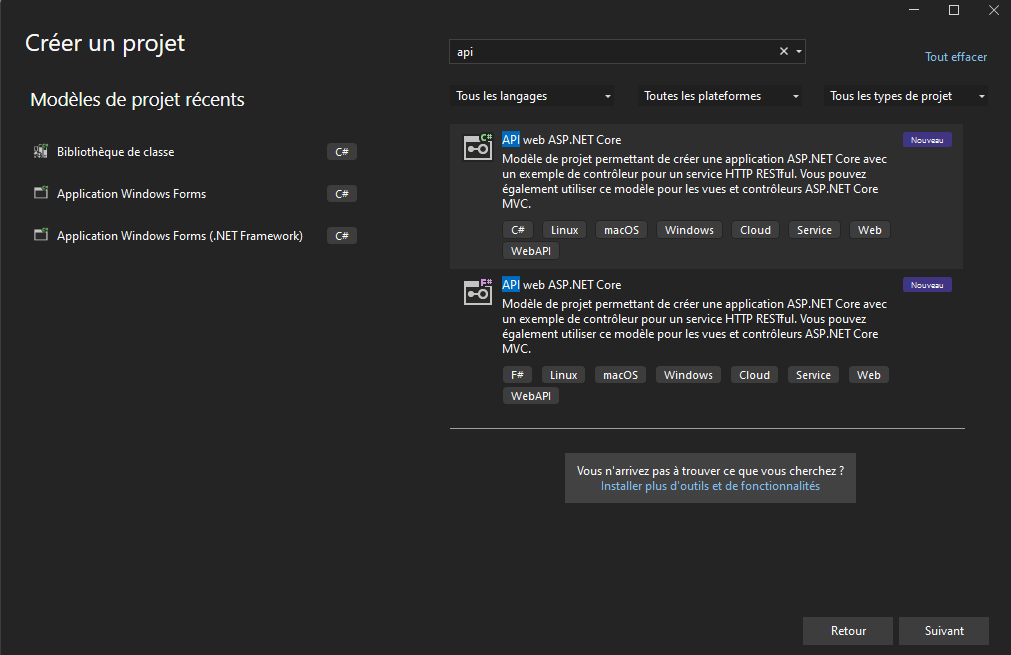
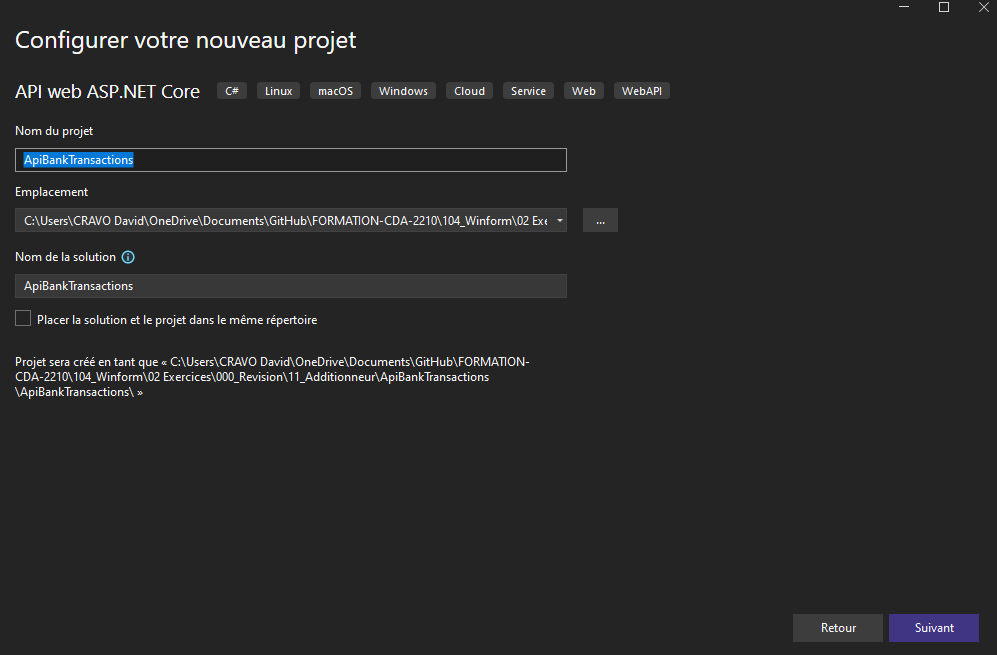
Prérequis :

