ASP.NET Core : Scaffolding (Web API + EF)

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

elmouelhi.achref@gmail.com

Objectif

Génération d'un contrôleur **Web API** avec actions utilisant **Entity Framework**.

Création d'un projet **Web API** avec **Visual Studio Community 2022**

- Allez dans `Fichier > Nouveau > Projet`
- Dans la zone de recherche, saisissez `web api`
- Sélectionner `API Web ASP.NET Core`
- Remplir le champs `Nom` par `CoursWebApiEF`
- Gardez les options par défaut
- Validez et attendre la fin de création du projet

Créons une première entité `Personne` dans le `Models`

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace CoursWebApiEF.Models
{
    public class Personne
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int Num { get; set; }
        public string? Nom { get; set; }
        public string? Prenom { get; set; }
        public int Age { get; set; }
    }
}
```

Remarque

Il est possible de générer le code d'un contrôleur **Web API**.

Remarque

Il est possible de générer le code d'un contrôleur **Web API**.

Pour cela, il faut

- Faire clic droit sur le répertoire `Controllers` et aller dans `Ajouter > Contrôleur`
- Choisir `Contrôleur d'API avec actions, utilisant Entity Framework`
- Dans `Classe de modèle` sélectionner `Personne` et ajouter une nouvelle classe de contexte
- Garder le nom `PersonnesController` pour le contrôleur puis valider

Chaîne de connexion ajoutée dans `appsettings.json`

```
"ConnectionStrings": {
    "CoursWebApiEFContext": "Server=(localdb)\\mssqllocaldb;Database=
      CoursWebApiEFContext-278ce80c-4f11-44ad-9180-98ae0f87b8d6;
      Trusted_Connection=True;MultipleActiveResultSets=true"
}
```

Chaîne de connexion ajoutée dans `appsettings.json`

```
"ConnectionStrings": {
    "CoursWebApiEFContext": "Server=(localdb)\\mssqllocaldb;Database=
      CoursWebApiEFContext-278ce80c-4f11-44ad-9180-98ae0f87b8d6;
      Trusted_Connection=True;MultipleActiveResultSets=true"
}
```

Chargement de la chaîne de connexion dans `Program.cs`

```
builder.Services.AddDbContext<CoursWebApiEFContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("
      CoursWebApiEFContext")));
```

Contexte généré dans `Data/CoursWebApiEFContext.cs`

```
using Microsoft.EntityFrameworkCore;
using CoursWebApiEF.Models;

namespace CoursWebApiEF.Data
{
    public class CoursWebApiEFContext : DbContext
    {
        public CoursWebApiEFContext (DbContextOptions<CoursWebApiEFContext> options)
            : base(options)
        {
        }

        public DbSet<CoursWebApiEF.Models.Personne> Personne { get; set; }
    }
}
```

Remarque

Pour créer la base et les tables, il faut faire une migration..

Mise en place de migrations

Dans le menu `Outils`, aller dans `Gestionnaire de package NuGet` et sélectionner `Console du gestionnaire de paquet`.

Mise en place de migrations

Dans le menu `Outils`, aller dans `Gestionnaire de package NuGet` et sélectionner `Console du gestionnaire de paquet`.

Dans la console `Console du Gestionnaire de Package`

- Sélectionner le projet dans la liste `Projet par défaut`.
- Lancer la commande `Add-Migration Init` : ça génère un répertoire `Migrations` contenant un fichier `date_Init.cs` .
- Pour créer la base de données : lancer la commande `Update-Database`.

Lancer le projet et tester les différentes actions de notre contrôleur `PersonnesController`.