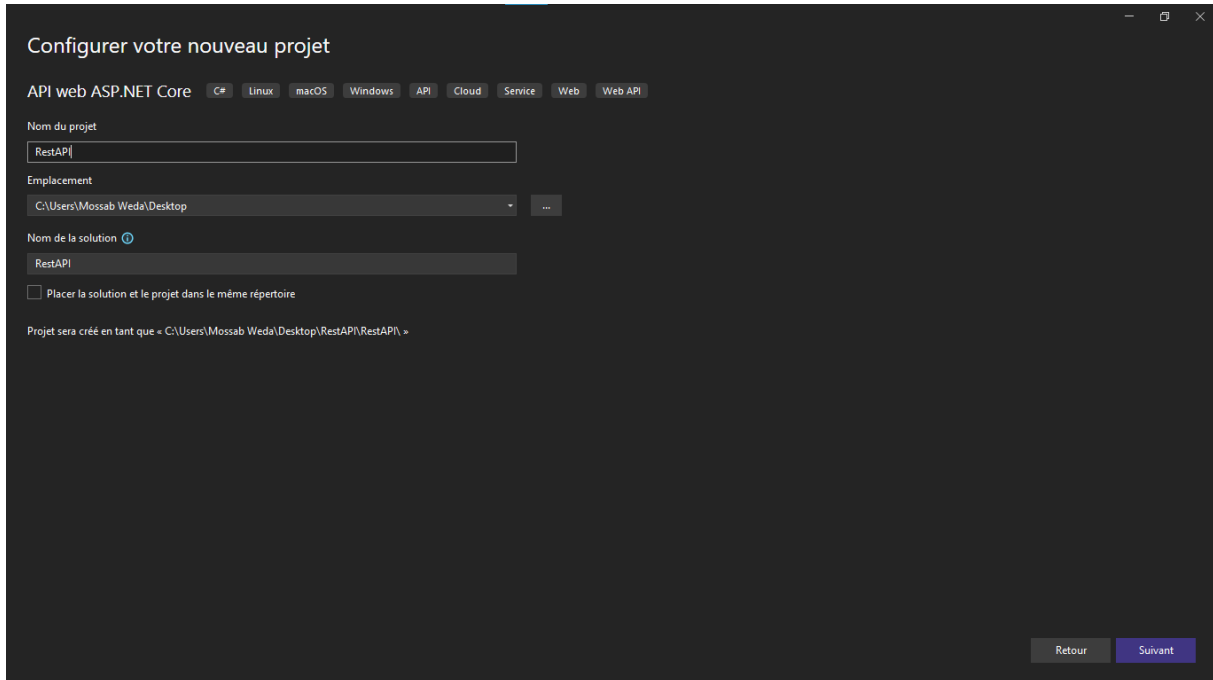


Atelier 3 SOA

1. Création du projet



Configurer votre nouveau projet

API web ASP.NET Core C# Linux macOS Windows API Cloud Service Web Web API

Nom du projet
RestAPI

Emplacement
C:\Users\Mossab Weda\Desktop

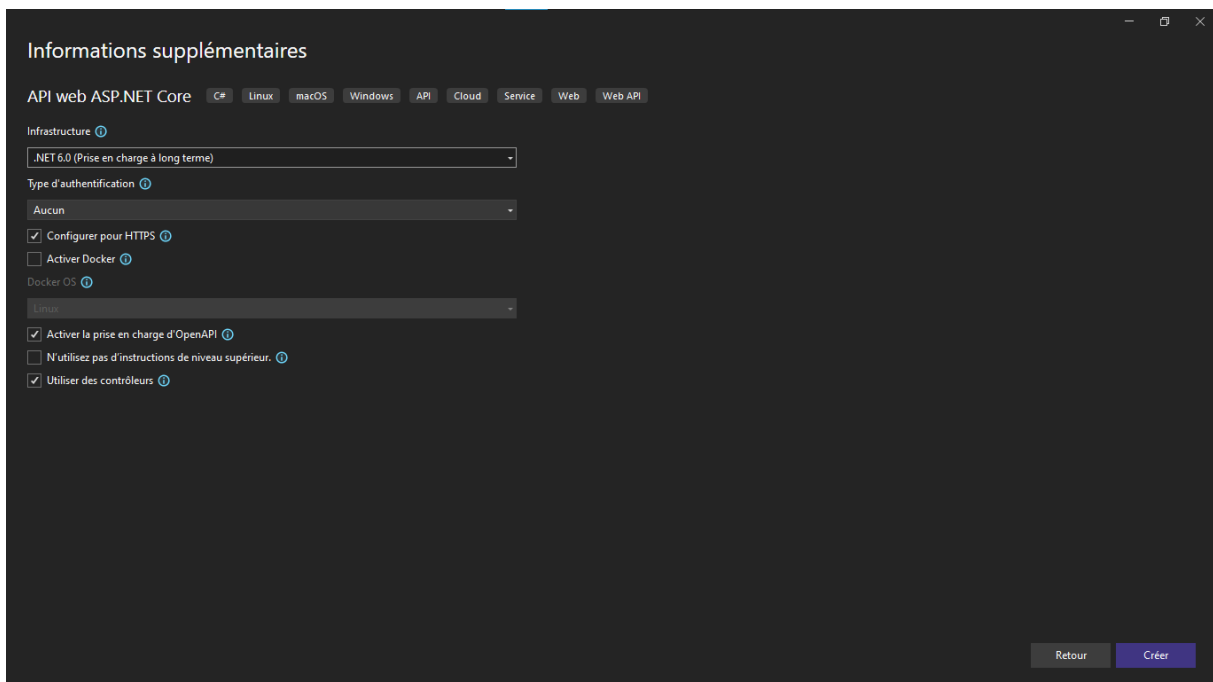
Nom de la solution
RestAPI

☐ Placer la solution et le projet dans le même répertoire

Projet sera créé en tant que « C:\Users\Mossab Weda\Desktop\RestAPI\RestAPI »

Retour Suivant

Figure 1 : Configuration du projet



Informations supplémentaires

API web ASP.NET Core C# Linux macOS Windows API Cloud Service Web Web API

Infrastructure
.NET 6.0 (Prise en charge à long terme)

Type d'authentification
Aucun

☒ Configurer pour HTTPS

☐ Activer Docker

Docker OS
Linux

☒ Activer la prise en charge d'OpenAPI

☐ N'utilisez pas d'instructions de niveau supérieur.

