Xander Wemmers is productspecialist systeemontwikkeling en trainer bij Compu'Train BV

Excel: van client naar serverapplicatie

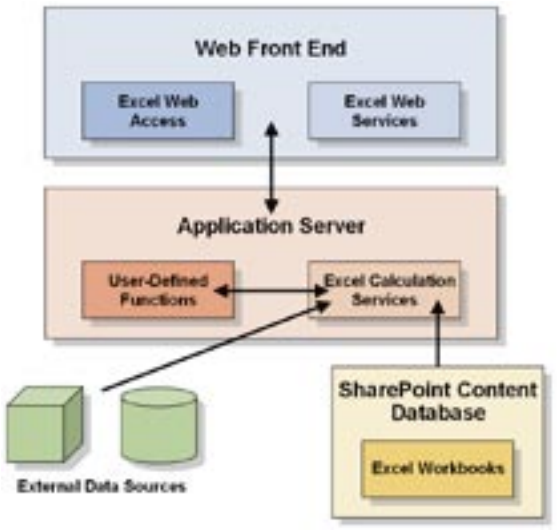
VANUIT ASP.NET BEREKENINGEN LATEN UITVOEREN DOOR EXCEL SERVICES

Microsoft Excel kent in elk bedrijf de meest uiteenlopende toepassingen. Daarbij wordt Excel voornamelijk gezien als een clientapplicatie: de auteur stuurt het document per mail rond, zodat iedereen zijn eigen versie ontvangt. De lezers kunnen hun eigen document wijzigen en zo ontstaan er 'multiple versions of the truth'. Behalve het probleem van de verschillende versies is het ook niet eenvoudig documenten goed te beveiligen. In 2007 Microsoft Office system is Excel uitgebreid met Excel Services. In dit artikel gaan we in op Excel Services en hoe deze te benaderen vanuit een ASP. NET-applicatie.

Excel Services is een serverapplicatie waarmee het veel eenvoudiger wordt om documenten te beheren en beveiliging te regelen. Als je SQL Server ziet als de grote broer van Access, dan is Excel Services de grote broer van Excel. Excel Services maakt het mogelijk de spreadsheet op een centrale plaats op te slaan en de berekeningen ook centraal uit te voeren. Gebruikers hebben daardoor altijd de recentste gegevens ter beschikking en de broncode van jouw Excel-sheet is beschermd tegen misbruik: alleen de resultaten van jouw berekeningen zijn zichtbaar. Ook kun je eventuele tussenresultaten in jouw spreadsheet verbergen voor de eindgebruiker. Een verschil met SQL Server is de relatieve eenvoud waarmee een spreadsheet wordt geschreven. De auteur schrijft zijn document gewoon in Excel 2007 en publiceert dat op de server. Vanaf dat moment hebben alle eindgebruikers direct de laatste versie van de gegevens tot hun beschikking.

Architectuur Excel Services

Excel Services is onderdeel van SharePoint Server 2007. SharePoint maakt het mogelijk Excel-documenten te beheren en te beveiligen. Om een Excel-sheet op de server te kunnen gebruiken, wordt gebruikgemaakt van Excel Services, dat uit de volgende drie componenten bestaat (zie afbeelding 1):



Afbeelding 1. Architectuur Excel Services

1. Excel Calculation Services voor het uitvoeren van berekeningen, het verversen van externe data en het beheer van sessies.
2. Excel Web Access voor het tonen van een spreadsheet in de webbrowser met behulp van dhtml.
3. Excel Web Services voor het programmatisch benaderen van een werkboek op de server.

Excel Web Access is een webpart in SharePoint Server 2007 waarmee een Excel-werkboek getoond wordt in de browser. De eindgebruiker heeft daarbij zelfs geen Excel-client meer nodig. Alles wat hij nodig heeft, is een internetbrowser en daardoor is het Excel-sheet toegankelijk. Excel Calculation Services laadt en berekent Excel-werkboeken. Evenals in een ASP.NET-webapplicatie wordt voor iedere gebruiker een sessie bijgehouden. Hoewel een gebruiker niet in staat is het Excel-document op de server aan te passen, kan hij wel zelf bepalen welke informatie op welke manier getoond wordt. Denk daarbij aan dropdown-boxen waarin de gebruiker een maand of regio kiest, waarna de bijbehorende verkoopcijfers getoond worden of een gebruiker die zijn eigen gegevens voor bijvoorbeeld een hypotheek opgeeft. Excel Calculation Services maakt het ook mogelijk User Defined-functies te maken en deze in de sheet te gebruiken.

