

Entity Framework : créer les objets Models à partir d'une base de données

I) Mise en place de la base de données

Dans une base de données de votre choix, créer la table personnes à l'aide de [ce script](#).

Vérifier que la table est bien créée et qu'elle contient des données.

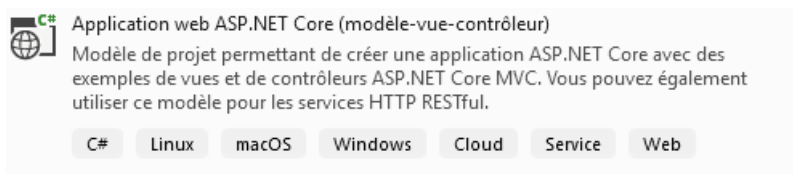
II) Créer un nouveau projet C#

II.1) La solution

Créer un dossier pour votre projet puis clic-droit Ouvrir avec Visual Studio

Fichier, Nouveau, Projet

Sélectionner le modèle



Donner un nom au projet, garder la version courante du framework et cliquer sur créer

II.2) Les dépendances nuget

Ajouter les dépendances au projet en cliquant sur Outils, Gestionnaire de package NuGet, Gérer les packages nugets pour la solution

[MySQL.EntityFrameworkCore](#)

[Microsoft.EntityFrameworkCore.Tools](#)

[Microsoft.EntityFrameworkCore](#)

Vous pouvez voir les packages inclus, en double cliquant sur le nom du projet (fichier nomProjet.csproj)

II.3) Le contexte

La classe contexte permet de créer la relation (session) avec la base de données. Elle étend la classe `System.Data.Entity.DbContext.DbContext`. La classe de contexte est utilisée pour interroger ou enregistrer des données dans la base de données. Il est également utilisé pour configurer les classes de domaine, les mappages liés à la base de données, les paramètres de suivi des modifications, la mise en cache, les transactions, etc.

- Ajouter le service DbContext au projet
 - Nous éviterons de mettre notre chaîne de connexion directement dans la méthode de configuration de service. Nous allons créer une entrée dans le fichier json de configuration
 - L'entrée doit porter le nom `"ConnectionStrings"` obligatoirement avec un s. On peut configurer plusieurs bases de données pour différentes utilisations
 - Dans le fichier `appsettings.json`, ajouter l'entrée suivante.

```
"ConnectionStrings": {  
  "Default": "server=localhost;user=root;database=EFBaseToModel;port=3306;ssl  
mode=none" },
```

- Dans le fichier `Program.cs`, nous allons ajouter le service DbContext

```
builder.services.AddDbContext<MyDbContext>(options =>  
options.UseMySQL(Configuration.GetConnectionString("Default")));
```

II.4) Utiliser EntityFramework Scaffold

Lancer la commande suivante dans la commande du package manager

```
scaffold-DbContext -Connection name=default -Provider MySQL.EntityFrameworkCore -  
OutputDir Models/Data -Context MyDbContext -ContextDir Models
```

Les entités correspondantes aux tables de la base de données ont été créées.