☒ Utiliser des contrôleurs

Retour Créer

Figure 2 : Informations Supplémentaires

2. Installation des Packages

Microsoft.EntityFrameworkCore : Ce package est le cœur d'Entity Framework Core. Il fournit les API de base pour le développement d'applications qui nécessitent une interaction avec une base de données

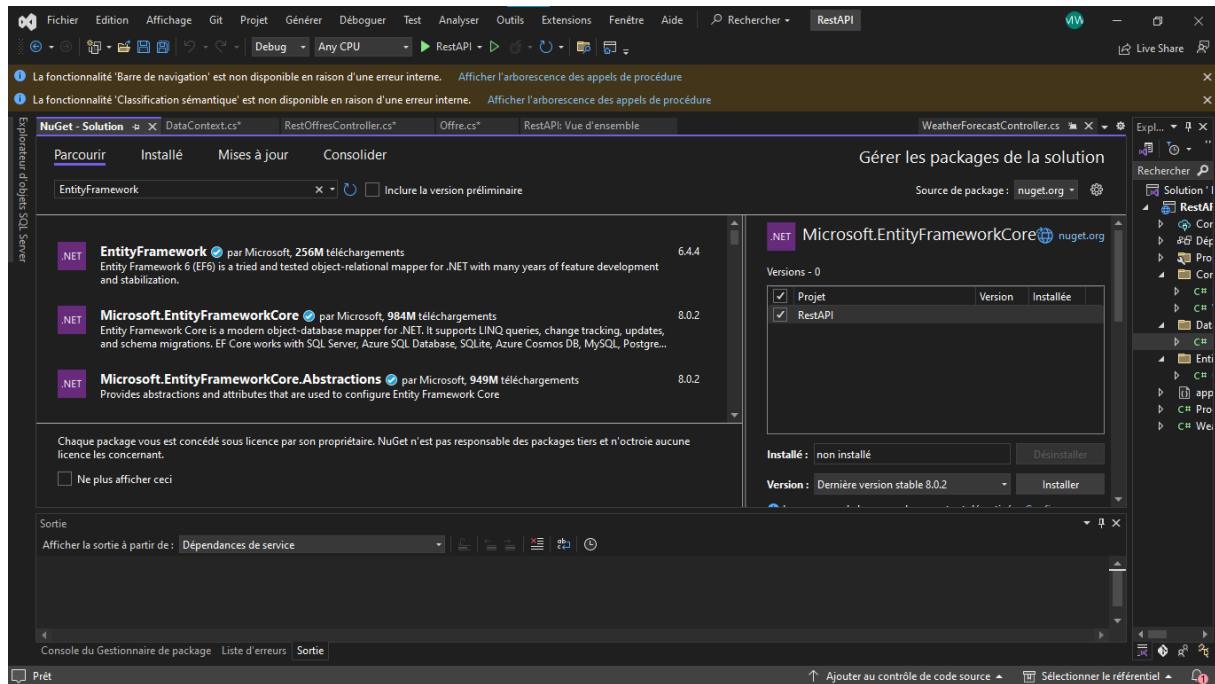


Figure 3 : installer Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.Tools : Ce package fournit des outils qui sont essentiels pour la génération de migrations, l'application des migrations à la base de données, et la génération de modèles à partir de bases de données existantes.

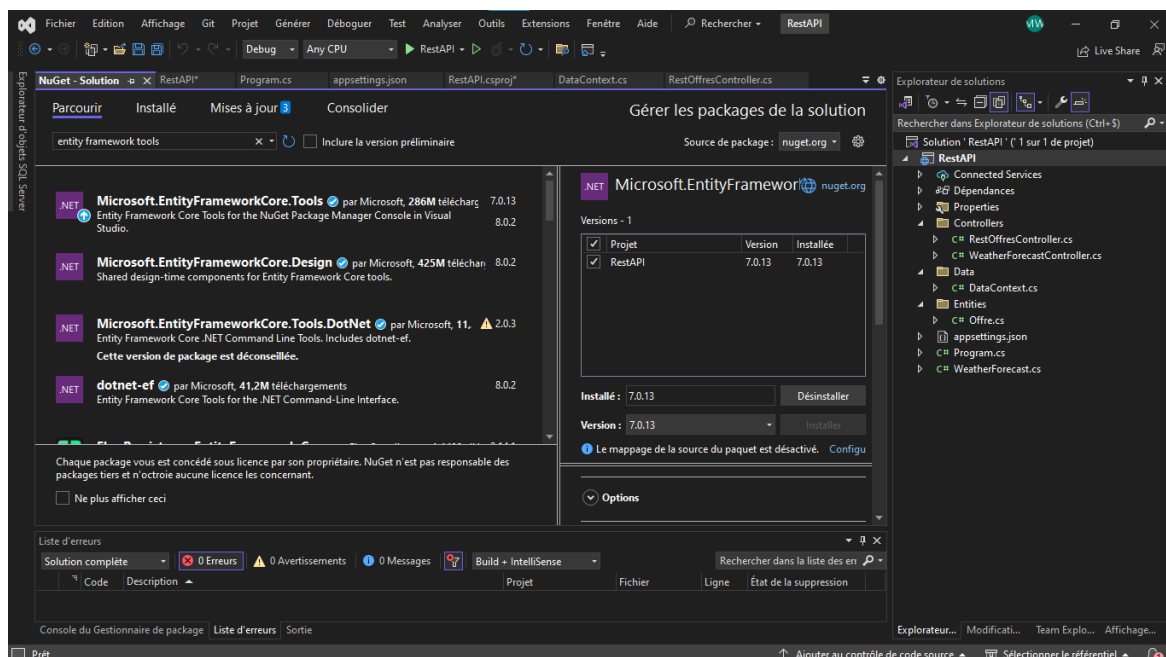


Figure 4 ; Installer Microsoft.EntityFrameworkCore.Tools

Microsoft.EntityFrameworkCore.SqlServer : Ce package est spécifique au fournisseur de base de données SQL Server pour Entity Framework Core. Il permet à Entity Framework Core de communiquer avec SQL Server.

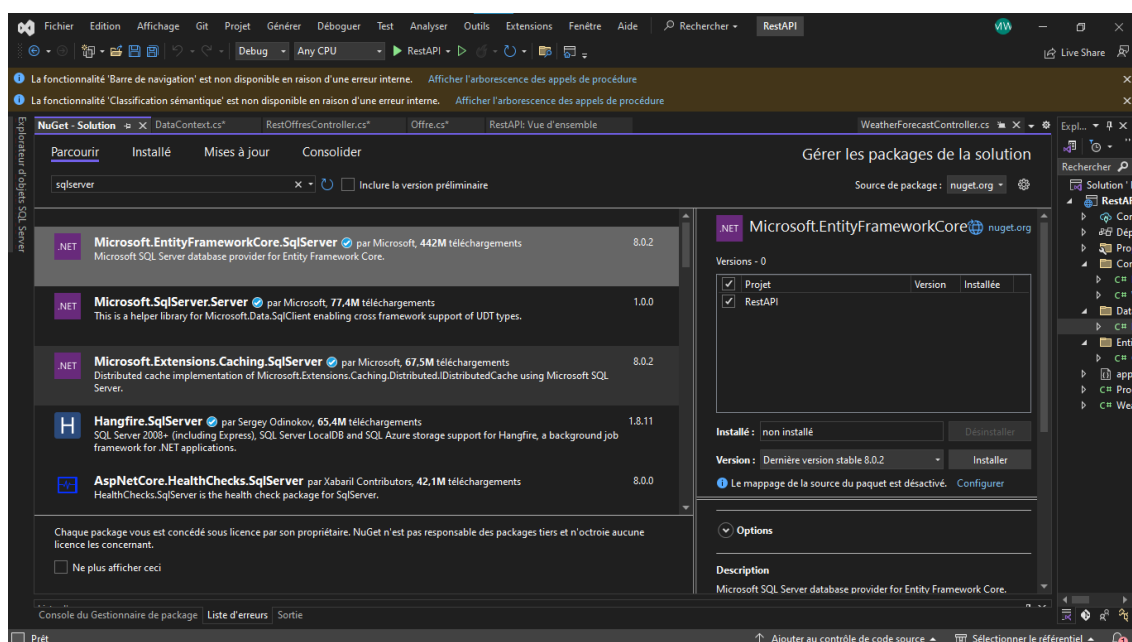


Figure 5 : Installer Microsoft.EntityFrameworkCore.SqlServer

3. Création du Classe Offre

Remarque : Le changement de nom de code en Id représente la clé primaire d'une entité dans .NET

