Dotnet France
Technologies Sharepoint, SQL Server & .NET

Association Dotnet France

ADO.NET – LINQ

LINQ to DataSet

Version 1.0

James RAVAILLE
http://blogs.dotnet-france.com/jamesr

Matthieu LAFFONT

Dotnet France
Technologies Sharepoint, SQL Server & .NET

Sommaire

1 Introduction 3
1.1 Présentation des DataSets 3
1.2 Présentation de Linq To DataSet 3
1.3 Pré-requis 3
2 Présentation de la source de données 4
3 Ecrire des requêtes LINQ sur un DataSet non typé 6
3.1 Création du DataSet 6
3.2 Exécuter une requête LINQ 8
3.3 Afficher la liste des stagiaires 9
3.4 Afficher la liste des stagiaires et de leurs cours 9
3.5 Accéder aux données résultant de l'exécution de requêtes LINQ 11
4 Ecrire des requêtes LINQ sur un DataSet typé 12
4.1 Conception du DataSet typé 12
4.2 Création du DataSet typé 13
4.3 Ecriture de requêtes LINQ 14
4.3.1 Liste des stagiaires 14
4.3.2 Liste des stagiaires et de leurs cours 14
5 Conclusion 16

1 Introduction

1.1 Présentation des DataSets

Un *DataSet* est un groupe de données organisées de manière relationnelle. Un *DataSet* ressemble à une base de données relationnelle. Il est constitué de *DataTable*, reliées entre elles par des *DataRelation*. Les *DataTable* sont structurées par des champs (*DataColumn*), et contiennent des enregistrements (*DataRow*), eux-mêmes contenant les données.

1.2 Présentation de Linq To DataSet

LINQ To DataSet, comme son nom le suggère, permet de requêter sur un *DataSet*. Toutes les requêtes sur des données du *DataSet* ne demanderont aucune future connexion à la source de données, ayant permis de l'alimenter.

Quels sont les cas d'utilisation de LINQ To DataSet ? Dans une application .NET, vous pouvez être amené à travailler avec une DataSet, que vous avez obtenu :

- En le créant dynamiquement (« à la main ») lors de l'exécution de l'application.
- Via l'exécution d'un service de données (par exemple un service Web, …).
- En exécutant une requête SQL de sélection tabulaire (requête SQL de sélection retournant un tableau de données) sur une base de données.
- …

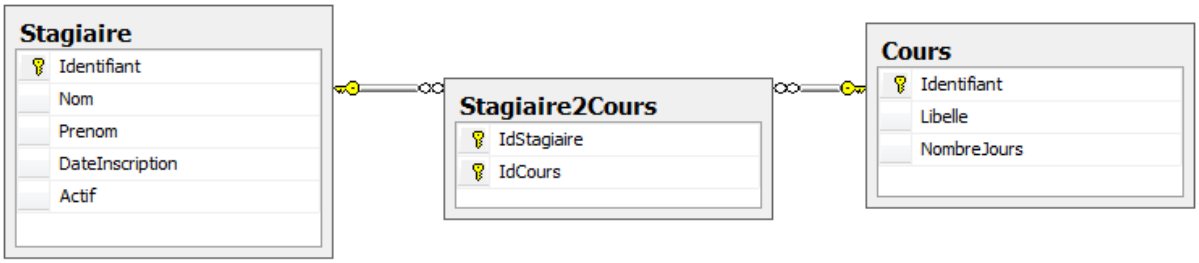
1.3 Pré-requis

Avant d'aborder ce cours, nous vous conseillons de lire les chapitres *« Introduction au LINQ »*, *« LINQ to Objects »*, publiés sur Dotnet-France.

Aussi, même si nous l'abordons sommairement dans ce cours, si vous avez besoin de plus amples informations sur la création d'un DataSet, nous vous conseillons de lire le cours « *Utilisation du DataSet en mode deconnecté* ».

