FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Labo frameworks voor serverapplicaties
ADO.NET

20 november 2024

1 Inleiding

Tijdens dit labo ontwikkelen we een datalaag die het mogelijk maakt om gegevens van een winkel uit een SQL-databank te halen. De applicatie wordt ontwikkeld m.b.v. ADO.NET in Visual Studio 2022.

2 Opzetten van de databank

In dit labo gebruiken we een SQL Express als lokale databaseserver waarbij we een mdf-bestand als databank gebruiken. De installatie verloopt als volgt.

- Download de database, gegeven onder vorm van het bestand classicmodels.mdf, van Ufora (je vindt het onder deze opgave). Deze databank heeft dezelfde structuur als de databank die we gebruikten voor het labo JDBC.
- Deze voeg je toe aan Visual Studio op de volgende manier:
  - Open Server Explorer (dit staat meestal ergens links of vind je via het menu View)
  - Klik rechts op Data Connections
  - Add Connection ...
  - Data Source moet de waarde "Microsoft SQL Server Database File (SqlClient)"hebben
  - Browse naar classicmodels.mdf.
  - Use Windows Authentication
  - Als er gevraagd wordt of de database een update mag ondergaan, stem je in.
- Je kan de inhoud van de databank bekijken in Visual Studio.
- Je kan SQL-opdrachten uitvoeren in Visual Studio.

3 Deel 1: DataReaders

Maak een console-applicatie Winkel die onderstaande acties uitvoert op de databank. Je kan een onvolledige versie van de applicatie downloaden (`Winkel.zip`), en deze aanvullen met de nodige code. Deze Visual Studio Solution bevat een consoleprogramma in `Program.cs` en een aanzet voor de datalaag (`DataStorage`, `DataStorageMetReader`, `Customer`, `Order` en `OrderDetail`).

- Een lijst van alle customers tonen. (Dus een methode met returntype List<Customer> kan handig zijn.)
- Een nieuwe customer toevoegen aan de databank.
- Een nieuw order inclusief orderdetails toevoegen aan de databank. Zorg ervoor dat wanneer het toevoegen van het order met orderdetails mislukt, de databank terugkeert naar de toestand voor het uitvoeren van de query.

De applicatie moet voldoen aan volgende eisen:

- Geschreven in C#.
- Het ADO.NET gedeelte maakt gebruik van DataReaders, niet van DataAdapters.
- Alle nodige query's komen uit een configuratiebestand.
- Zorg voor een goede foutafhandeling.
- Zorg dat je connectie steeds correct wordt afgesloten.

Tips

- De indentatie goed zetten doe je aan de hand van de toetsencombinatie Ctrl K gevolgd door Ctrl D.
- Hulp krijgen bij het oplossen van fouten gaat met de toetsencombinatie Ctrl ;.
- Allicht vindt Visual Studio de klasse System.Configuration niet. In het venster Solution Explorer klik je rechts op Dependencies, je kiest voor "Manage NuGet Packages" en zoekt naar System.Configuration. Eventueel zelfde opmerking voor System.Data.
- De database refreshen (in het venster Server Explorer) werkt niet altijd. In dat geval sluit je op de database zelf (in het snelmenu) de connectie, om ze daarna weer te openen.
- Bekijk de SqlException-foutmeldingen, ze geven veel info!
- Als bepaalde elementen in de databank ontbreken, is hun waarde in de reader gelijk aan de constante System.DBNull. Vergeet je hierop te controleren, dan krijg je foutmeldingen. (Controleer enkel als de waarde nul kán zijn.)
- Overbodige opmerking: vermijd overbodige regels code.

4 Deel 2: DataAdapters

We breiden het console-project uit deel 1 uit. De klasse DataStorageMetReader werd al geïmplementeerd. Nu wordt de klasse DataStorageMetDataTable gevraagd. Deze klasse zal een DbDataAdapter gebruiken om Customer-informatie aan de databank toe te voegen, te wijzigen en te verwijderen. Gebruik een aparte klasse voor al de functionaliteit van de DbDataAdapter,

de functionaliteit van de DataTable staat in de klasse DataStorageMetDataTable (we werken maar met één tabel uit de databank, dus geen DataSet nodig).

Voorzie volgende methodes in de nieuwe DataStorageMetDataTable-klasse.

- Een methode die een klant met bepaalde nummer uit de databank zal verwijderen (bij de eerstvolgende update, die hier echter nog niet uitgevoerd wordt).
- Een methode die een klant aan de databank zal toevoegen (bij de eerstvolgende update, die hier echter nog niet uitgevoerd wordt).
- Een methode die een klant met bepaalde nummer zal wijzigen (bij de eerstvolgende update...). Deze methode heeft drie parameters: een string die alle te wijzigen velden bevat, een string die alle in te vullen nieuwe waarden bevat en tenslotte het nummer van de klant. Een voorbeeld van de eerste twee parameters: “phone;addressLine1;city” en “04569856;Erling Skakkes gate 78;Stavern”.
- Een methode die alle voorlopige wijzigingen definitief maakt.
- Een methode die een lijst van Customer-objecten teruggeeft. Hierbij voorzie je twee versies: ofwel stemmen de objecten overeen met de informatie zoals momenteel in de databank aanwezig, ofwel weerspiegelen de objecten ook al de informatie die ondertussen toegevoegd/gewijzigd/aangepast werd, maar nog niet voor definitieve update naar de database doorgestuurd werd.

Vul het consoleprogramma aan om deze functionaliteit uit te testen.