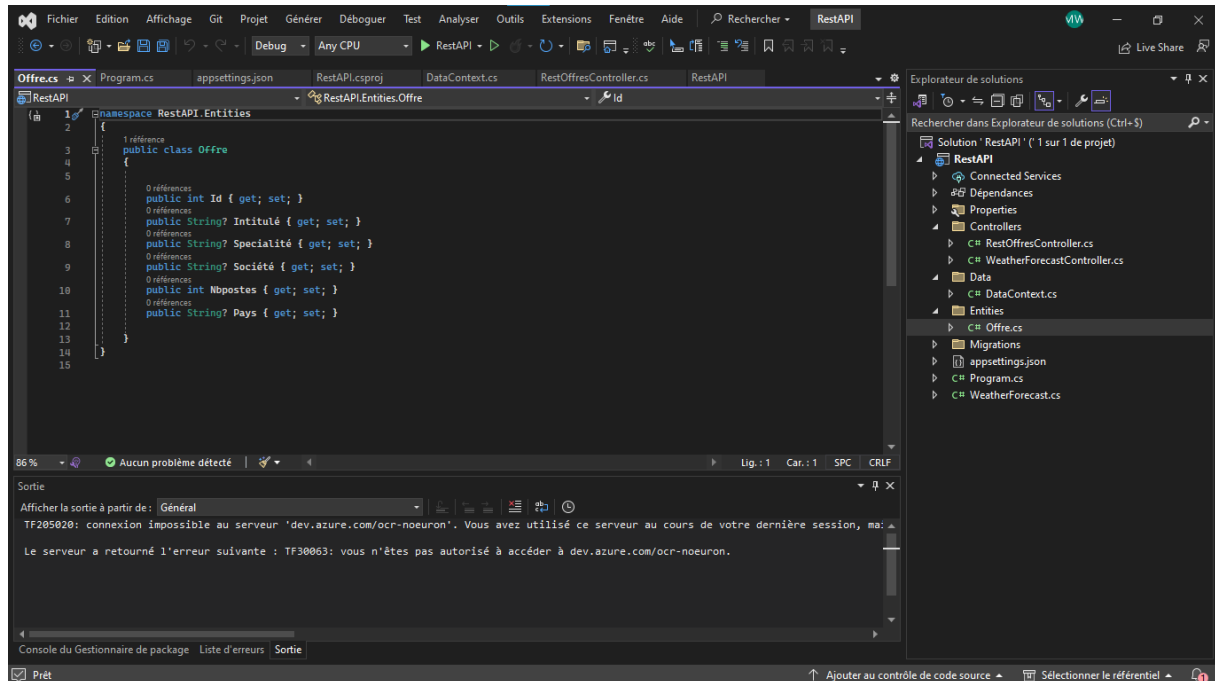


Figure 6 : La création du classe Offre

4. L'ajout du Data Context

DataContext : Le DataContext est la source de toutes les entités mappées sur une connexion à la base de données.

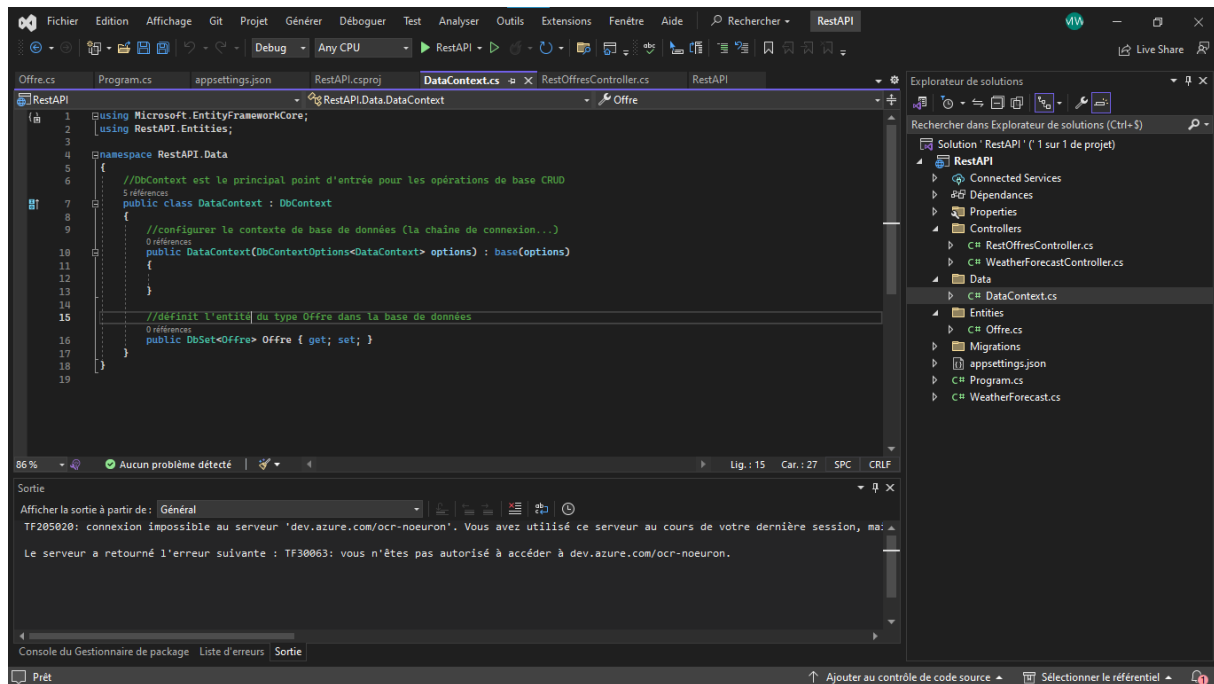


Figure 7 : L'ajout du Data Context

5. Configuration de la Chaîne de Connexion

- **DefaultConnection** : c'est le nom de la chaîne de connexion.
- **Server=DESKTOP-0AHL5Q0\ATELIER_ENTREPOT** : spécifie que la base de données.
- **Database=DotnetDB** : spécifie le nom de la base de données
- **Trusted_Connection=true** : signifie que la connexion utilise l'authentification Windows intégrée.
- **TrustServerCertificate=true** : c'est une option spécifique à SQL Server qui permet de faire confiance au certificat SSL du serveur sans validation.

