



Support de cours ADO.NET



1.

ADO.NET

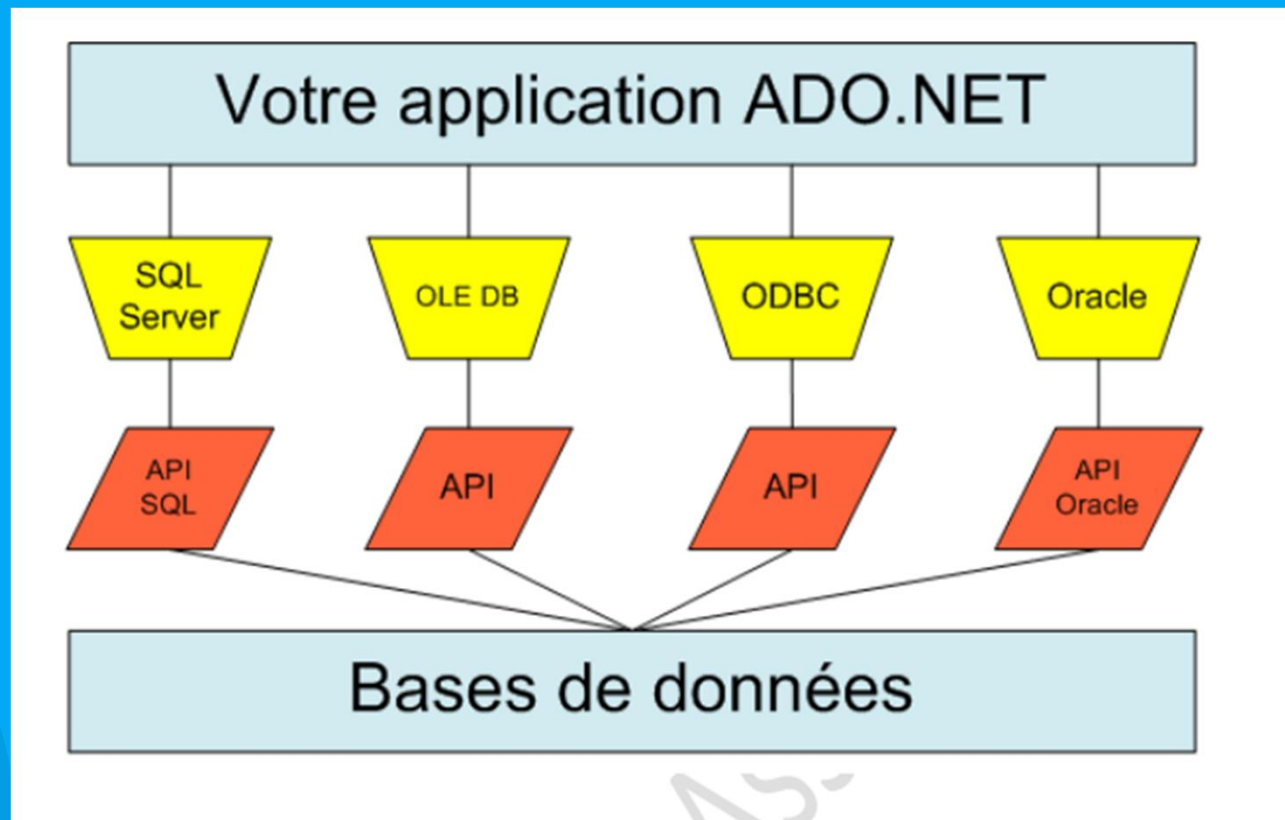


Qu'est-ce qu'
ADO.NET?

Qu'est-ce qu' ADO.NET?

ADO.NET est une technologie permettant d'interagir avec une base de données à partir d'une application .NET.