2 Présentation de la source de données

Voici une présentation du schéma de la base de données SQL Server, que nous allons utiliser pour alimenter nos *DataSet* :



Voici le script de base de données, permettant de créer ce schéma de base de données :

```
// SQL

CREATE TABLE [dbo].[Stagiaire](
      [Identifiant] [int] IDENTITY(1,1) NOT NULL,
      [Nom] [varchar](50) NOT NULL,
      [Prenom] [varchar](50) NOT NULL,
      [DateInscription] [datetime] NULL,
      [Actif] [bit] NULL,
 CONSTRAINT [PK_Stagiaire] PRIMARY KEY CLUSTERED
(
      [Identifiant] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY],
 CONSTRAINT [UQ__Stagiaire__7F60ED59] UNIQUE NONCLUSTERED
(
      [Identifiant] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

CREATE TABLE [dbo].[Cours](
      [Identifiant] [int] IDENTITY(1,1) NOT NULL,
      [Libelle] [varchar](50) NOT NULL,
      [NombreJours] [int] NOT NULL,
 CONSTRAINT [PK_Cours] PRIMARY KEY CLUSTERED
(
      [Identifiant] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY],
 CONSTRAINT [UQ__Cours__023D5A04] UNIQUE NONCLUSTERED
(
      [Identifiant] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

CREATE TABLE [dbo].[Stagiaire2Cours](
      [IdStagiaire] [int] NOT NULL,
      [IdCours] [int] NOT NULL,
 CONSTRAINT [PK_Stagiaire2Cours] PRIMARY KEY CLUSTERED
(
      [IdStagiaire] ASC,
      [IdCours] ASC
)WITH (IGNORE_DUP_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

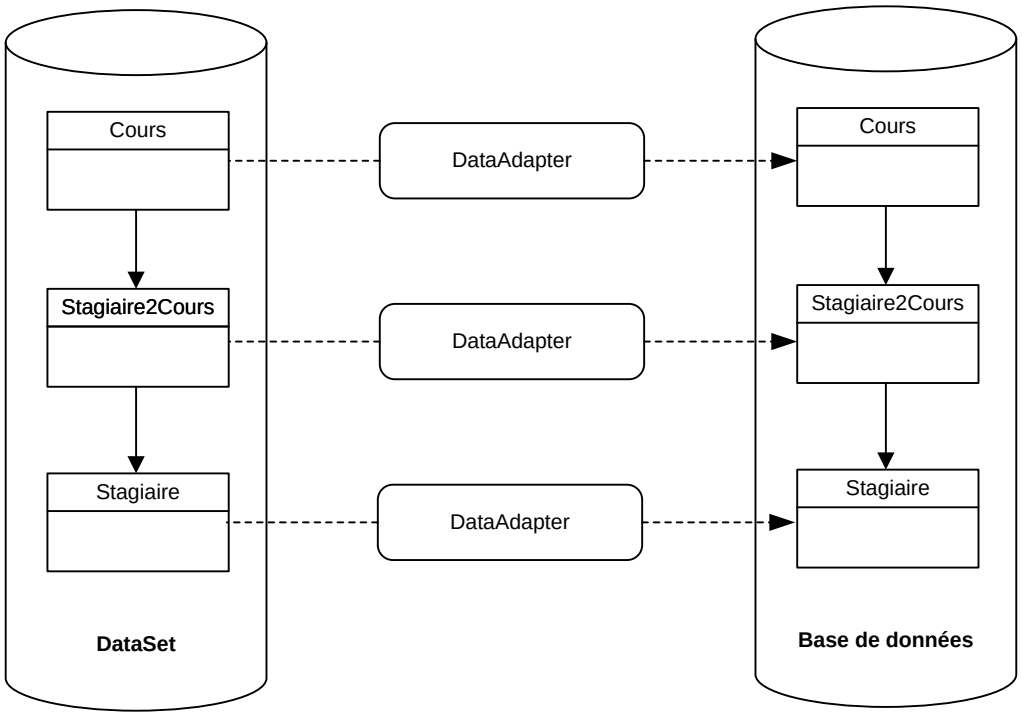
```
ALTER TABLE [dbo].[Stagiaire2Cours]  WITH CHECK ADD  CONSTRAINT
[FK_Stagiaire2Cours_Cours] FOREIGN KEY([IdCours])
REFERENCES [dbo].[Cours] ([Identifiant])
ON UPDATE CASCADE
ON DELETE CASCADE

ALTER TABLE [dbo].[Stagiaire2Cours]  WITH CHECK ADD  CONSTRAINT
[FK_Stagiaire2Cours_Stagiaire] FOREIGN KEY([IdStagiaire])
REFERENCES [dbo].[Stagiaire] ([Identifiant])
ON UPDATE CASCADE
ON DELETE CASCADE
```

3 Ecrire des requêtes LINQ sur un DataSet non typé

3.1 Création du DataSet

Dans les blocs de code ci-dessous, nous allons créer un *DataSet*. Ce *DataSet* aura la même structure que notre base de données : il contiendra trois tables, telles qu'elles sont indiquées dans le schéma ci-dessous. Pour créer et alimenter ces trois tables au sein de notre *DataSet*, nous créerons trois *DataAdapter* :



```
// C#

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

// ...
// ...

