.net column

ADO.NET

In een service-georiënteerde architectuur zijn er diverse objecten verantwoordelijk voor het opslaan van data en het teruggeven van opgeslagen data. Het is natuurlijk wenselijk dat deze services in staat zijn om data terug te geven die zoveel mogelijk overeenkomt met de werkelijke databehoefte. Hiervoor wordt in ADO.NET een DataAdapter gebruikt.

De ADO.NET architectuur is opgebouwd uit de volgende componenten: Data Provider, Command, DataReader, DataAdapter en DataSet.

De eerste stap die benodigd is voor het ophalen of manipuleren van data is het maken van een connectie met de onderliggende dataprovider. Kiest men voor SQL Server 7.0 of hoger, dan kan er gewerkt worden met de managed .NET provider, die zonder tussenkomst van unmanaged code, rechtstreeks een connectie kan maken naar de SQL Server. Voor andere data providers kan er gebruik gemaakt worden van de managed OLEDB provider, die wel unmanaged code gebruikt en dus tot minder performance zal leiden.

Om een SQL statement of stored procedure uit te voeren wordt gebruik gemaakt van een Command object. Het Command object bevat een verzameling parameters, die meegegeven wordt aan de stored procedure. Op het command object zijn diverse methoden beschikbaar, waaronder de methode ExecuteReader, die een DataReader object terug geeft. De DataReader is connectie georiënteerd en kan gebruikt worden om door de rijen van de resultaatset te lopen. Tijdens het verwerken van de rijen uit de resultaatset is er een connectie naar de onderliggende data provider benodigd.

Om disconnected te kunnen werken biedt ADO.NET een zogenaamd DataSet object aan. Het fraaie van een DataSet is dat het gebruikt kan worden om precies die data terug te geven, die vanuit de databehoefte gewenst is. Om dit mogelijk te maken, biedt een DataSet een collectie van DataTable objecten aan. Een DataTable bevat weer een collectie van rijen, waarbij elke rij is onderverdeeld in een of meerdere kolommen, die mogelijk verschillende datatypen hebben. Het is zelfs mogelijk om tussen de verschillende tabellen in de DataSet relaties te definiëren. Hiervoor biedt de DataSet een collectie DataRelation objecten aan. Het is natuurlijk niet de bedoeling om een in-memory kopie van de gehele database te maken en in een DataSet op te slaan, dus nog steeds blijven goede database query's van groot belang. Het is nu wel mogelijk om bijvoorbeeld alle orderregels terug te geven die horen bij een geselecteerde order. De DataSet zou in dit voorbeeld kunnen bestaan uit twee tabellen, Order en OrderRegel met een relatie ertussen.

Een DataSet kan zichzelf opslaan in XML formaat, waarbij ook de structuur van alle DataTables en DataRelations in een XSD-schema wordt vastgelegd. Ook kan een nieuw DataSet object worden gereconstrueerd vanuit een XML document.

Een DataAdapter wordt gebruikt als intermedair tussen de Data Provider en de DataSet. Zo is de DataAdapter niet alleen verantwoordelijk voor het vullen van een tabel binnen de DataSet, maar zorgt er ook voor dat wijzigingen in de DataSet teruggeschreven worden naar de onderliggende Data Provider.

ADO.NET biedt de mogelijkheid om connected te werken via een DataReader en disconnected via een DataSet. De DataSet maakt het mogelijk dat verschillende services in de architectuur precies die data opleveren, die past bij de data behoefte, waarbij onderlinge relaties tussen gegevens behouden blijven. Een DataSet kan eenvoudig worden opgeslagen in XML formaat, met behoud van de structuur van de tabellen, relaties en constraints.

Ing. Xander Buffart is werkzaam bij Info Support als IT-Architect.