* Installation de Swagger
* Installation de Postman

Créer un projet « API web ASP.NET Core »





Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Gérer les packages NuGet au niveau de la solution

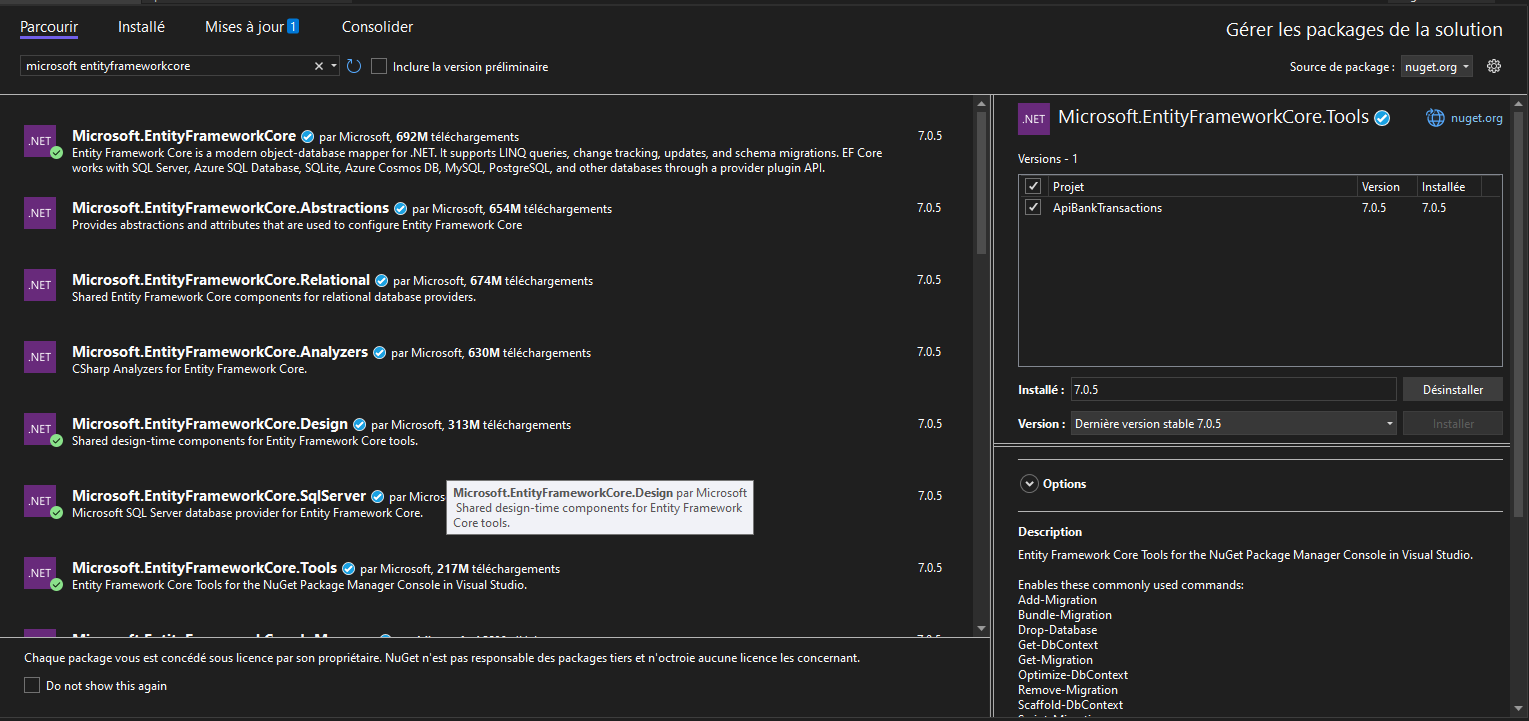
Parcourir -> Microsoft EntityFrameworkCore