Présentation
générale
Fonctionnement



Connecteurs disponibles

- **SQL Server** (System.Data.SqlClient)
- **Oracle** (System.Data.OracleClient)
- **OLE DB** : BD relationnelles + Excel + Fichiers plats ... (System.Data.OleDb) *OBSOLETE*
- => **SQL Server Native Client**
- **ODBC** : BD relationnelles (System.Data.Odbc)

DEMO SqlConnectionString

- Connection base de données
- SqlConnection

```
static void Main(string[] args)
{
    using (SqlConnection s = new SqlConnection())
    {
        s.ConnectionString = ConfigurationManager.ConnectionStrings["NorthWindDb"].ToString();
        s.Open();
    }
}
```

SQL Command

Stocke les informations sur la commande et permet son exécution sur le serveur de base de données.

```
SqlCommand command = new SqlCommand("SELECT TOP 100 * FROM Customers", connection);
```


SQL Command : Propriétés principales

CommandText : La requête SQL sous forme d'une string

Connection : Attache la Connection

CommandTimeout : Timeout par défaut 30 secondes

CommandType : Text, TableDirect(OLEDB), StoredProcedure

Parameters : Liste de paramètres nécessaire à l'exécution de la requête.

Transaction : Obtient ou définit l'objet SqlTransaction dans lequel SqlCommand s'exécute.

```
SqlCommand command = new SqlCommand("SELECT TOP 100 * FROM Customers", connection);
```

```
SqlDataReader reader = command.ExecuteReader();
```

SQL Command Méthodes

ExecuteNonQuery : Utilisé pour des requêtes qui ne renvoient aucun résultats (INSERT,DELETE,UPDATE)

ExecuteReader : Utilisé pour des requetes qui envoient plusieurs lignes (SELECT TOP 100 ...)

ExecuteScalar : Utilisé pour des requetes qui envoient une nombre (SELECT COUNT(*) ...)

Dispose : Libère toutes les ressources utilisées

DEMO SQL COMMAND

- ▶ ExecuteQuery
- ▶ ExecuteScalar
- ▶ ExecuteNonQuery

```
"SELECT TOP 100 * FROM Customers WHERE City = 'toto' OR 1=1"
```

Passer des parametres

Select * from Customers where City = "Berlin"

```
//A éviter - ATTAQUE PAR INJECTION SQL  
var queryText = "SELECT TOP 100 * FROM Customers WHERE City =" + city;
```

```
SqlCommand command = new SqlCommand(queryText, connection);  
command.Parameters.AddWithValue("@city", city);
```

```
SqlCommand command = new SqlCommand(queryText, connection);  
command.Parameters.Add("@city", SqlDbType.VarChar);  
command.Parameters["@city"].Value = city;
```

ExecuteReader : Récupération de champs spécifiques

2 manières de faire

```
SqlDataReader reader = command.ExecuteReader();
var c = new Customer();
while (reader.Read())
{
    c.CustomerId = reader.GetInt32(reader.GetOrdinal("CustomerId"));
    c.ContactName = reader["ContactName"].ToString();
    c.CompanyName = reader.GetString(reader.GetOrdinal("CompanyName"));
}
reader.Close();
return c;
```

Procédure Stockée

Il est possible d'appeler une procédure stockée depuis notre code

```
using (SqlConnection connection = new SqlConnection())
{
    connection.ConnectionString = Db.ConnectionString;
    connection.Open();

    SqlCommand command = new SqlCommand("SalesByCategory", connection);
    command.CommandType = System.Data.CommandType.StoredProcedure;
    command.Parameters.AddWithValue("@CategoryName", "Beverages");

    SqlDataReader reader = command.ExecuteReader();
    while (reader.Read())
    {
        System.Diagnostics.Debug.WriteLine("\t{0}\t{1}",
            reader[0], reader[1]);
    }
    reader.Close();
}
```

A vous de travailler !

- Configurer la connection à NorthWindDb
- Créer une méthode qui récupère les 20 premiers Customers vivant en France et renvoie une liste
- Récupérer le résultat de la procédure CustOrderHist pour le customerId BLONP
- Insérer une nouvelle ligne dans la table Customers
- Modifier le pays de cette meme ligne

Transaction

Une transaction est une unité de travail.

Lorsqu'une transaction aboutit, toutes les modifications de données apportées lors de la transaction sont validées et intégrées de façon permanente à la base de données.

Si une transaction rencontre des erreurs et doit être annulée ou restaurée, toutes les modifications de données sont supprimées.