// Variables locales.
DataSet oDataSet;
string sChaineConnexion;
SqlDataAdapter oDaStagiaire, oDaStagiaireToCours, oDaCours;

try
{
    // Initialisation.
    sChaineConnexion = @"Data source=localhost\sql2005; Initial
Catalog=DotnetFrance_ExoRequetes; Integrated Security=True";

    // Création du DataSet.
    oDataSet = new DataSet("DotnetFrance");

    // Création et alimentation de la table Stagiaire dans le DataSet.
    oDaStagiaire = new SqlDataAdapter("SELECT * FROM Stagiaire",
sChaineConnexion);
    oDaStagiaire.Fill(oDataSet, "Stagiaire");

    // Création et alimentation de la table Cours dans le DataSet.
    oDaCours = new SqlDataAdapter("SELECT * FROM Cours",
sChaineConnexion);
    oDaCours.Fill(oDataSet, "Cours");

    // Création et alimentation de la table Stagiaire2Cours dans le
DataSet.
    oDaStagiaireToCours = new SqlDataAdapter("SELECT * FROM
Stagiaire2Cours", sChaineConnexion);
    oDaStagiaireToCours.Fill(oDataSet, "Stagiaire2Cours");

    // Création des relations entre les tables dans le DataSet.
    oDataSet.Relations.Add("Stagiaire_Stagiaire2Cours",
         oDataSet.Tables["Stagiaire"].Columns["Identifiant"],
         oDataSet.Tables["Stagiaire2Cours"].Columns["IdStagiaire"]);
    oDataSet.Relations.Add("Stagiaire2Cours_Stagiaire",
         oDataSet.Tables["Stagiaire2Cours"].Columns["IdCours"],
         oDataSet.Tables["Cours"].Columns["Identifiant"]);
}
catch (Exception aEx)
{
    MessageBox.Show(aEx.Message);
}
```

```
' VB.NET

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

' ...
' ...

' Variables locales.
Dim oDataSet As DataSet
Dim sChaineConnexion As String
Dim oDaStagiaire As SqlDataAdapter
Dim oDaStagiaireToCours As SqlDataAdapter
Dim oDaCours As SqlDataAdapter

Try
    ' Initialisation.
    sChaineConnexion = "Data source=localhost\sql2005; Initial
Catalog=DotnetFrance_ExoRequetes; Integrated Security=True"

    ' Création du DataSet.
    oDataSet = New DataSet("DotnetFrance")

    ' Création et alimentation de la table Stagiaire dans le DataSet.
    oDaStagiaire = New SqlDataAdapter("SELECT * FROM Stagiaire",
sChaineConnexion)
    oDaStagiaire.Fill(oDataSet, "Stagiaire")

    ' Création et alimentation de la table Cours dans le DataSet.
    oDaCours = New SqlDataAdapter("SELECT * FROM Cours",
sChaineConnexion)
    oDaCours.Fill(oDataSet, "Cours")

    ' Création et alimentation de la table Stagiaire2Cours dans le
DataSet.
    oDaStagiaireToCours = New SqlDataAdapter("SELECT * FROM
Stagiaire2Cours", sChaineConnexion)
    oDaStagiaireToCours.Fill(oDataSet, "Stagiaire2Cours")

