TP
ADO.net & SQL Server
Accès à SQL Server via C#

Plan :

I. Introduction
II. Chaîne de connexion
III. Connexion à la base de données
IV. Commandes SQL
V. Déconnexion de la base de données

VI. Exemple

I. Introduction

Dans ce TP, nous allons voir comment insérer et lire les données à partir d'une base de données SQL Server. Ce code devrait fonctionner à la fois sur SQL Server, j'utilise 2008 R2. J'utilise Visual Studio 2010, mais cela devrait fonctionner avec Visual Studio 2013.

II. Chaîne de connexion

La première étape est d'ajouter l'espace de nom SQL du client :

```
using System.Data.SqlClient;
```

Ensuite, nous créons une SqlConnection et en spécifiant la chaîne de connexion :

```
SqlConnection maConnection = new SqlConnection("Persist Security Info=False; Trusted_Connection=True;
database=XXX; server=(local)");
```

La chaîne de connexion est simplement une compilation des options et des valeurs pour spécifier comment se connecter. Dans ce TP, je n'ai pas inclus toutes les options pour la chaîne de connexion, mais en voici certaines :

ID de l'utilisateur
L'User ID est utilisé lorsque vous utilisez l'authentification SQL. Dans notre TP, cela est ignoré lorsque vous utilisez une Trusted_Connection, ou l'authentification Windows.

Si le nom d'utilisateur est associé à un mot de passe : Password ou Pwd sera utilisé.

Mot de passe

Le mot de passe doit être utilisé avec l'ID de l'utilisateur, il n'aurait tout simplement pas de sens de se connecter sans un nom d'utilisateur, juste un mot de passe.

" Password=validpassword;" **-ou-** " Pwd=validpassword;"

Source de données ou serveur ou adresse ou Adr ou Network Address

En regardant dans la documentation MSDN j'ai trouvé qu'il y a plusieurs façons de spécifier l'adresse réseau. La documentation ne mentionne pas de différences entre eux et ils semblent être interchangeables. L'adresse est une adresse réseau valide.

" Data Source=localhost;" **-ou-** " Server=localhost;" **-ou-** " Address=localhost;" **-ou-** " Addr=localhost;" **-ou-** " Network Address=localhost;"

Sercurité intégrée ou Trusted_Connection

Integrated Security et Trusted_Connection sont utilisés pour spécifier si la connexion est sûr, comme l'authentification Windows ou SSPI. Les valeurs reconnues sont true, false.

Connect Timeout ou de délai de connexion

Ceux-ci précisent le temps, en secondes, d'attendre une réponse du serveur avant de générer une erreur. La valeur par défaut est 15 (secondes).

" Connect Timeout=10;" **-ou-** " Connection Timeout=10;"

Initial Catalog ou base de données

Initial Catalog et Database sont tout simplement deux façons de sélectionner la base de données associée à la connexion.

" Inital Catalog=main;" **-ou-** " Database=main;"

III. Connexion à la base de données

Il s'agit de la dernière partie pour se connecter et est simplement exécutée par le code suivant (n'oubliez pas de vous assurer que votre connexion a une chaîne de connexion en premier) :

```
try
{
    maConnection.Open();
}
catch(Exception e)
{
    Console.WriteLine(e.ToString());
}
```

SqlConnection.Open() est une fonction ne retournant rien (void), même pas de message d'erreur. Par contre, elle peut lever une exception alors n'oubliez pas de le mettre dans un try / catch.