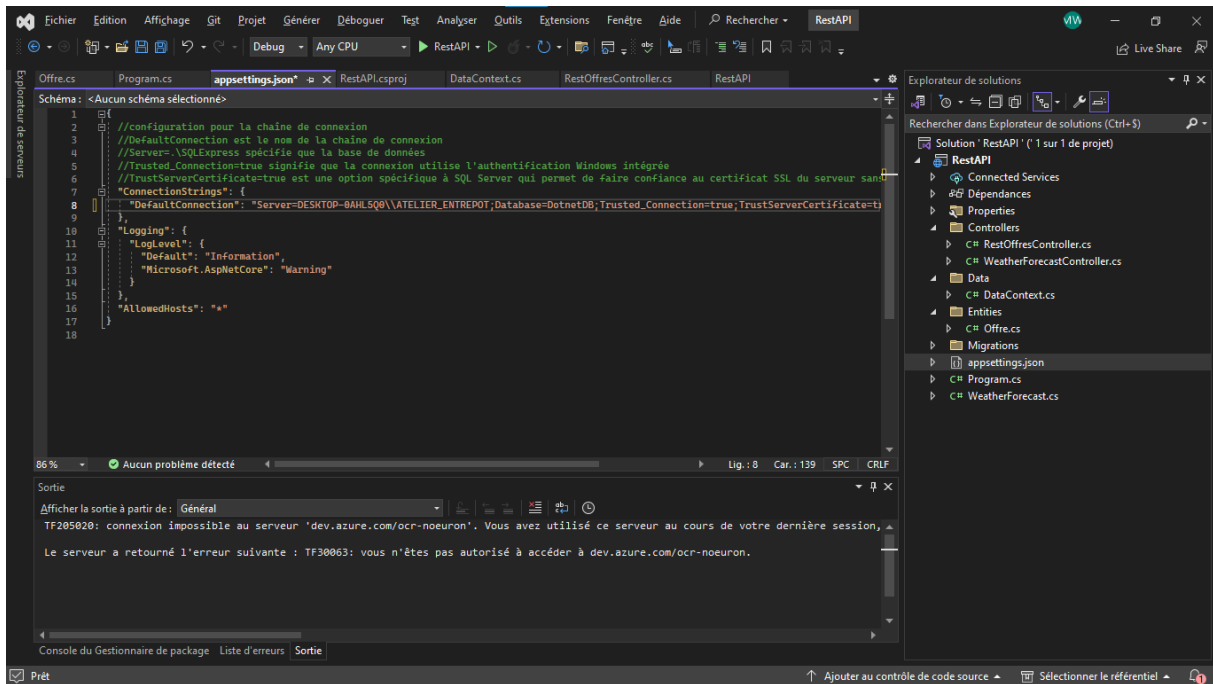


Figure 8 : Configuration de la Chaîne de Connexion

6. Configurer et Enregistrer le DataContext dans le conteneur de services

Configuré le DataContext pour utiliser SQL Server avec la chaîne de connexion nommée DefaultConnection provenant des configurations de l'application.

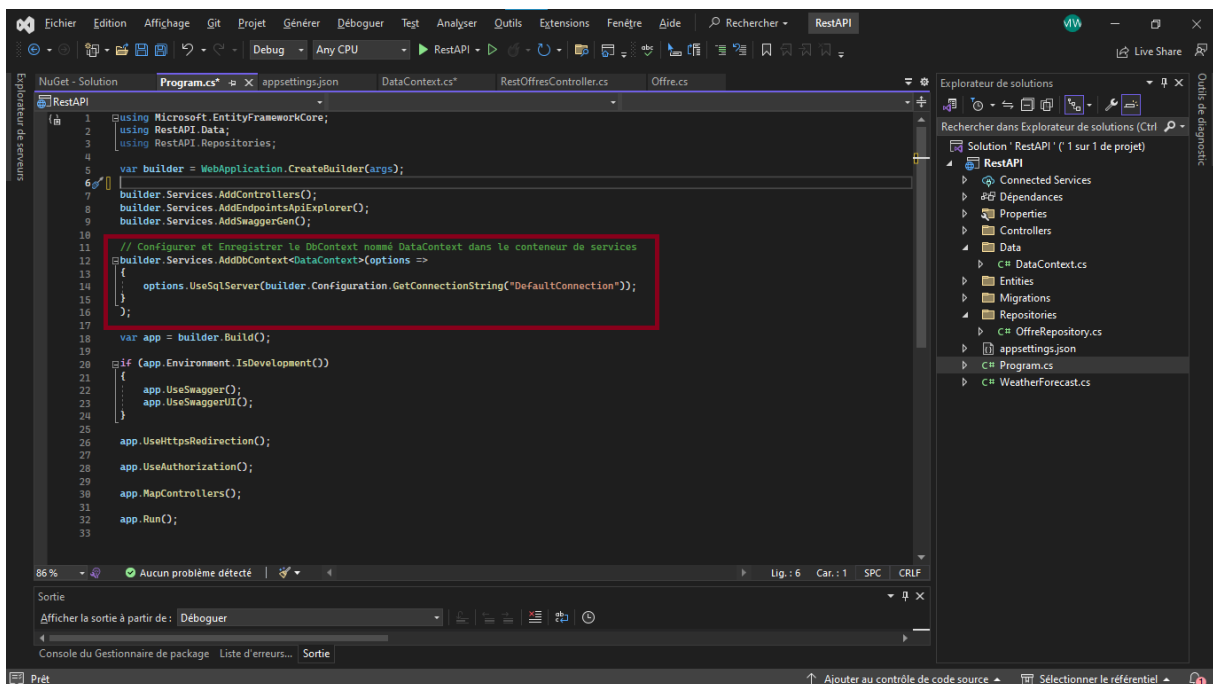


Figure 9 : Enregistrer le DataContext dans Program.cs

7. Initialisation la Migration et Mise à Jour de la Base de Données

Add-migration First : c'est une étape clé dans le flux de travail de développement Code First, permettant de générer le code nécessaire pour appliquer les modifications du modèle de données à la base de données, et facilitant ainsi la création et la mise à jour de la base de données selon le modèle défini dans le code.