Excel Web Services

Voor de ASP.NET-ontwikkelaar zijn de Excel Web Services de API om te communiceren met Excel Services. Terwijl Excel Web Access ons in staat stelt een werkboek in de browser te tonen, kunnen we via deze API de kracht van Excel en ASP.NET combineren. In dit artikel gaan we ervan uit dat het Excel-werkboek al is gepubliceerd in SharePoint Server. Let wel: het Excel-document moet in SharePoint zijn opgeslagen in een Trusted Location.
Om gebruik te maken van Excel Web Services moet een Web Reference worden toegevoegd, bijvoorbeeld naar http://moss.litwareinc.com/_vti_bin/ExcelService.asmx. In dit voorbeeld maken we gebruik van een Excel-sheet, waarin we maandelijkse hypo-

```
ExcelService es = new ExcelService();
es.Credentials = CredentialCache.DefaultCredentials;
Status[] stati;
```

Codevoorbeeld 1a.

```
this.Label1.Text = es.GetApiVersion(out stati);
```

Codevoorbeeld 1b.

theeklasten kunnen berekenen met behulp van een aantal door de gebruiker ingevoerde waarden. Zie ook het filmpje bij de link 'Using Excel Web Services in a SharePoint Web Part' onderaan dit artikel. Importeer de namespace System.Net en maak een Excel Services-object met DefaultCredentials: zoals te zien is in codevoorbeeld 1a.
De array stati bevat, in geval dat er iets fout gaat, een of meer foutmeldingen. In codevoorbeeld 1b is een eenvoudige test te zien of onze webservice goed werkt: de aanroep van de functie GetApiVersion.
Vervolgens kan een werkboek op de server worden geopend. Merk op dat het openen van een werkboek ook een nieuwe sessie op de server start. De aanroep van OpenWorkbook in codevoorbeeld 1c retourneert een sessie-ID die we later in onze code nodig hebben om onze sessie te identificeren.
Nu ons werkboek is geopend, kunnen we dit gebruiken. De methode SetCell verwacht achtereen de sessie-ID, de naam van de sheet, het rij- en het kolomnummer en ten slotte de waarde die in de cel wordt ingevuld. In codevoorbeeld 1d worden de cellen C3 tot en met C5 gevuld met een waarde die door de gebruiker in de website is opgegeven.
De wijziging in de Excel-sheet geldt overigens alleen voor onze sessie. Excel Services maakt voor onze sessie een aparte kopie van het werkboek en werkt daarmee. Andere gebruikers merken niets van onze wijzigingen. Er bestaat overigens een alternatief voor de functie SetCell, namelijk SetCellA1; zie codevoorbeeld 1e. Deze functie heeft hetzelfde effect, maar in plaats van concrete rij- en kolomnummers geven we aan deze functie een 'named range' mee. In Excel Web Services bestaat een aantal functies dat in beide varianten voorkomt.
In de Excel-sheet verwijst 'MortgageAmount' dus naar cel B1. Deze manier van programmeren is robuuster dan het werken met concrete rij- en kolomnummers. Na het invullen van het salaris is de maximale hypotheek in cel C6 terug te vinden; zie codevoorbeeld 1f.
En ook voor deze functie is er een alternatief met named ranges; zie codevoorbeeld 1g.
En zoals dat ook noodzakelijk is bij een databaseconnectie, moet ook een connectie met een Excel-werkboek netjes worden afgesloten; zie codevoorbeeld 1h.

```
string sessID = es.OpenWorkbook(
   @"http://moss.litwareinc.com/demo/Book1.xlsx", "", "", out stati);
```
Codevoorbeeld 1c.

```
stati = es.SetCell(
   sessID, "Sheet1", 3, 3,
   Convert.ToInt32(txtPrinciple.Text));

stati = es.SetCell(
   sessID, "Sheet1", 4, 3,
   Convert.ToInt32(txtInterest.Text));

stati = es.SetCell(
   sessID, "Sheet1", 5, 3,
   Convert.ToInt32(txtLength.Text));
```
Codevoorbeeld 1d.

```
stati = es.SetCellA1(sessID, "Sheet1", "MortgageAmount",
   Convert.ToInt32(txtAmount.Text));
```
Codevoorbeeld 1e.

```
decimal Antwoord = (decimal)es.GetCell(sessID,
   "Sheet1", 7, 3, false, out stati);
```
Codevoorbeeld 1f.

```
decimal Antwoord = (decimal)es.GetCellA1(sessID,
   "Sheet1", "Payment", false, out stati);
```
Codevoorbeeld 1g.

```
es.CloseWorkbook(sessID);
```
Codevoorbeeld 1h. Bovenstaande code vind je als geheel terug in codevoorbeeld 1.