    ' Création des relations entre les tables dans le DataSet.
    oDataSet.Relations.Add("Stagiaire_Stagiaire2Cours", _
        oDataSet.Tables("Stagiaire").Columns("Identifiant"), _
        oDataSet.Tables("Stagiaire2Cours").Columns("IdStagiaire"))
    oDataSet.Relations.Add("Stagiaire2Cours_Stagiaire", _
        oDataSet.Tables("Stagiaire2Cours").Columns("IdCours"), _
        oDataSet.Tables("Cours").Columns("Identifiant"))
Catch aEx As Exception
    MessageBox.Show(aEx.Message)
End Try
```

3.2 Exécuter une requête LINQ

L'exécution de requêtes sur une *DataTable* est relativement similaire à l'établissement de requête un objet contenant un ensemble de données, dont les classes ayant permis de les créer implémente l'interface *IEnumerable* du Framework .NET. Cependant, la classe *DataTable* n'implémente pas cette interface. C'est pourquoi, il est nécessaire de lui appliquer la méthode *AsEnumerable()* pour requêter l'ensemble des *DataRow* composant les *DataTable*, et la méthode *Field<T>() / Field (Of T)* sur les *DataRow* (lignes) des DataTable pour accéder aux *DataColumn* (champs).

3.3 Afficher la liste des stagiaires

Voici un bloc d'instructions, permettant d'obtenir la liste des stagiaires via une requête LINQ, triés suivant le nom et le prénom dans l'ordre alphabétique, et de l'afficher dans une grille de données via une opération de DataBinding :

```
// C#

DataView oListeStagiaires =
    (from oStagiaire in oDataSet.Tables["Stagiaire"].AsEnumerable()
     orderby oStagiaire.Field<string>("Nom"),
             oStagiaire.Field<string>("Prenom")
     select oStagiaire).AsDataView();

LstStagiaires.DataSource = oListeStagiaires;
```

```
' VB .NET

Dim oListeStagiaires As DataView = _
    (From oStagiaire In oDataSet.Tables("Stagiaire").AsEnumerable() _
     Order By oStagiaire.Field(Of String)("Nom"), _
             oStagiaire.Field(Of String)("Prenom") _
     Select oStagiaire).AsDataView()

LstStagiaires.DataSource = oListeStagiaires
```

On obtient le résultat suivant :

Identifiant	Nom	Prenom	DateInscription	Actif
1	DEROUX	Alain	01/09/2008	☑
4	EMATO	Julie	07/07/2008	☐
2	RAVAILLE	James	10/12/2007	☐
3	SIRON	Karl	23/02/2007	☑

3.4 Afficher la liste des stagiaires et de leurs cours

Voici un bloc d'instructions, permettant d'obtenir au travers d'une requête LINQ, la liste des stagiaires avec pour chacun d'entre eux, la liste des cours auxquels ils sont inscrits, et de l'afficher dans une grille de données via une opération de DataBinding :

```
// C#

var oListeStagiairesAvecCours =
    (from oStagiaire in oDataSet.Tables["Stagiaire"].AsEnumerable()
     from oStagiaire2Cours in
oStagiaire.GetChildRows("Stagiaire_Stagiaire2Cours")
     from oCours in
oStagiaire2Cours.GetChildRows("Stagiaire2Cours_Stagiaire")
     select new
     {
         NomPrenom = oStagiaire.Field<string>("Nom") + " " +
oStagiaire.Field<string>("Prenom"),
         Cours = oCours.Field<string>("Libelle"),
         Duree = oCours.Field<int>("NombreJours")
     }).ToList();

LstCoursStagiaires.DataSource = oListeStagiairesAvecCours;
```

```
' VB .NET

Dim oListeStagiairesAvecCours = _
    (From oStagiaire In oDataSet.Tables("Stagiaire").AsEnumerable() _
     From oStagiaire2Cours In
oStagiaire.GetChildRows("Stagiaire_Stagiaire2Cours") _
     From oCours In
oStagiaire2Cours.GetChildRows("Stagiaire2Cours_Stagiaire") _
     Select New With { _
         .NomPrenom = oStagiaire.Field(Of String)("Nom") + " " +
oStagiaire.Field(Of String)("Prenom"), _
         .Cours = oCours.Field(Of String)("Libelle"), _
         .Duree = oCours.Field(Of Integer)("NombreJours") _
     }).ToList()