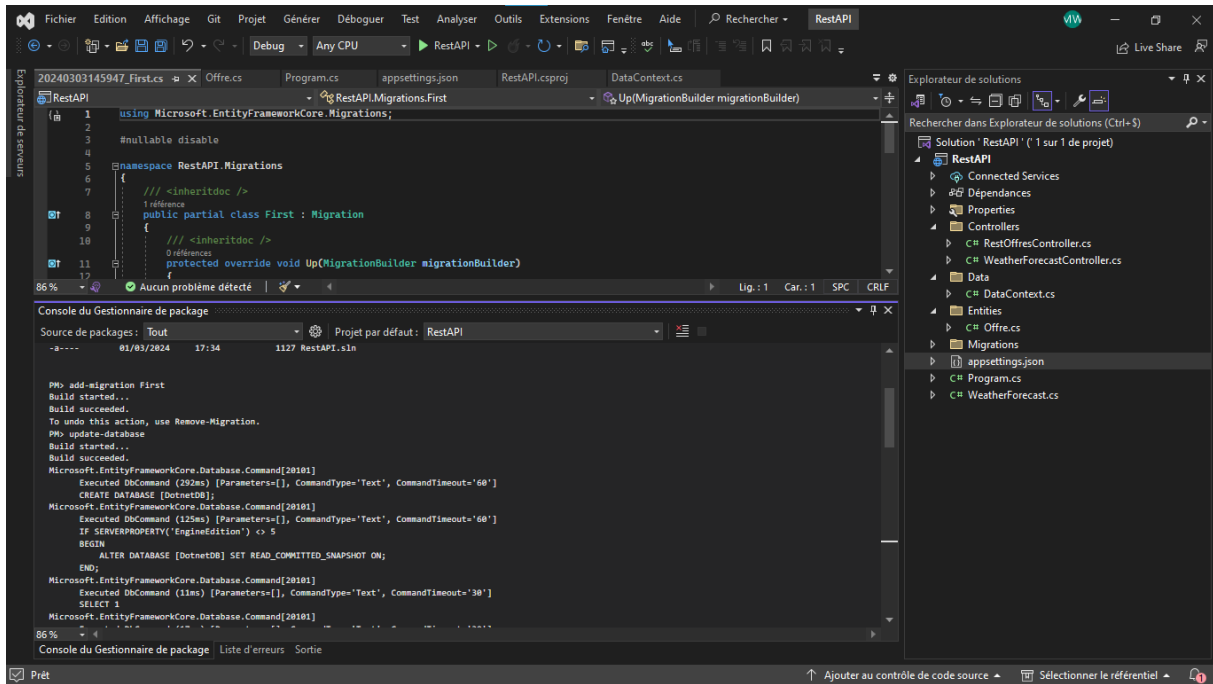


Figure 10 : Initialisation la Migration

Update-database : C'est une commande dans Entity Framework Core (EF Core) utilisée pour appliquer les migrations existantes à la base de données.