Transaction



```
SqlTransaction trans;

var sqlCommandList = new List<string>();
sqlCommandList.Add("INSERT INTO dbo.Customers(CustomerID, CompanyName, ContactName, PostalCode, Country, Phone, Fax) VALUES('ALFKI2', 'Alfreds Futterkiste2', 'Maria Str. 57', 'Berlin', NULL, '12209', 'Germany', '030-0074321', '030-0076545')");

sqlCommandList.Add("INSERT INTO dbo.Customers(CustomerID, CompanyName, ContactName, PostalCode, Country, Phone, Fax) VALUES('ALFKI3', 'Alfreds Futterkiste3', 'Maria Str. 57', 'Berlin', NULL, '12209', 'Germany', '030-0074321', '030-0076545')");

SqlConnection connection = new SqlConnection(Db.ConnectionString);
trans = connection.BeginTransaction();
try
{
    connection.Open();
    foreach (var commandString in sqlCommandList)
    {
        SqlCommand command = new SqlCommand(commandString, connection, trans);
        command.ExecuteNonQuery();
    }

    trans.Commit();
}
catch (Exception ex) //error occurred
{
    trans.Rollback();
    //Handle error
}
```

Conclusion

ADO.NET permet d'interagir avec une base de données à partir de code .NET

Même si la technologie est ancienne vous êtes quasiment certain de la trouver en entreprise dans presque tous vos projets.

ADO.NET et DataSet

DataSet

Un DataSet est une structure de données, composée de DataTables sur laquelle il est possible de faire des modifications/lecture de manière déconnectée.

Avantages


- Mode Déconnecté : Moins de connection à la base de données donc (en théorie) moins de stress sur la base.

Inconvénients

- Concurrence : Si plusieurs utilisateurs modifient la meme donnée.
- La gourmandise est un vilain défaut (mémoire + temps d'exécution)

→ TRES PEU UTILISE ACTUELLEMENT (Remplacé par L'ORM EntityFramework)

Comment travailler avec un DATASET ?



```
using (SqlConnection connection = new SqlConnection())
{
    connection.ConnectionString = Db.ConnectionString;
    connection.Open();

    var queryText = "SELECT TOP 100* FROM Customers";

    DataSet dataSet = new DataSet();
    SqlCommand command = new SqlCommand(queryText, connection);

    SqlDataAdapter adapter = new SqlDataAdapter();
    adapter.SelectCommand = command;

    adapter.Fill(dataSet);

    for (int i = 0; i <= dataSet.Tables[0].Rows.Count - 1; i++)
    {
        System.Diagnostics.Debug.WriteLine(dataSet.Tables[0].Rows[i].ItemArray[0] + " -- " + dataSet.Tables[0].Rows[i].ItemArray[1]);
    }
}
```

DataSet Visualizer

Table:

Table

| | CustomerID | CompanyName | ContactName | Contact Title | Address | City | Regi |
|--|------------|----------------------|--------------------|---------------------|---------------------|-------------|------|
| | ALFKI | Alfreds Futterkiste | Maria Anders | Sales Represent... | Obere Str. 57 | Berlin | |
| | ANATR | Ana Trujillo Empa... | Ana Trujillo | Owner | Avda. de la Cons... | México D.F. | |
| | ANTON | Antonio Moreno ... | Antonio Moreno | Owner | Mataderos 2312 | México D.F. | |
| | AROUT | Around the Horn | Thomas Hardy | Sales Represent... | 120 Hanover Sq. | London | |
| | BERGS | Berglunds snabb... | Christina Berglund | Order Administrator | Berguvsvägen 8 | Luleå | |
| | BLAUS | Blauer See Delik... | Hanna Moos | Sales Represent... | Forsterstr. 57 | Mannheim | |
| | BLONP | Blondesddsl père... | Frédérique Citeaux | Marketing Manager | 24, place Kléber | Strasbourg | |
| | BOLID | Bólido Comidas p... | Martín Sommer | Owner | C/ Araquil, 67 | Madrid | |
| | BONAP | Bon app' | Laurence Lebihan | Owner | 12, rue des Bouc... | Marseille | |

Close

Exercice

Créer une classe Customer avec les colonnes en base de données

Créer une classe CustomerRepository
Dans cette class implémenter les méthodes suivantes :

- GetCustomerById
- GetCustomersByCountry
- UpdateCustomerCity
- DeleteCustomerByName