Excel Web Access en dashboards

Business Intelligence is een groeiende markt en niet zonder reden. Bedrijven willen hun verkoop en productiecijfers graag op vele manieren inzichtelijk maken. Excel Services kan hierin samen met WebParts in SharePoint Server een belangrijke rol spelen. In SharePoint is het eenvoudig een dashboard op te zetten. Een dashboard is een webpagina waarin verschillende webparts informatie tonen uit een data warehouse. Het ligt voor de hand om ook hier Excel Services in te zetten en in het bijzonder Excel Web Access. In het artikel 'Excel Services Technical Overview' (zie de links onderaan dit artikel) wordt hier verder op ingegaan. Met Excel Web Access zijn we in staat een standaardweergave van een Excel-sheet te tonen. De eindgebruiker kan hierop nog wel enige invloed uitoefenen via parameters. De auteur van de spreadsheet moet dit dan zelf aangeven in Excel 2007: kies in het hoofdmenu voor Publish – 'Excel Services' en dan voor 'Excel Services Options'. Op het tabblad 'Parameters' kunnen een of meer parameters worden opgegeven. SharePoint zal deze herkennen en in de webpart kan de gebruiker zelf een waarde hiervoor opgeven.

Kracht en eenvoud

Met Excel Web Services is het mogelijk de kracht en eenvoud van het Excel-rekenprogramma ook aan de serverkant te gebruiken. De Excel-expert creëert naar eigen inzicht een rekenblad en publiceert

```
protected void Button1_Click(object sender, EventArgs e)
{
   ExcelService es = new ExcelService();
   es.Credentials = System.Net.CredentialCache.DefaultCredentials;

   Status[] stati;

   this.Label1.Text = es.GetApiVersion(out stati);
   string sessID = es.OpenWorkbook(
      @"http://moss.litwareinc.com/demo/Book1.xlsx", "", "", out stati);

   // Geef de cellen C3 t/m C5 een waarde.
   stati = es.SetCell(sessID, "Sheet1", 3, 3,
           Convert.ToInt32(txtPrinciple.Text));

   stati = es.SetCell(sessID, "Sheet1", 4, 3,
      Convert.ToInt32(txtInterest.Text));

   stati = es.SetCell(sessID, "Sheet1", 5, 3,
      Convert.ToInt32(txtLength.Text));

   // Functieaanroep met named range:
   //
   // stati = es.SetCellA1(sessID, "Sheet1","MortgageAmount",
   // Convert.ToInt32(txtAmount.Text));
   // etc..

   // Lees de maximale hypotheek uit cel C7.
   int Antwoord = (int)es.GetCell(sessID, "Sheet1",
      7, 3, false, out stati);

   // Functieaanroep met named range:
   // int Antwoord = (int)es.GetCellA1(sessID,
   // "Sheet1", "Payment" false, out stati);

   es.CloseWorkbook(sessID);
}
```
Codevoorbeeld 1.

dat op SharePoint Server. De ASP.NET-ontwikkelaar hoeft geen specifieke Excel-kennis in huis te hebben, maar kan gebruikmaken van een webservice om te communiceren met het rekenblad. Dit alles gebeurt in een omgeving, waarin de broncode van de spreadsheet voor de buitenwereld is beschermd.
Ook in Business Intelligence zal Excel Services veel gebruikt kunnen worden. Vooral met Excel Web Access, in combinatie met SharePoint-webparts, is het eenvoudig om snel een dashboard met relevante informatie te creëren.
We hebben in dit artikel gekeken hoe we op eenvoudige wijze verbinding kunnen maken met een spreadsheet op de server. Een eindgebruiker krijgt een eigen sessie toebedeeld, zodat wijzigingen aan de spreadsheet slechts voor hem zichtbaar zijn. Al het rekenwerk wordt zo overgelaten aan degene die er goed in is: Excel 2007.

Xander Wemmers is productspecialist systeemontwikkeling en trainer bij Compu'Train BV (www.computrain.nl)

Referenties

Excel Services Overview: http://msdn2.microsoft.com/en-us/library/aa972194.aspx
Using Excel Web Services in a SharePoint Web Part: http://msdn2.microsoft.com/en-us/library/aa973804.aspx
Walkthrough: Developing a Custom Application Using Excel Web Services: http://msdn2.microsoft.com/en-us/library/ms519100.aspx
Excel 2007 Blog: http://blogs.msdn.com/excel/