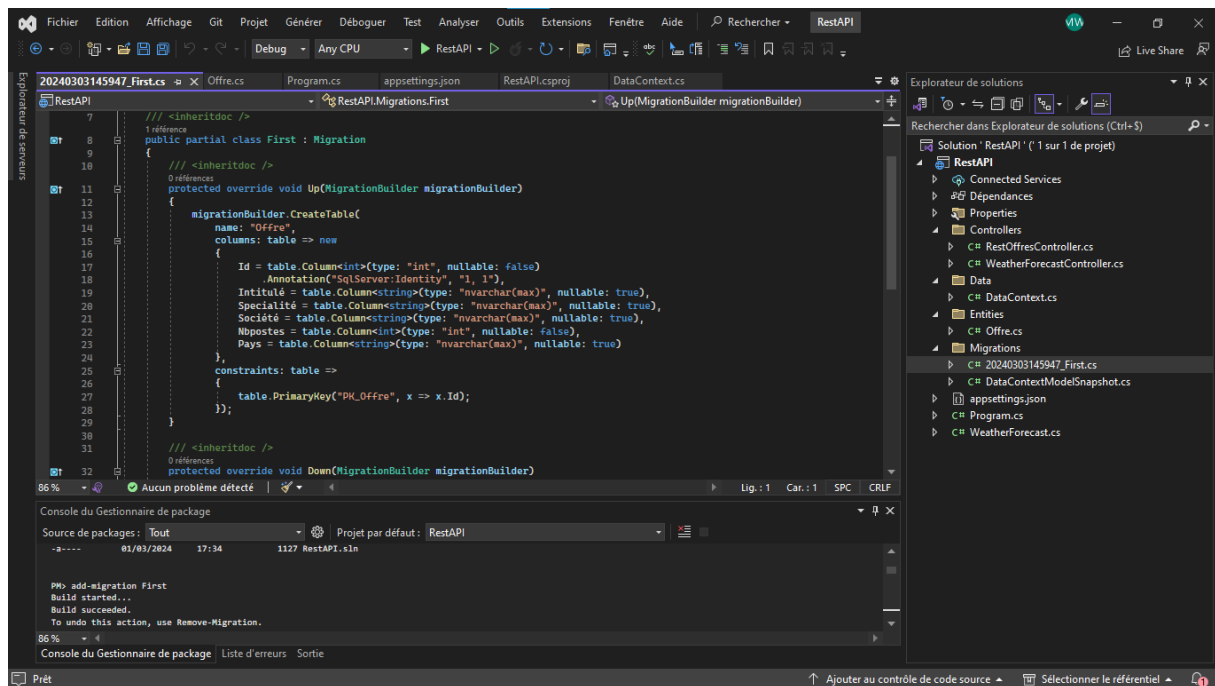


Figure 11 : Mise à Jour de la Base de Données

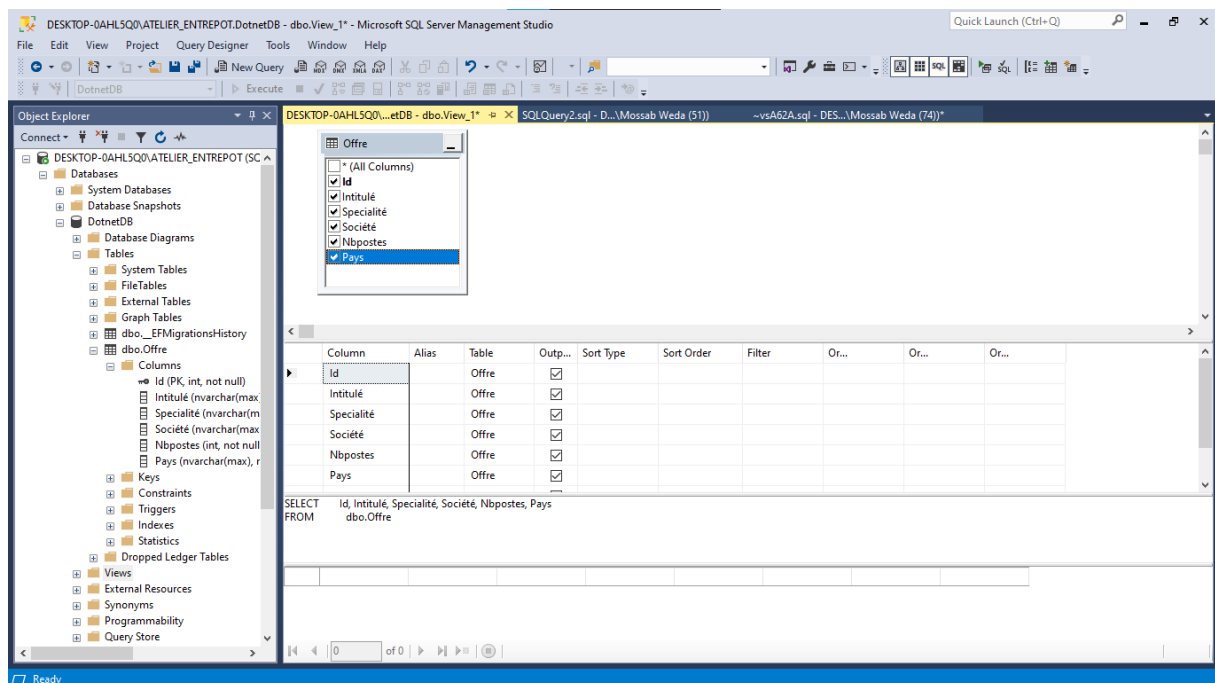


Figure 12 : Vérification de la création du table

8. Configuration et Création du Repository OffreRepository dans Program.cs

Program.cs : Ce fichier est le point d'entrée principal d'une application ASP.NET Core, où la configuration de l'application est définie et l'application est construite et exécutée.

builder.Services.AddScoped<OffreRepository>(); : Ce ligne est utilisée pour enregistrer le service OffreRepository dans le conteneur d'injection de dépendances (DI) de l'application ASP.NET Core.