->MicrosoftEntityFrameworkCore

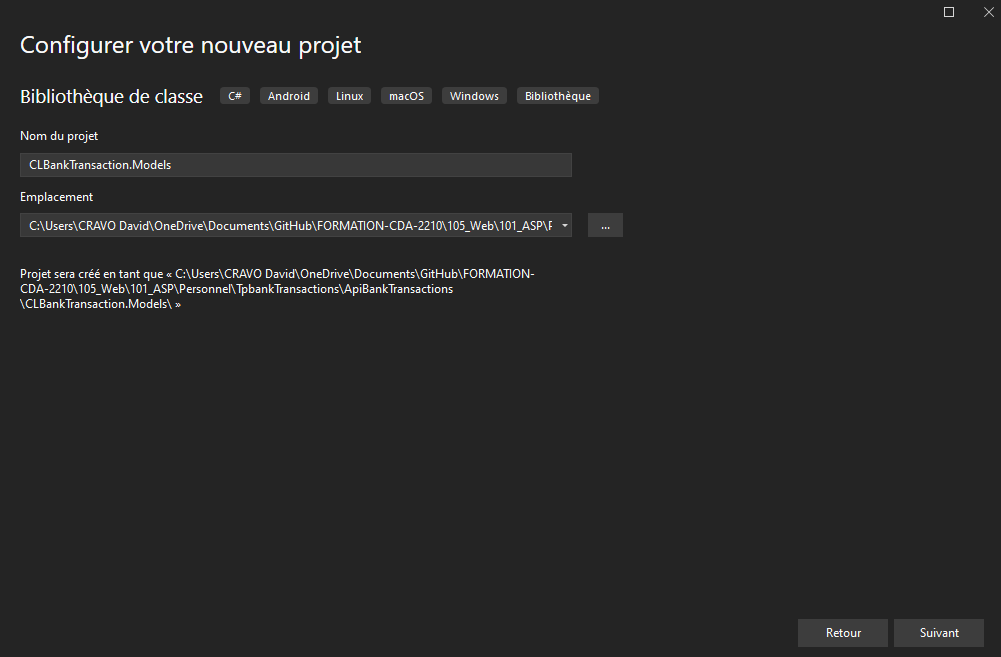
->MicrosoftEntityFrameworkCore.Design

->MicrosoftEntityFrameworkCore.SqlServer

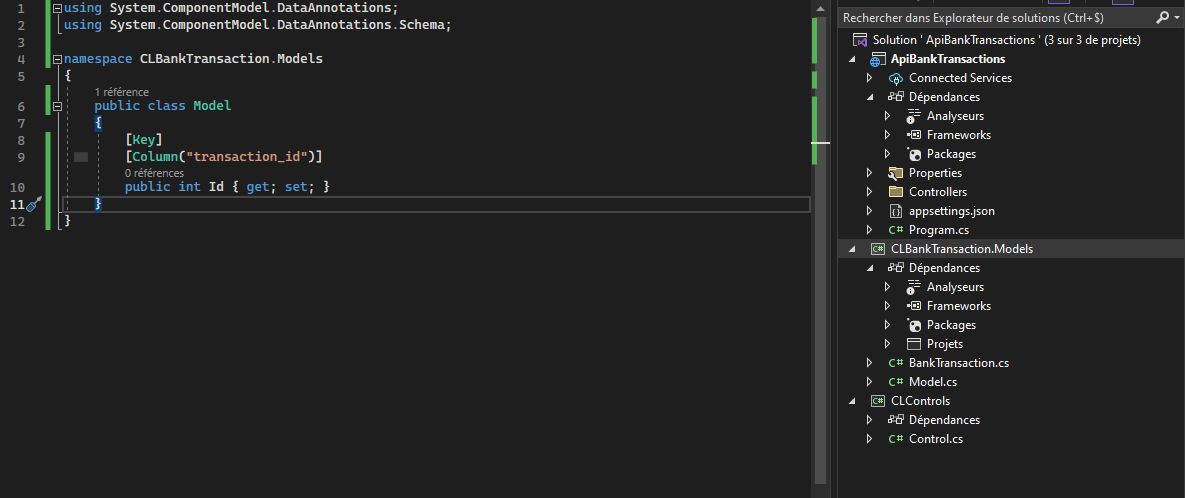
->MicrosoftEntityFrameworkCore.Tools



Ajouter -> Nouveau Projet -> Bibliothèque de classe -> CLBankTransaction.Models



Renomer « Class1 » en « Model » et implémenter la classe



Implémenter la classe BankTransaction avec les annotations nécessaires

-> rajouter le using System.ComponentModel.DataAnnotations;