IV. Commandes SQL

SqlCommand :

Une SqlCommand a besoin d'au moins deux choses pour fonctionner. Une chaîne de commandement, et une connexion. Il y a deux façons de spécifier la connexion, les deux sont illustrés ci-dessous :

```
SqlCommand maCommand = new SqlCommand("Requête SQL" de type string, maConnection);
// - or -
maCommand.Connection = maConnection;
```

La chaîne de connexion peut aussi être spécifiée dans les deux sens à l'aide de la propriété SqlCommand.CommandText.
Regardons maintenant notre première SqlCommand. Pour faire simple, ce sera une commande INSERT.

```
SqlCommand maCommand= new SqlCommand("INSERT INTO  table (Column1, Column2) " + "Values ('string', 1)", maConnection);
// - or -
maCommand.CommandText = "INSERT INTO table (Column1, Column2) " + "Values ('string', 1)";
```

Maintenant, nous allons jeter un oeil à ces valeurs. « table » est tout simplement la table dans la base de données. « Column1 » et « Column2 » sont simplement les noms des colonnes. Dans la section des valeurs, j'ai démontré comment insérer un type string et une valeur int. La valeur de la chaîne est placé entre guillemets simples et comme vous pouvez voir un nombre entier est juste indiqué tel quel. La dernière étape consiste à exécuter la commande avec :

```
maCommand.ExecuteNonQuery();
```

SqlDataReader :

Pour obtenir des données : SqlDataReader.
Non seulement avez-vous besoin d'un lecteur de données, mais vous avez besoin d'une SqlCommand.
Le code suivant montre comment mettre en place et exécuter un simple lecteur :

```
try
{
    SqlDataReader myReader = null;
    SqlCommand maCommand = new SqlCommand("select * from table", maConnection);
    myReader = maCommand.ExecuteReader();
    while(myReader.Read())
    {
        Console.WriteLine(myReader["Column1"].ToString());
        Console.WriteLine(myReader["Column2"].ToString());
    }
}
catch (Exception e)
{
    Console.WriteLine(e.ToString());
}
```

Comme vous pouvez le voir le SqlDataReader n'accède pas à la base de données, il se contente de renvoyer les données et fournit une interface facile pour utiliser les données. Comme il ya une très forte probabilité qu'il y ait plus d'une ligne en retour, une boucle while est nécessaire pour récupérer tous les enregistrements. Et comme toujours, vous utilisez le try et catch pour gérer les exceptions.

SqlParameter :

Il y a un petit problème avec l'utilisation d'une SqlCommand. Elle laisse un grand trou de sécurité. Par exemple, si la chaîne de commande était construite avec une saisie de l'utilisateur :

```
SqlCommand maCommand = new SqlCommand("SELECT * FROM table WHERE Column = " + input.Text,
maConnection);
```

Si l'utilisateur utilise une syntaxe correcte, pas de problème. Par contre, que se passe t-il si l'utilisateur saisi value1, DROP table. Meilleur scénario, il provoquera une exception, le pire des cas, votre table est supprimée.

Vous pouvez soit faire un contrôle de saisie des entrées de l'utilisateur et vérifier tout ce qui pourrait causer des problèmes ou vous pouvez utiliser un SqlParameter.

La classe SqlParameter est assez importante, mais je vais juste vous montrer une utilisation de paramètres de base.
Fondamentalement, vous avez besoin de trois choses pour créer un paramètre. Un nom, type de données, et la taille.

```
// Création et déclaration des paramètres
maCommand.Parameters.Add(new SqlParameter("@Param1", SqlDbType.NVarChar, 11);
maCommand.Parameters.Add(new SqlParameter("@Param2", SqlDbType.Int, 4);

// Attribution des valeurs aux paramètres
maCommand.Parameters["@Param1"].Value = <nom variable de type string> ;
maCommand.Parameters["@Param2"].Value = Convert.ToInt32(<nom variable de type string>);
```

Maintenant, voyons comment utiliser un paramètre.

```
SqlCommand maCommand = new SqlCommand("SELECT * FROM table WHERE Column = @Param2",
maConnection);
```

V. Déconnexion de la base de données

Ne pas oublier de se déconnecter de la base de données lorsqu'on a fini de travailler dessus.