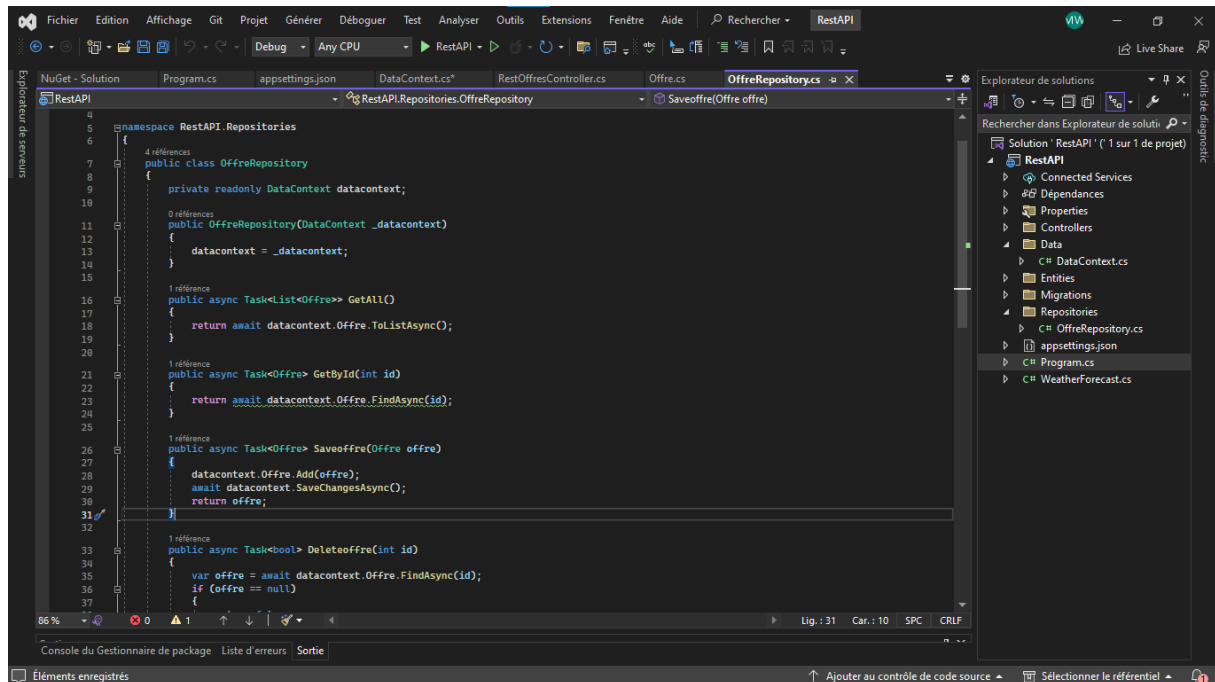


Figure 13 : Mise en place du Repository OffreRepository

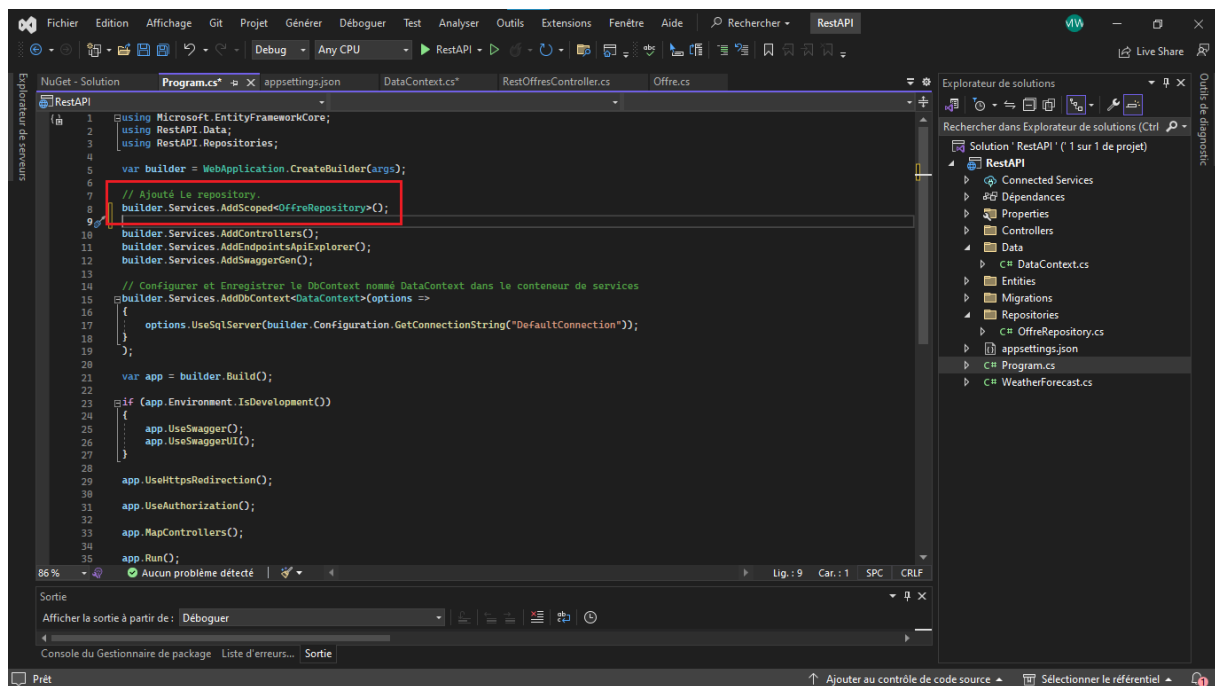


Figure 14 : Configuration du OffreRepository dans Program.cs

9. Définition du contrôleur « RestOffresController »

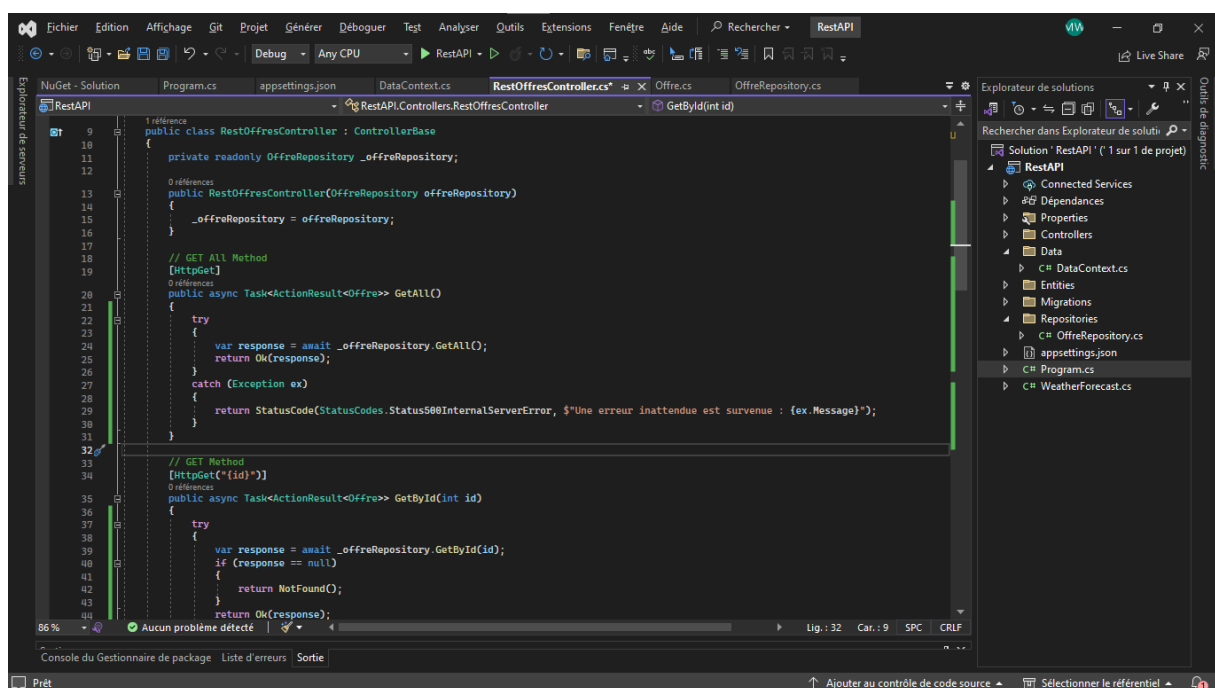


Figure 15 : Définition du contrôle

10. Test avec Postman

- Méthode GetAll :

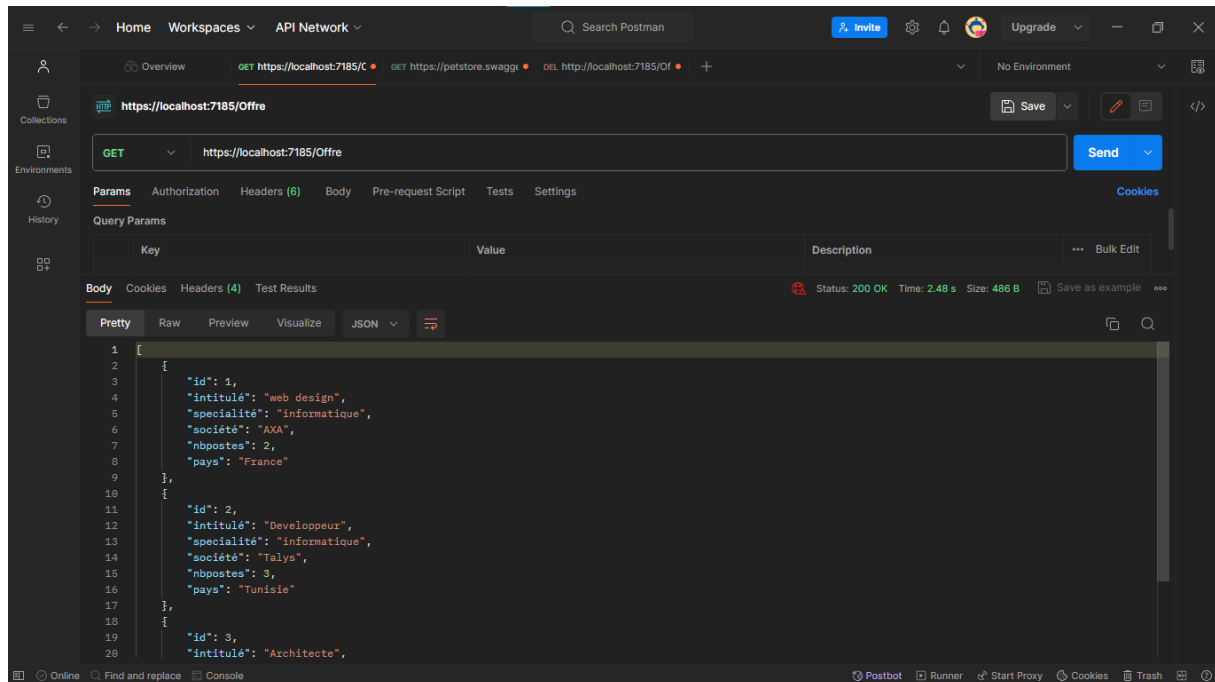


Figure 16 : Test de la méthode GetAll

- Méthode GetById :