LstCoursStagiaires.DataSource = oListeStagiairesAvecCours
```

Vous remarquerez :

- L'utilisation d'un type anonyme, permettant de réaliser une projection de données, afin d'obtenir uniquement le nom et prénom du stagiaire agrégé au sein d'une même propriété, le libellé et la durée des cours auxquels ils sont inscrits.
- L'utilisation de la méthode *GetChildRows*, permettant d'obtenir un tableau de *DataRow* dépendant, au travers d'une relation que nous avons précédemment définie. En l'occurrence, elle est utilisée pour obtenir la liste des cours des stagiaires.

Voici le résultat d'exécution :

NomPrenom	Cours	Duree
DEROUX Alain	SQL Server - Administration de serveurs	5
DEROUX Alain	C#	5
RAVAILLE James	SQL Server - Administration de serveurs	5
RAVAILLE James	XHTML / CSS	3
RAVAILLE James	C#	5
RAVAILLE James	ASP .NET 3.5	5

3.5 Accéder aux données résultant de l'exécution de requêtes LINQ

Sur un *DataSet*, il est aussi possible d'accéder aux données sans utiliser de requête, via l'instruction itérative *foreach* / *For Each*. Là aussi, nous retrouvons l'obligation d'utiliser les méthodes suivantes :

- *Field<T>()* / *Field(Of T)* pour lire les données dans les champs des enregistrements.
- *SetField<T>()* / *SetField(Of T)* pour modifier les données dans les champs des enregistrements.

Les blocs de code ci-dessous illustre leur utilisation, en remplissant une liste des données de type *ListBox* :

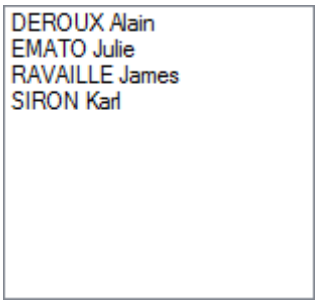
```
// C#

LstNomsPrenoms.Items.Clear();
foreach (DataRowView oStagiaire in oListeStagiaires)
{
    LstNomsPrenoms.Items.Add(oStagiaire.Row.Field<string>("Nom") + " " +
                             oStagiaire.Row.Field<string>("Prenom"));
}
```

