

WPF – MVVM – Entity Framework : Exercice

Objectifs

Développer une petite application professionnelle intégrant les notions d'Entity Framework, WPF, MVVM vues

Aperçu de l'application à développer

First Name	Birth Date
Andrew Fuller	19/02/1952
Janet Leverling	30/08/1963
Margaret Peacock	19/09/1937
Steven Buchanan	4/03/1955
Michael Suyama	2/07/1963
Rob King	29/05/1960
Laura Callahan	9/01/1958
Anne Dodsworth	27/01/1966

OrderID	OrderDate	OrderTotal
11074	6/05/1998	17,4500
11066	1/05/1998	40,6500
11055	28/04/1998	91,0000

Exercice

1. Vérifiez que la base de données «Northwind» est bien installée sur votre machine.
 1. Cfr. Cours sur Linq & Entity Framework
2. Récupérez sur Moodle le projet de base (WpfEmployee)
3. Première étape → lien avec la base de données
 1. Générez le modèle à partir de la base de données Northwind
4. Deuxième étape → Remarquez que le binding dans la vue a déjà été fait par le designer
 1. Vous devrez respecter ceci
5. Troisième étape → affichage des « Employees » dans le datagrid

1. Pensez à respecter MVVM
2. Créez une méthode LoadEmployees()
3. Pensez à créer une classe EmployeeModel qui encapsulera un Employee(généré par LINQ to Entities)
4. Votre liste d'« Employee » contiendra des « EmployeeModel »
5. Le datagrid affiche le fullName (concaténation du nom et prénom) et le birthdate
 1. Que faire pour que cela s'affiche ? Où placer les propriétés de binding ? Que contient votre « EmployeeList » ?
6. Pensez également au DataContext
 1. Comment la vue est-elle liée au viewModel ?
6. Quatrième étape → affichage dans le formulaire lors d'une sélection dans le datagrid
 1. Ne vous préoccupez pas trop de la combo-box pour l'instant
7. Cinquième étape → affichage des différents titres de courtoisie dans la combo-box
 1. Rappelez-vous que le datacontext est hérité → la combo-box a donc un datacontext par défaut lié au grid « form ». Celui-ci contient un « employeeModel ». Ceci ne nous arrange pas car nous voulons tous les titres de courtoisies possibles et non pas celui d'un employé en particulier. C'est pourquoi le designer a précisé dans le XAML d'utiliser le datacontext de la fenêtre et nous utiliserons une propriété ListTitle dans ce datacontext.
 2. Vous en savez maintenant normalement assez → faites en sorte d'afficher les différents titres de courtoisie possibles dans la combo-box.
 3. Inspirez-vous également de LoadEmployees()
8. Sixième étape → Répercussion des modifications dans le formulaire dans le datagrid
 1. But : la modification du lastname est repercutée dans le champ fullname du datagrid
 2. La vue devra être informée qu'elle doit mettre à jour le datagrid. Où le faire et comment le faire ?
9. Septième étape → Les boutons
 1. Remarquez l'utilisation la propriété Command dans le fichier XAML pour le Add et Save
 2. Utiliser le mécanisme des Command pour implémenter les boutons Add et Save
 1. Réfléchissez bien à ce que va faire le bouton Add et le bouton Save
10. Huitième étape → Afficher les 3 dernières commandes de l'employé sélectionné
 1. Ajoutez un datagrid à côté du formulaire qui affichera les 3 dernières commandes de l'employé sélectionné.
 2. Afficher dans un premier temps uniquement le champ OrderId dans le datagrid
 3. Quel datacontext vais-je utiliser ?
 4. Erreur : **System.InvalidOperationException** : 'There is already an open DataReader associated with this Connection which must be closed first.'
 1. Activer le MultipleActiveResultSets dans votre NorthwindContext !
 2. `optionsBuilder.UseSqlServer("Data Source=(localdb)\\MSSQLLocalDB;Initial Catalog=Northwind;MultipleActiveResultSets=true");`

11. Neuvième étape → Afficher la date de commande et son total

1. Pour les 3 dernières commandes, on affichera dans le datagrid le numéro de commande (OrderId), la date de la commande (Order_Date) ainsi que son montant total (celui devra être calculé à partir de Order_Details)

Vous allez être tenter de tout faire en un query LINQ mais celui-ci sera très difficile à écrire et surtout à **relire**. Faites plutôt 2 queries !