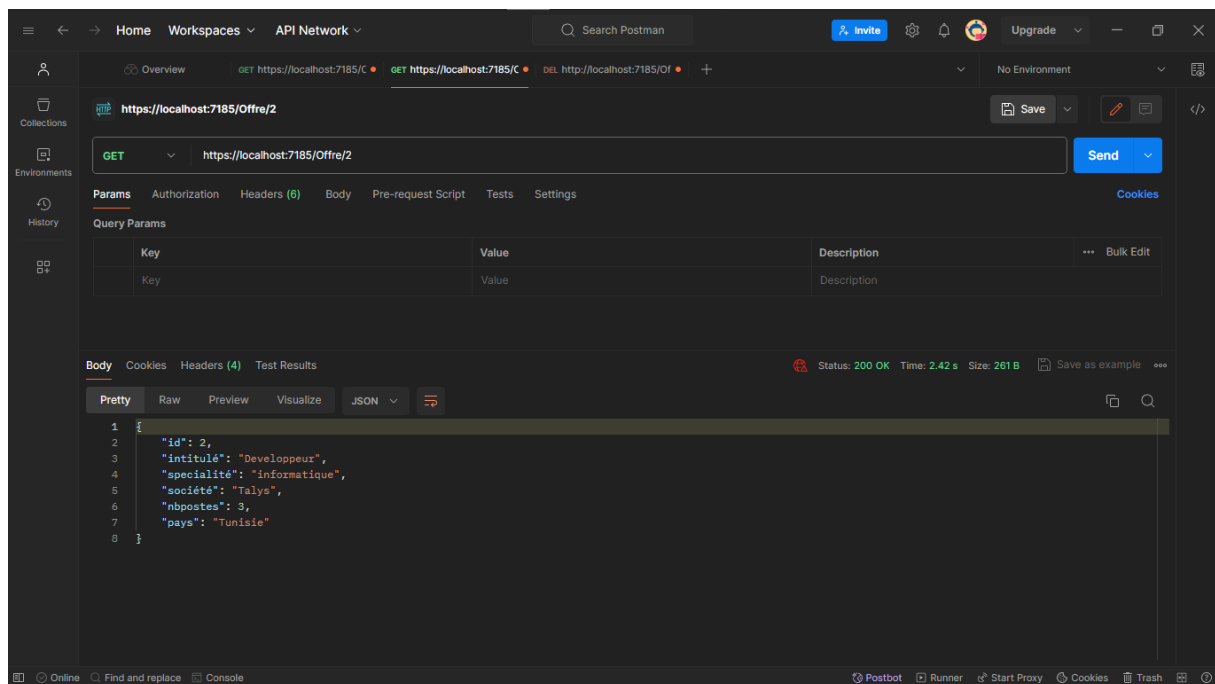


Figure 17 : Test de la méthode GetById

- **Méthode Post :**

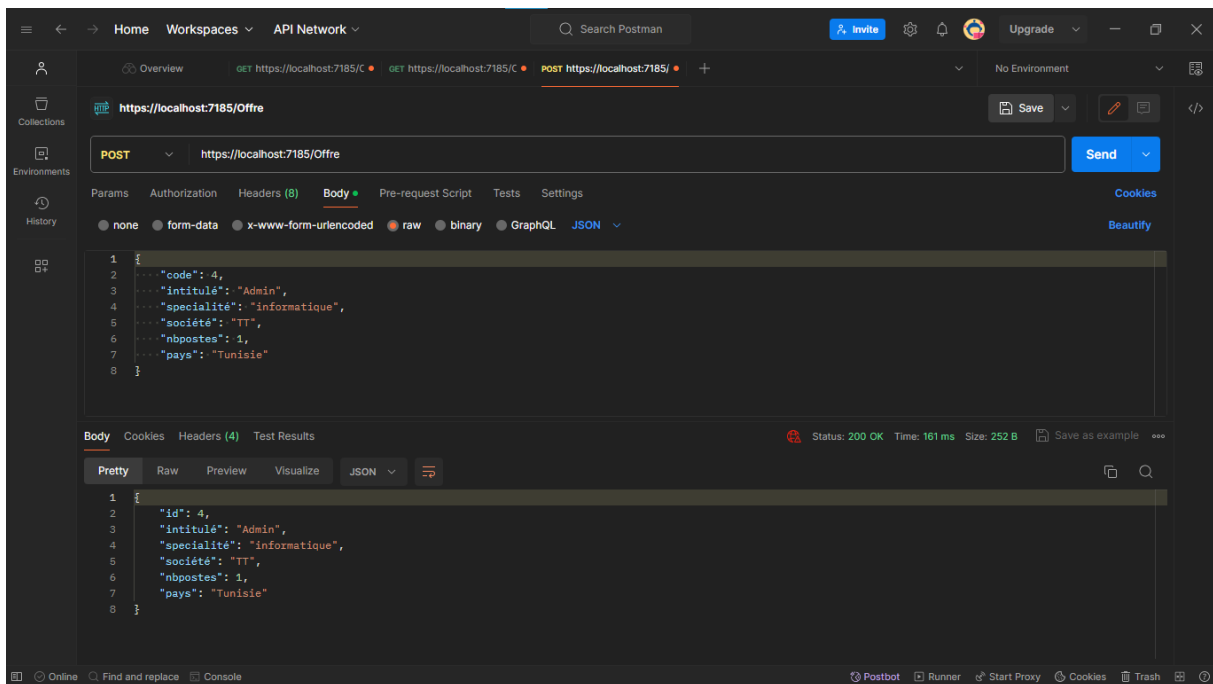


Figure 18 : Test de la méthode Post

- **Méthode Delete :**

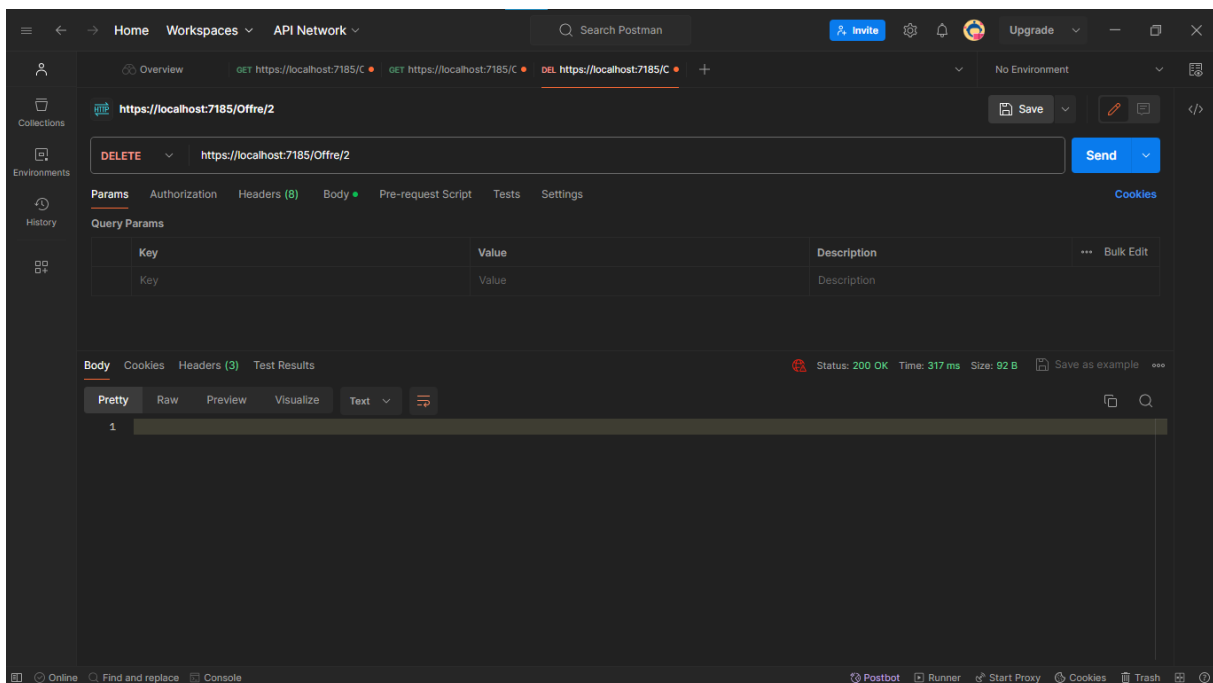


Figure 19 : Test de la méthode Post