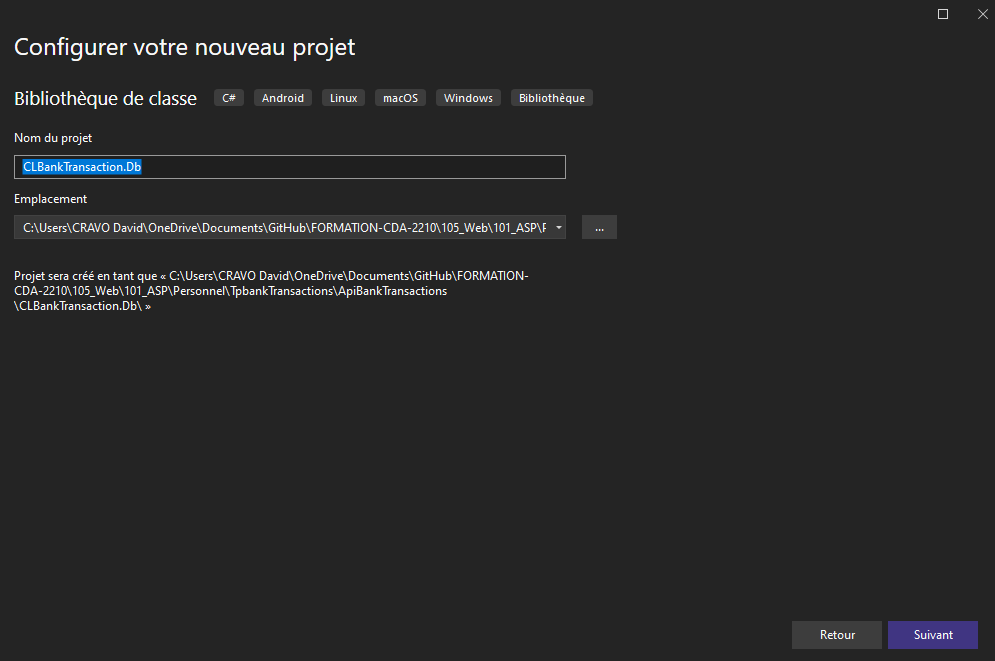
-> using System.ComponentModel.DataAnnotations.Schema;

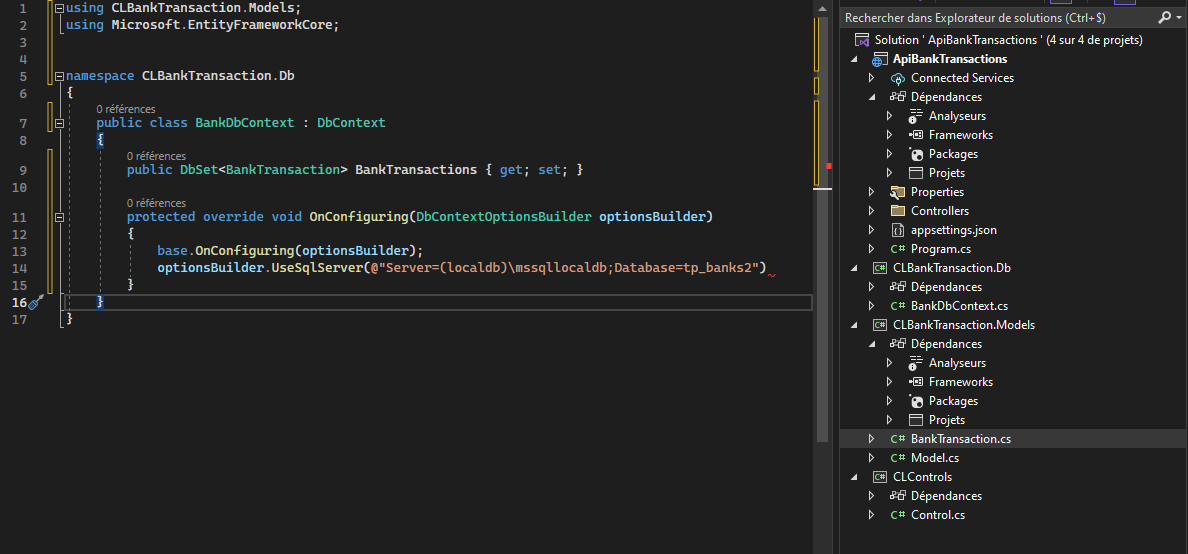
-> using Microsoft.EntityFrameworkCore;