Pour fermer une connexion il suffit d'appeler SqlConnection.Close(), mais n'oubliez pas de le mettre dans try / catch parce que, comme SqlConnection.Open(), elle ne renvoie pas d'erreurs, mais lève une exception à la place.

```
try
```

```
{
    maConnection.Close();
}
catch(Exception e)
{
    Console.WriteLine(e.ToString());
}
```

VI. Exemple

Exécutez le script ci-dessous sous SQL Server 2008 R2 pour obtenir :

- ✓ Une base de données nommée : TP_ADO
- ✓ Une table dans cette BD nommée : T_Clients
- ✓ 2 champs dans cette table nommés : ID_client, Nom_client de types respectifs : int, NvarChar(50). ID_client ne pouvant être « Null ».

```
-- Suppression d'une ancienne base
DROP DATABASE TP_ADO;
GO

-- Création de la base de données
CREATE DATABASE TP_ADO;
GO

-- Indique la BDD à utiliser et dans laquelle on exécute les requêtes
USE TP_ADO;
GO

-- Création de la table et création et paramétrage des champs
CREATE TABLE T_Clients
(
    ID_client int NOT NULL,
    Nom_client nvarchar(50) NULL
);
GO
```

Créez un nouveau projet « Console » sous Visual Studio 2010 et insérer y le code suivant dans le fichier Program.cs (ou récupérez le projet complet dans l'atelier).

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;

namespace TP_ADO_SQL
{
    class Program
```

```
    {
        // Variable qui gère la sortie de l'application
        private static bool reponse = true;

        // Compteur du nombre de saisie(s)
        private static int cpt = 0;

        static bool saisie()
        {
            // Chaîne de connexion SQL Server
            SqlConnection maConnection = new SqlConnection("Persist
Security
Info=False;Trusted_Connection=True;database=TP_ADO;server=(local)");

            try
            {
                // Connexion à la base de données
                maConnection.Open();
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }

            string idi;
            int id;
            string nom;

            Console.Clear();
            Console.WriteLine("");
            Console.WriteLine("-------------------------------------
-----------------------------------------");
            Console.WriteLine("|                                Module de
saisie en BD                                |");
            Console.WriteLine("-------------------------------------
-----------------------------------------");
            Console.WriteLine("");

            Console.Write("Voulez-vous faire une saisie dans la base
de données (o ou n) ? ");
            string rep = Console.ReadLine();
            while (rep != "o" && rep != "n")
            {
                Console.Write("Erreur de saisie ! Veuillez, s'il
vous plait, saisir 'o' ou 'n' ! ");
                rep = Console.ReadLine();
            }
            Console.WriteLine("");

            while (rep == "o")
            {
                cpt = cpt + 1;
                Console.Write("Veuillez saisir un identifiant entier
: ");
                idi = Console.ReadLine();
```

```
                while (int.TryParse(idi, out id) == false)
                {
                    Console.Write("Erreur de saisie ! Veuillez, s'il
vous plait, saisir un entier ! ");
                    idi = Console.ReadLine();
                }
                Console.Write("Veuillez saisir un nom : ");
                nom = Console.ReadLine();

                // Requete mise a jour de la table T_Clients
                String maRequete = "INSERT INTO T_Clients
(ID_client, Nom_client) Values (@ID_client, @Nom_client)";

                // Création de la commande SQL
                SqlCommand maCommand = new SqlCommand(maRequete,
maConnection);

                // Création et déclaration des paramètres
                maCommand.Parameters.Add(new
SqlParameter("@ID_client", System.Data.SqlDbType.Int, 4));
                maCommand.Parameters.Add(new
SqlParameter("@Nom_client", System.Data.SqlDbType.NVarChar, 50));

                // Attribution des valeurs aux paramètres
                maCommand.Parameters["@ID_client"].Value =
Convert.ToInt32(id);
                maCommand.Parameters["@Nom_client"].Value = nom;

                try
                {
                    // Exécution de la requete
                    maCommand.ExecuteNonQuery();
                    Console.WriteLine("");
                    Console.WriteLine("Saisie n°" + cpt + "
enregistrée en base de donnée !");
                    Console.WriteLine("");
                }
                catch (Exception e)
                {
                    Console.WriteLine(e.ToString());
                }