```
' VB.NET

LstNomsPrenoms.Items.Clear()
For Each oStagiaire As DataRowView In oListeStagiaires
    LstNomsPrenoms.Items.Add(oStagiaire.Row.Field(Of String)("Nom") + " "
+ oStagiaire.Row.Field(Of String)("Prenom"))
Next
```

Le résultat d'exécution de ce bloc d'instructions est le suivant :

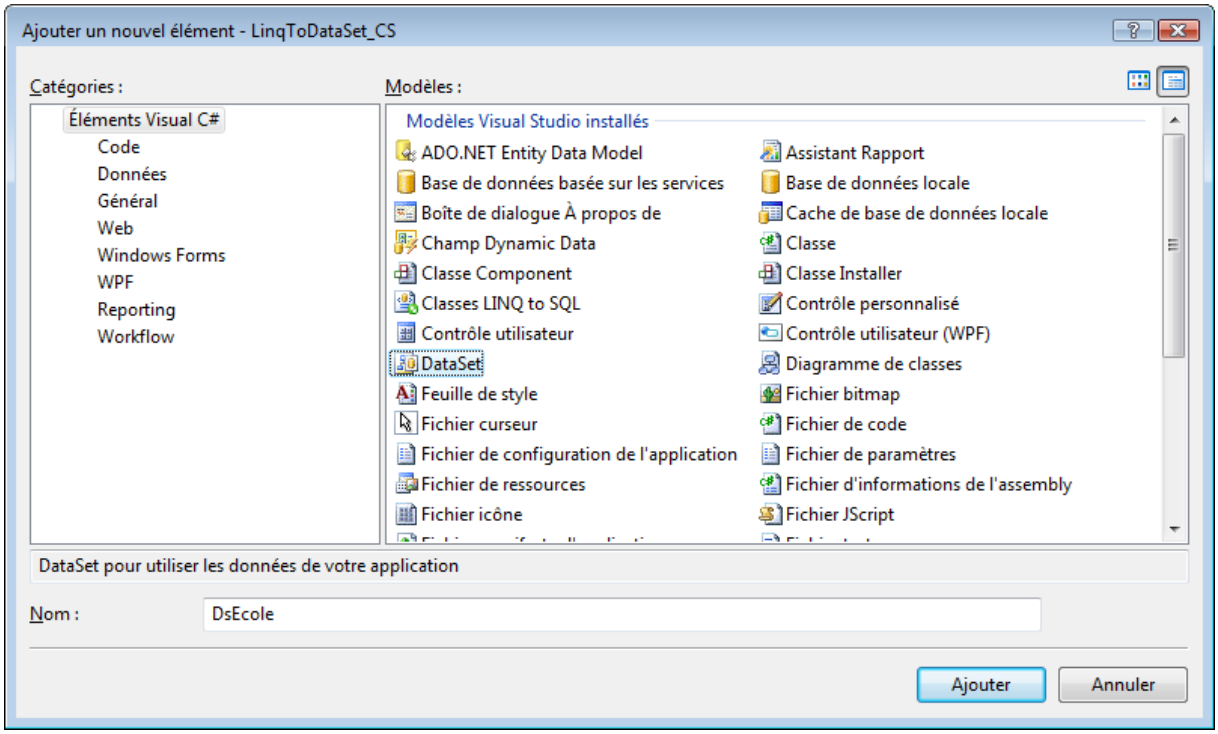


4 Ecrire des requêtes LINQ sur un DataSet typé

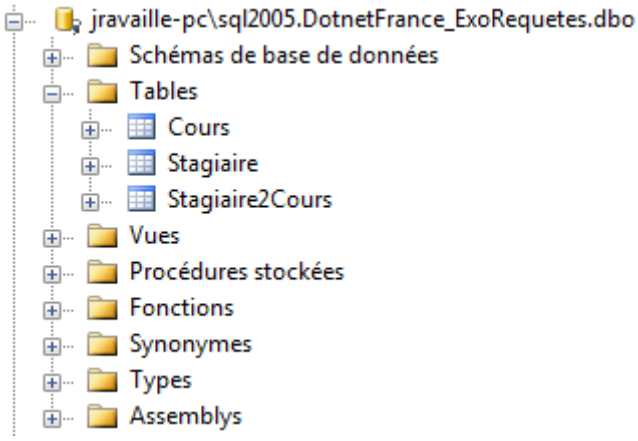
L'utilisation d'un *DataSet* typé, permet d'écrire des requêtes LINQ plus simples.

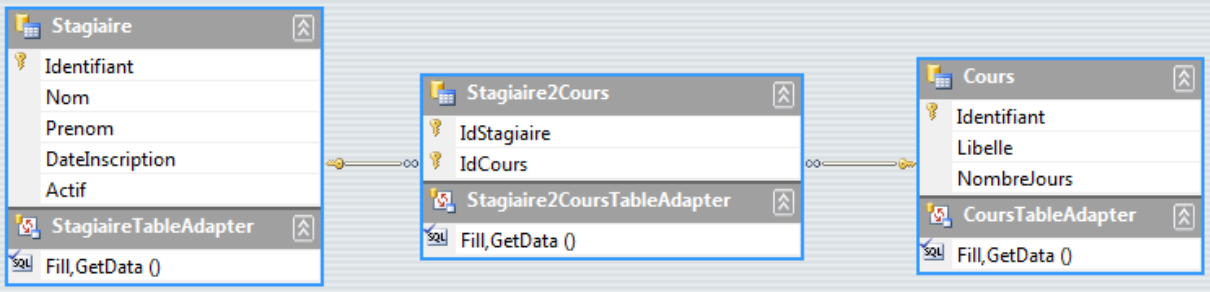
4.1 Conception du DataSet typé

Pour créer un *DataSet* typé dans Visual Studio, ajouter un *« Nouvel élément »* au projet de type *« DataSet »*. Nommez le, puis validez en cliquant sur le bouton *Ajouter* :



Puis, dans la fenêtre *Explorateur de serveurs*, créez une connexion vers votre base de données. Après avoir accédé aux tables, faire glisser les tables voulues vers le *DataSet*.





Nous pouvons observer la création :

- De *DataTable*, ayant la même structure que les tables de la base de données.
- De *TableApdater*, qui correspondent à des *DataAdapter* spécialisés par *DataTable*, permettant d'alimenter une table, et répercuter les modifications des données dans la base de données.
- De relations entre les *DataTable*, issues des relations entre les tables de la base de données.

4.2 Création du DataSet typé

Voici les blocs d'instructions permettant de créer un *DataSet* typé :

```
// C#

using System.Data.SqlClient;
using LinqToDataSet_CS;
using LinqToDataSet_CS.DsEcoleTableAdapters;

// ...
// ...

