.

Bij het opstarten van de Messenger-client wordt gecontroleerd of een MSGRP2P.xml-configuratiedocument aanwezig is. Als dit het geval is, verschijnt er een extra menu-item in het Actions-menu van het conversatiescherm en kan de activity worden gestart.

My first Activity

Om het een en ander te verduidelijken, wordt aan de hand van codevoorbeelden de Activity API toegelicht. De volledige sourcecode van deze activity is te downloaden van de .NET Magazine-site. Een deel van de activity-configuratie is weergegeven in codevoorbeeld 1 (let op: codevoorbeeld 1 is niet volledig). Elke activity krijgt een unieke EntryID toegewezen. Voor test- en ontwikkeldoeleinden wordt altijd EntryID 7 gebruikt. De url specificeert de locatie van de activity, in dit geval is dat de rootfolder van de localhost webserver. Uitgebreide documentatie van alle configuratie-instellingen is terug te vinden in de Messenger Activity SDK.

De voorbeeld-activity is een eenvoudige html-pagina en maakt gebruik van JavaScript. Voor het afhandelen van channel-events in JavaScript is een conventie opgesteld. Een JavaScript-functienaam bestaande uit Channel_[event naam] wordt aangeroepen als dat event wordt getriggered. Event-parameters worden niet ondersteund, maar waar nodig zijn de parameters als properties beschikbaar.

baar op het Channel-object. Het initialiseren van een activity wordt gedemonstreerd in codevoorbeeld 2. Nadat de activity is geladen, wordt eerst gecontroleerd of de Activity API beschikbaar is. Als dat het geval is, wordt de methode Initialize aangeroepen. Als er binnen tien seconden geen bericht binnenkomt dat de remote-instantie is geïntialiseerd, wordt er een connectie- time-outfout gegenereerd.

De data die via de activity API worden verstuurd en ontvangen, hoeven niet aan een bepaald formaat te voldoen. De ontwikkelaar is zelf verantwoordelijk voor het aanmaken en parsen van datastructuren. Het gebruik van XML is hier een mogelijkheid, zoals geïllustreerd in codevoorbeeld 3. In dit voorbeeld wordt gebruikt gemaakt van een XML-data island. Wanneer er veel kleine databerichtjes heen en weer worden gestuurd, kan het gebruik van XML een onevenredige hoeveelheid processortijd in beslag nemen. Voor elk binnenkomend bericht moet namelijk een XML-document worden gecreëerd en geparsed. Hanteer daarom de volgende vuistregel: gebruik eenvoudige strings voor chatty (veel en kleine berichten) dataverkeer en gebruik XML-documenten voor chunky (weinig en grote berichten) dataverkeer.

```
var _timeoutId;
var _initFailed = false;

function Activity_OnLoad(){
    // Check whether we're running in the Messenger
    if (window.external.Channel){
        // Activity API found, initialize channel
        window.external.Channel.Initialize();
        // Set initialize timeout
        _timeoutId = window.setTimeout(Activity_OnInitTimeout, 10000);
    }
    else{
        // No Activity API
        alert("Activity API unavailable");
    }
}

function Activity_OnInitTimeout(){
    _initFailed = true;
    alert("Activity Timeout");
}

// Remote activity loaded
function Channel_OnRemoteAppLoaded(){
    if (!_initFailed){
        // Clear initialize timeout
        window.clearTimeout(_timeoutId);

        // Start sending and receiving data
    }
}
```

Codevoorbeeld 2.

```
// Remote data received event handler
function Channel_OnDataReceived(){
    // Get the received data
    var data = window.external.Channel.Data;
    // Parse data using xml data island
    var xmlData = document.createElement("xml");

    if (xmlData.loadXML(data)){
        // Handle message
    }
    else{
        alert("Invalid data message received");
    }
}
```

Codevoorbeeld 3.

Tips

Als je een activity hebt ontwikkeld en deze wilt publiceren, kun je de activity officieel registreren bij Microsoft. De activity moet wel voldoen aan een aantal voorwaarden. Zaken als het gebruik van ActiveX, het uitwisselen van bestanden en het uitlezen van gebruikersgegevens zijn niet toegestaan zonder expliciete toestemming. Na een conformancetest door Microsoft wordt de activity toegevoegd aan de publieke Activity-registry, waarna de activity door elke Windows Live-gebruiker kan worden gebruikt. In de referenties is een url opgenomen van de website met alle details voor het officieel registreren en publiceren van activiteiten. Een activity leent zich uitstekend voor het gebruik van Ajax-toepassingen. Wanneer de Activity-webpagina is geladen, wordt deze niet meer ververst. Een Ajax-framework als Microsoft Ajax (voorheen Atlas) kan worden gebruikt voor het laden van data van de server. Hoewel het gebruik van ActiveX-objecten voor officieel geregistreerde activiteiten niet is toegestaan, biedt Internet Explorer 7 uitkomst met het gebruik van het native XmlHttpRequest-object. Om het MSN-netwerk te beschermen is de datatransfer-rate voor indirecte connecties gelimiteerd. Het maximum aantal berichten dat per minuut ontvangen en verstuurd kan worden is 120. Voor directe connecties wordt de maximale transfer-rate bepaald door de snelheid van de internetverbinding. De Activity API van de Windows Live Messenger ondersteunt momenteel maximaal twee gelijktijdige gebruikers. Mogelijk wordt dit in de toekomst uitgebreid.

Eccky

Een goed voorbeeld van een Messenger Activity en chatbot is het in Nederland ontwikkelde spel Eccky. Eccky is een online internet spel voor kinderen, waarbij de spelers samen een virtueel kindje maken en opvoeden in zes dagen. Tijdens die zes dagen kunnen de spelers met hun Eccky chatten, deze eten geven en naar de winkel gaan om kleding te kopen. Het doel van het spel is Eccky zo gelukkig mogelijk te maken door middel van het geven van aandacht. Eccky wordt vooral gespeeld in en via de Live Messenger. Nadat een Eccky is gemaakt, wordt deze toegevoegd aan de lijst van Messenger-contacts van de speler. De kunstmatige intelligentie van de chatbot verzorgt de geanimeerde conversaties en de Eccky Activity biedt de mogelijkheid het spel vanuit Live Messenger te starten.

Samenvatting

Met Windows Live Messenger Activities biedt Microsoft een laagdrempelige oplossing voor het ontwikkelen van peer-to-peer-applicaties. Activities zijn niet-technologie gebonden, het gebruik van eenvoudige html-pagina's tot geavanceerde Ajax-based ASP.NET-applicaties behoren tot de mogelijkheden. Elke ontwikkelaar kan activiteiten registreren en publiceren voor het publieke MSN-netwerk.

Koen Zwikstra is senior consultant bij Avanade Nederland (www.avanade.nl). Sinds 2003 heeft hij verschillende activiteiten ontwikkeld en mede door zijn participatie in het Eccky-project heeft hij veel ervaring opgedaan met het ontwikkelen van software voor de Live Messenger. Zijn e-mailadres is koenz@avanade.com.

Referenties

My First Activity source-code: www.microsoft.com/netmagazine16
Messenger Development Portal: <http://dev.live.com/messenger>
Activity-registratie en -publicatie: <http://gallery.live.com>
Windows Live Developer Center <http://msdn.microsoft.com/live/msnmessenger>
Bot SDK's: <http://www.robotinvaders.com/main/About.aspx>
Eccky: <http://www.eccky.nl> en <http://www.eccky.com>
Live AddIn example: <http://www.firstfloorsoftware.com/Products/LiveAddIn>