Atelier 5 – Données : LinQ, Entity Framework

Objectifs

Au terme de l'atelier, vous saurez :

- Construire un modèle d'après une source de données
- Implanter des règles métiers dans le modèle
- Effectuer des requêtes LinQ sur une source de données

Vue d'ensemble

Via un mapping Entity Framework et LinQ to Entities, vous créez les filtres qui permettent d'exploiter les données d'une base de données SqlServer Compact.

Etapes

Etape 1 – Ossature du projet WPF (30')

Créer un projet application WPF, une Vue et un VueModèle à partir des codes ci-contre, puis intégrer la vue dans la fenêtre principale.

```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="auto" />
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>
    <ComboBox ItemsSource="{Binding Filters}"
              SelectedIndex="{Binding SelectedFilter}"/>
    <DataGrid ItemsSource="{Binding FilteredList}"
              IsReadOnly="True" Grid.Row="1"/>
</Grid>
```

```
public class FilteredListViewModel : INotifyPropertyChanged
{
    private int _selectedFilter=0;
    private readonly string[] _filters;
    private Model.EntrepriseEntities _context;

    public FilteredListViewModel(Model.EntrepriseEntities context)
    {
       _context = context;
       _filters = "Tout le staff,10$ et moins".Split(',');
    }
    public IEnumerable<object> FilteredList
    {
        get {
            switch(this._selectedFilter)
            {
                case 0:
                    return from employee in _context.Employees
                           select new {
                               Prénom = employee.First_Name,
                               Nom = employee.Last_Name
                           };
                case 1:
                    return from product in _context.Products
                           where product.Unit_Price<10.0m
                           select new {
                               Produit = product.Product_Name,
                               Prix = product.Unit_Price
                           };
                default:
                    return new string[] {
                        "(Not implemented filter)"
                    };
            }
        }
    }
    public IEnumerable<String> Filters
    {
        get { return _filters; }
    }
    public int SelectedFilter
    {
        get { return this._selectedFilter; }
        set {
            this._selectedFilter = value;
            if (PropertyChanged != null)
                PropertyChanged(this,
                    new PropertyChangedEventArgs("FilteredList")
            );
        }
    }
    public event PropertyChangedEventHandler PropertyChanged;
}
```

Etape 2 – Construction du modèle (30')

En vous aidant du screencast et à partir d'une base de données fournie (cf ressources), construisez le modèle de cette la base fournie. Instanciez ce modèle pour le passer lors de l'instanciation du VueModèle puis attachez le VueModèle à la vue.

Etape 3 – (30')

Ajouter les filtres suivants :

- « Anniversaires » qui affiche les jours et les âges des employés dont l'anniversaire est en janvier,
- Trier par âge croissant en utilisant orderby et let pour éviter de refaire le calcul,
- « Commandes françaises » qui affiche pour chaque client Français le nom et le nombre de commandes,
- « Prix moyen par catégorie » qui utilise une lambda expression dans la méthode Average pour afficher les prix unitaires moyens par catégorie.

Etape 4 – Règles/données métier (30')

Vous pouvez compléter un modèle en écrivant une classe partielle (public partial class) qui porte le nom du modèle à compléter dans le même espace de nom.

Ajoutez une propriété Amount type decimal dans le modèle Order qui calcule le montant de la commande en additionnant le champ Freight si il est non nul à la somme des montants des Order_Detail (méthode Sum).

Vérifiez le calcul en ajoutant le filtre Commande : return _context.Orders;

Livrables

Votre travail individuel sur cet atelier sur GitHub avant la prochaine séance.

Ressources

- Base de données SQL Server Compact : Partage Google Drive : Entreprise.sdf
- Screencast : Construction/utilisation d'un modèle EntityFramework (VS 2010) : http://youtu.be/p3pAcBkAgFI