DsEcole oDsEcole;
StagiaireTableAdapter oStagiaireTableAdapter;
CoursTableAdapter oCoursTableAdapter;
Stagiaire2CoursTableAdapter oStagiaire2CoursTableAdapter;

oDsEcole = new DsEcole();
oStagiaireTableAdapter = new StagiaireTableAdapter();
oStagiaireTableAdapter.Fill(oDsEcole.Stagiaire);

oCoursTableAdapter = new CoursTableAdapter();
oCoursTableAdapter.Fill(oDsEcole.Cours);

oStagiaire2CoursTableAdapter = new Stagiaire2CoursTableAdapter();
oStagiaire2CoursTableAdapter.Fill(oDsEcole.Stagiaire2Cours);
```

```
' VB .NET

Imports System.Data.SqlClient
Imports LinqToDataSet_VB.DsEcoleTableAdapters
Imports LinqToDataSet_VB.DsEcole

' ...
' ...

Dim oDsEcole As DsEcole
Dim oStagiaireTableAdapter As StagiaireTableAdapter
Dim oCoursTableAdapter As CoursTableAdapter
Dim oStagiaire2CoursTableAdapter As Stagiaire2CoursTableAdapter

oDsEcole = New DsEcole()
oStagiaireTableAdapter = New StagiaireTableAdapter()
oStagiaireTableAdapter.Fill(oDsEcole.Stagiaire)

oCoursTableAdapter = New CoursTableAdapter()
oCoursTableAdapter.Fill(oDsEcole.Cours)

oStagiaire2CoursTableAdapter = New Stagiaire2CoursTableAdapter()
oStagiaire2CoursTableAdapter.Fill(oDsEcole.Stagiaire2Cours)
```

4.3 Ecriture de requêtes LINQ

4.3.1 Liste des stagiaires

Voici un bloc d'instructions, permettant d'obtenir au travers d'une requête LINQ, la liste des stagiaires :

```
// C#

List<DsEcole.StagiaireRow> oListeStagiaires =
    (from oStagiaire in oDsEcole.Stagiaire
     orderby oStagiaire.Nom,
             oStagiaire.Prenom
     select oStagiaire).ToList();
```

```
' VB .NET

Dim oListeStagiaires As List(Of StagiaireRow) = _
    (From oStagiaire In oDsEcole.Stagiaire _
     Order By oStagiaire.Nom, _
             oStagiaire.Prenom _
     Select oStagiaire).ToList()
```

4.3.2 Liste des stagiaires et de leurs cours

Voici un bloc d'instructions, permettant d'obtenir au travers d'une requête LINQ, la liste des stagiaires avec pour chacun d'entre eux, la liste des cours auxquels ils sont inscrits :

```
// C#

var oListeStagiairesAvecCours =
    (from oStagiaire in oDsEcole.Stagiaire
     from oStagiaire2Cours in oStagiaire.GetStagiaire2CoursRows()
     select new
     {
         NomPrenom = oStagiaire.Nom + " " + oStagiaire.Prenom,
         Cours = oStagiaire2Cours.CoursRow.Libelle,
         Duree = oStagiaire2Cours.CoursRow.NombreJours
     }).ToList();
```

```
' VB .NET

Dim oListeStagiairesAvecCours = _
    (From oStagiaire In oDsEcole.Stagiaire _
     From oStagiaire2Cours In oStagiaire.GetStagiaire2CoursRows() _
     Select New With { _
         .NomPrenom = oStagiaire.Nom + " " + oStagiaire.Prenom, _
         .Cours = oStagiaire2Cours.CoursRow.Libelle, _
         .Duree = oStagiaire2Cours.CoursRow.NombreJours _
     }).ToList()
```

5 Conclusion

Dans ce cours, nous avons abordé l'écriture de requêtes LINQ sur des *DataSet* typés et non typés, ainsi que des *DataTable*. Si vous souhaitez en apprendre davantage sur la manipulation de données avec LINQ, les chapitres *« Introduction au LINQ »*, *« LINQ to Objects »* et *« LINQ to SQL »* sont à votre disposition le site Dotnet-France.

Pour de plus amples informations sur Linq To DataSet, vous pouvez consulter le site MSDN à l'URL suivante : http://msdn.microsoft.com/fr-fr/library/bb386977.aspx