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Implémenter la classe BankDbContext dans CLBankTransaction.Db



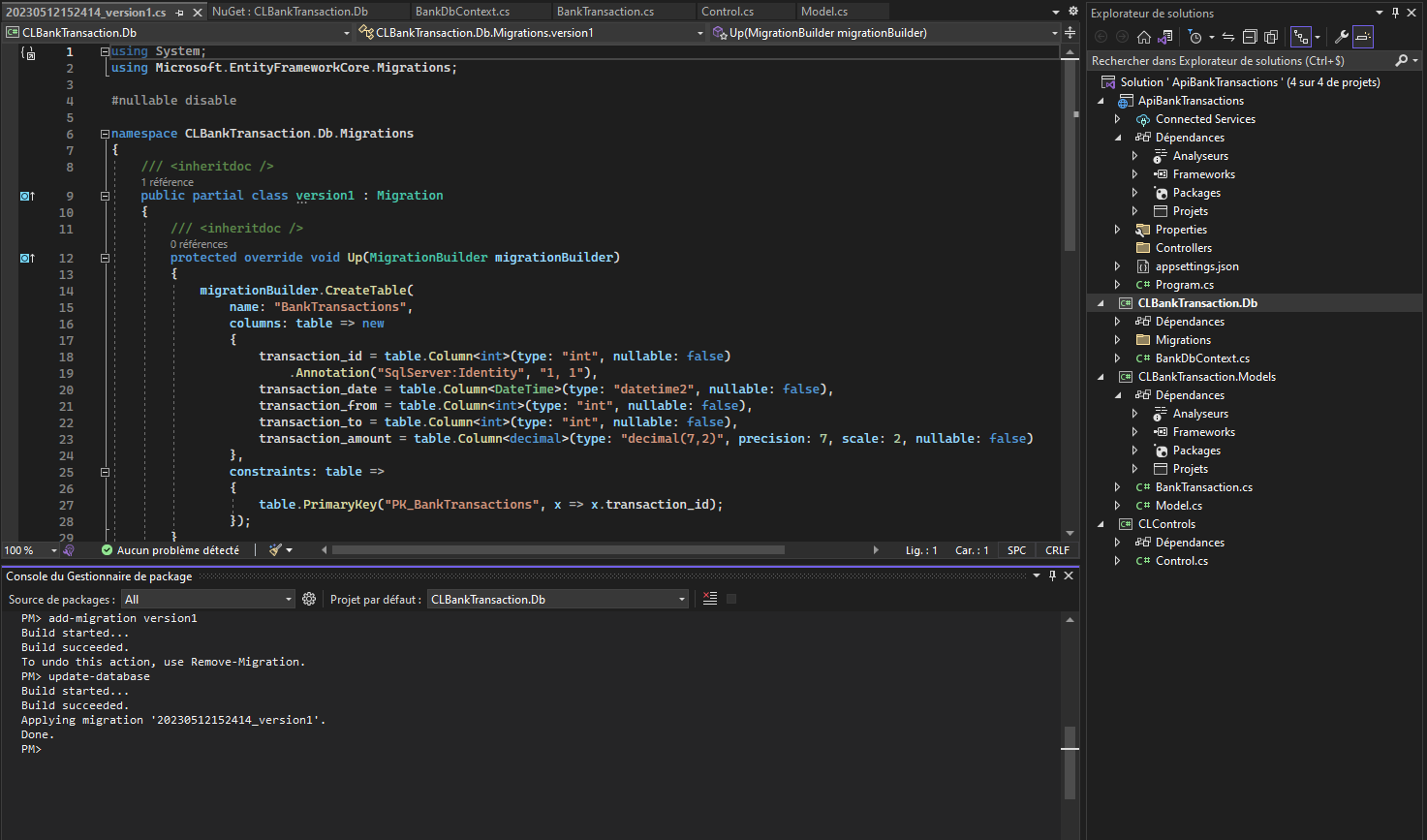


On génère ensuite la base de données dans la Console du Gestionnaire de package

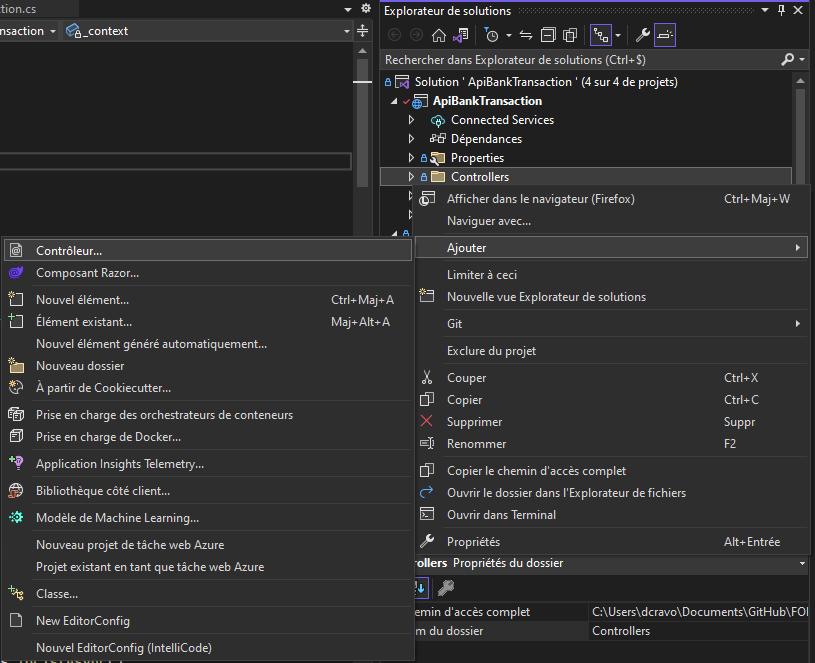
On s’assure d’abord que CLBankTransaction.Db est défini comme projet de démarrage

et qu’il est bien défini ncomme projet par défaut dans la Console du Gestionnaire de Package

On fait ensuite le « add-migration version1» et le « update-database »