                Console.Write("Voulez-vous re-faire une saisie dans
la base (o ou n) ? ");
                rep = Console.ReadLine();
                while (rep != "o" && rep != "n")
                {
                    Console.Write("Erreur de saisie ! Veuillez, s'il
vous plait, saisir 'o' ou 'n' ! ");
                    rep = Console.ReadLine();
                }
                Console.WriteLine("");
            }

            try
```

```
            {
                // Deconnexion de la base de données
                maConnection.Close();
            }
            catch (Exception e)
            {
                Console.WriteLine(e.ToString());
            }

            return reponse;
        }

        static bool lecture()
        {
            Console.Clear();
            Console.WriteLine("");
            Console.WriteLine("----------------------------------
----------------------------------------");
            Console.WriteLine("|                                Module de
lecture en BD                              |");
            Console.WriteLine("----------------------------------
----------------------------------------");
            Console.WriteLine("");

            Console.Write("Voulez-vous faire une lecture dans la
base de données (o ou n) ? ");
            string rep = Console.ReadLine();
            while (rep != "o" && rep != "n")
            {
                Console.Write("Erreur de saisie ! Veuillez, s'il
vous plait, saisir 'o' ou 'n' ! ");
                rep = Console.ReadLine();
            }
            Console.WriteLine("");

            while (rep == "o")
            {
                // Chaîne de connexion SQL Server
                SqlConnection maConnection = new
SqlConnection("Persist Security
Info=False;Trusted_Connection=True;database=TP_ADO;server=(local)");

                try
                {
                    // Connexion à la base de données
                    maConnection.Open();
                }
                catch (Exception e)
                {
                    Console.WriteLine(e.ToString());
                }

                try
                {
                    Console.Clear();
```

```
                    Console.WriteLine("");
                    Console.WriteLine("-----------------------------
-------------------------------------------------");
                    Console.WriteLine("|
Module de lecture en BD                          |");
                    Console.WriteLine("-----------------------------
-------------------------------------------------");
                    Console.WriteLine("");
                    SqlDataReader myReader = null;
                    SqlCommand maCommand = new SqlCommand("SELECT *
FROM T_Clients", maConnection);
                    myReader = maCommand.ExecuteReader();
                    Console.WriteLine("Identifiant    -    Nom");
                    Console.WriteLine("_____________________");
                    Console.WriteLine("");
                    while (myReader.Read())
                    {

Console.WriteLine(myReader["ID_client"].ToString() + " - " +
myReader["Nom_client"].ToString());
                    }
                }
                catch (Exception e)
                {
                    Console.WriteLine(e.ToString());
                }

                Console.WriteLine("");
                Console.Write("Voulez-vous re-faire une lecture dans
la base (o ou n) ? ");
                rep = Console.ReadLine();
                while (rep != "o" && rep != "n")
                {
                    Console.Write("Erreur de saisie ! Veuillez, s'il
vous plait, saisir 'o' ou 'n' ! ");
                    rep = Console.ReadLine();
                }
                Console.WriteLine("");

                try
                {
                    // Deconnexion de la base de données
                    maConnection.Close();
                }
                catch (Exception e)
                {
                    Console.WriteLine(e.ToString());
                }
            }

            return reponse;
        }

        static void Main(string[] args)
        {
```

```
            while (reponse == true)
            {
                Console.Clear();
                Console.WriteLine("");
                Console.WriteLine("----------------------------------
------------------------------------------");
                Console.WriteLine("|
ECRITURE / LECTURE EN BD                          |");
                Console.WriteLine("----------------------------------
------------------------------------------");
                Console.WriteLine("");

                Console.Write("Voulez-vous faire une écriture /
lecture en BD (e ou l) ou quitter (q) ? ");
                string rep = Console.ReadLine();
                while (rep != "e" && rep != "l" && rep != "q")
                {
                    Console.Write("Erreur de saisie ! Veuillez, s'il
vous plait, saisir 'e', 'l' ou 'q' ! ");
                    rep = Console.ReadLine();
                }
                Console.WriteLine("");

                if (rep == "e")
                {
                    reponse = saisie();
                }
                else if (rep == "l")
                {
                    reponse = lecture();
                }
                else
                {
                    reponse = false;
                }
            }
        }
    }
}
```