

Visual C# .Net using framework 4.5

Eng. Mahmoud Ouf

Lecture 09

Working with Connection

1. Before dealing with database, you have to establish a session with the database server
2. This is done by making an object from connection, which is an instance of a class that implements the `System.Data.IDbConnection` interface for a specific data provider.
3. The name of the Connection class differ from data provider to another.
4. We have 2 data providers:

 Sql server (using `System.Data.SqlClient`)

 OleDb (using `System.Data.OleDb`)

The Connection class for SQL Server (`SqlConnection`)

The Connection class for MS-Access (`OleDbConnection`)

Working with SqlConnection

Create an object from SqlConnection, it has the following constructors

SqlConnection(), Initializes a new instance of the SqlConnection class.

SqlConnection(String), Initializes a new instance of the SqlConnection class when given a string that contains the connection string.

You can either send the connection string to the constructor when creating the object, or set the property named ConnectionString

The SqlConnection has the following properties:

ConnectionString

ConnectionTimeout

Database

DataSource

Open() method

Close() method

Working with SqlConnection

Connection String Parameters:

Name	Alias	Default	Description
Connect Timeout	Connection Timeout	15	Seconds to wait to connect
Data Source	Server		Name of the target SQL Server instance
Encrypt		false	Whether to use SSL encryption
Initial Catalog	Database		Database name
Integrated Security		false	Authentication mode

Working with Connected Model

The Data Provider:

Data provider is a set of related component that work together.

.Net has 2 data provider:

Sql server (using System.Data.SqlClient)

OleDb (using System.Data.OleDb)

Establish a connection using OleDbConnection:

Create object from OleDbConnection

```
OleDbConnection cn = new OleDbConnection();
```

Set the ConnectionString property:

```
cn.ConnectionString = "Provider = ....." + "Data Source=.....";
```

Provider: Microsoft.Jet.OLEDB.4.0 (access)

SQLOLEDB

(sql server below 6.5)

Open Connection

```
cn.Open();
```

Working with Connected Model

Building a SQL command using OleDbCommand:

Create an object from OleDbCommand

```
OleDbCommand m_Com = new OleDbCommand();
```

Create the SQL Command:

```
string str = "Select Make from Inventory where color = red";
```

Set the OleDbCommand object properties

```
m_Com.Connection = cn;
```

```
m_Com.CommandText = str;
```

Open the connection and execute the command:

ExecuteReader: Return an instance of OleDbDataReader (select)

ExecuteScalar: Return a single value from database query (aggregate function)

ExecuteNonQuery: command with no return value (insert, update, delete)

Working with Connected Model

Working with OleDbDataReader:

Provide a forward only, readonly connected record set.

It can't be instantiated, it comes from the execution of ExecuteReader method

It only return one record at a time in the memory (we have to iterate over it)

```
OleDbDataReader myReader;
```

```
myReader = m_Com.ExecuteReader();
```

```
While(myReader.Read())
```

```
{
```

```
    Console.WriteLine("Red Car:" + myReader["Make"].ToString());
```

```
}
```

```
myReader.Close()
```

```
Cn.Close();
```