

Cours n°13

Programme du cours

| Activité | Durée |
|--|-------|
| Introduction à EntityFramework Core Code First | 5m |
| Installation de l'environnement | 15h |
| Création d'un modèle | 10m |
| Création du contexte | 10m |
| Gestion des migration | 15m |

EntityFramework Core code first

Objectif

Générer une base de données à partir de classes C#

Introduction au code first

L'objectif de cette section est de présenter Entity Framework Core (EF Core) et le paradigme Code First.

- EF Core est un framework d'accès aux données pour .NET, qui simplifie les opérations de la base de données en traduisant les requêtes LINQ en SQL.
- Code First est une approche où la structure de la base de données est déterminée par votre code, plus précisément par vos classes de modèle.

Mise en œuvre

- Dans une nouvelle solution, ajouté un projet bibliothèque de classes nommé DataLayer
- Installer les paquets Nuget « Microsoft.EntityFrameworkCore.SqlServer »
- Installer la CLI EF Core en effectuant la commande « dotnet tool install --global dotnet-ef » dans le CMD
- Créer un projet ASP.net Core MVC et installer lui les paquets Microsoft.EntityFrameworkCore.SqlServer et Microsoft.EntityFrameworkCore.Design
- Ajouter une référence de projet entre le projet ASP.Net Core et le projet DataLayer

Création des modèles

Les modèles sont des classes qui représentent les tables de votre base de données.

- Chaque propriété dans une classe de modèle représente une colonne dans la table correspondante.
- Par exemple, si vous avez une table Users, vous pourriez avoir un modèle comme ceci :

```
0 références
public class User
{
    0 références
    public int Id { get; set; }
    0 références
    public string Username { get; set; }
    0 références
    public string Password { get; set; }
}
```

Création du contexte

Le contexte de la base de données est la classe principale qui coordonne les fonctionnalités d'EF Core pour un modèle de données donné.

- Vous créez cette classe en dérivant de DbContext et en définissant des DbSet<T> pour vos modèles.

Par exemple :

```

public class ApplicationDbContext : DbContext
{
    0 références
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }

    0 références
    public DbSet<User> Users { get; set; }
    // autres DbSets...
}

```

- Crée une base de données via Visual studio
- Dans votre projet ASP ajouter la ligne suivante pour ajouter le contexte

```

builder.Services.AddDbContext<MonSuperContext>(options =>
    options.UseSqlServer("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=<Chemin Vers LE MDF>;Integrated Security=True"));

```

Gestion des migrations

Les migrations sont la façon dont Code First crée une base de données correspondant à vos modèles, et comment elle met à jour cette base de données en réponse aux changements de vos modèles.

Vous pouvez créer une migration avec la commande « dotnet ef migrations add <NomDeLaMigration> --project ../DataLayer » commande « dotnet ef database update ». Ces commandes doivent être fait dans le cmd au niveau du projet ASP.Net Core

Liens vers d'autres tables

Nous allons ajouter un lien vers une autre table.

- Ajoutez un nouveau modèle

```

0 références
public class Role
{
    0 références
    public int Id { get; set; }
    0 références
    public string Nom { get; set; }
}

```

- Dans le modèle User nous allons ajouter un lien vers ce modèle

```

1 référence
public class User
{
    0 références
    public int Id { get; set; }
    0 références
    public string Username { get; set; }
    0 références
    public string Password { get; set; }
    0 références
    public ICollection<Role> Roles { get; set; }
}

```

- Puis ajouter le DbSet dans le contexte

```
public class MonSuperContext : DbContext
{
    0 références
    public MonSuperContext(DbContextOptions<MonSuperContext> options)
        : base(options)
    {
    }

    0 références
    public DbSet<User> Users { get; set; }
    0 références
    public DbSet<Role> Roles { get; set; }
    // autres DbSets...
}
```

- Relancez une migration puis mettez la base de données à jour comme vu dans la section précédente