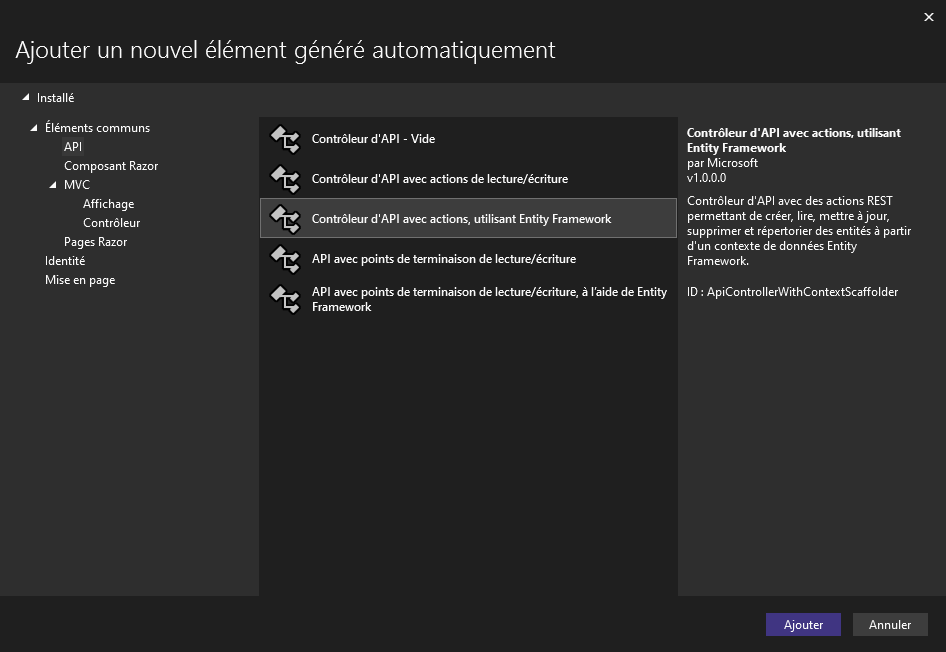


On ajoute un contrôleur dans le dossier Controllers



Dans l’explorateur de gauche on sélectionne API

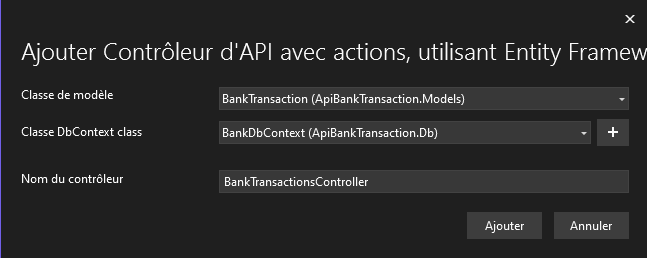
Puis dans la partie centrale le troisième contrôleur : Contrôleur D’API avec actions, utilisant Entity Framework



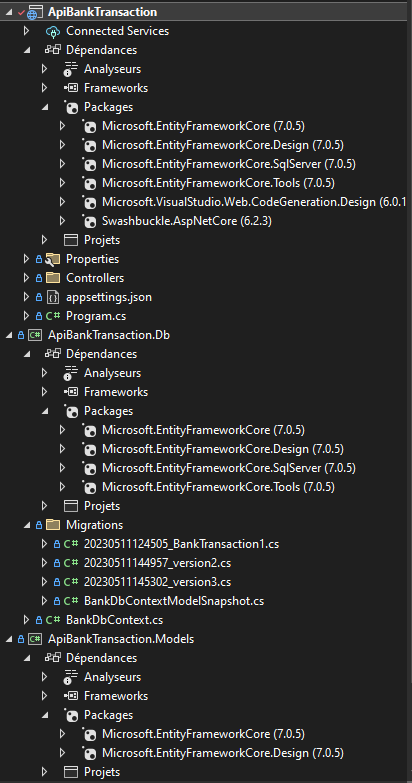
On sélectionne l’élément ApiBankTransaction.Models dans Classe de modèle

On sélectionne notre DbContext dans Classe DbContext class

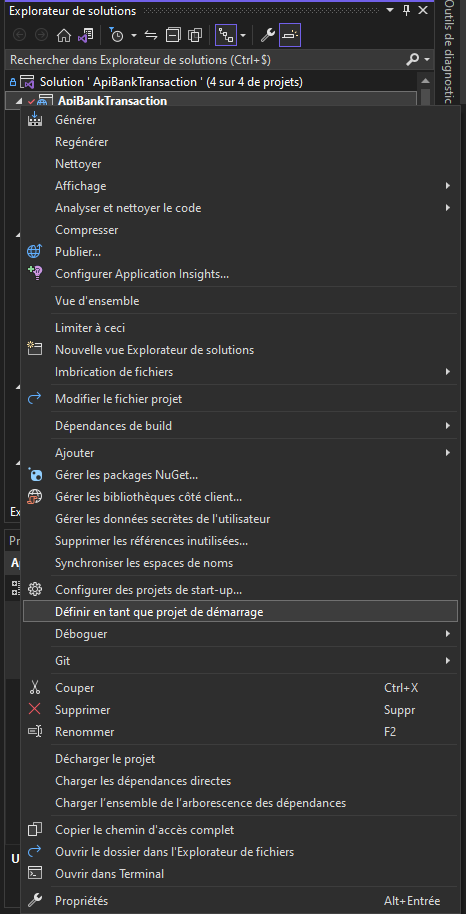
On peut modifier le nom si nécessaire mais à retenir que la partie du nom BankTransactions correspondra au chemin d’accès dans l’API



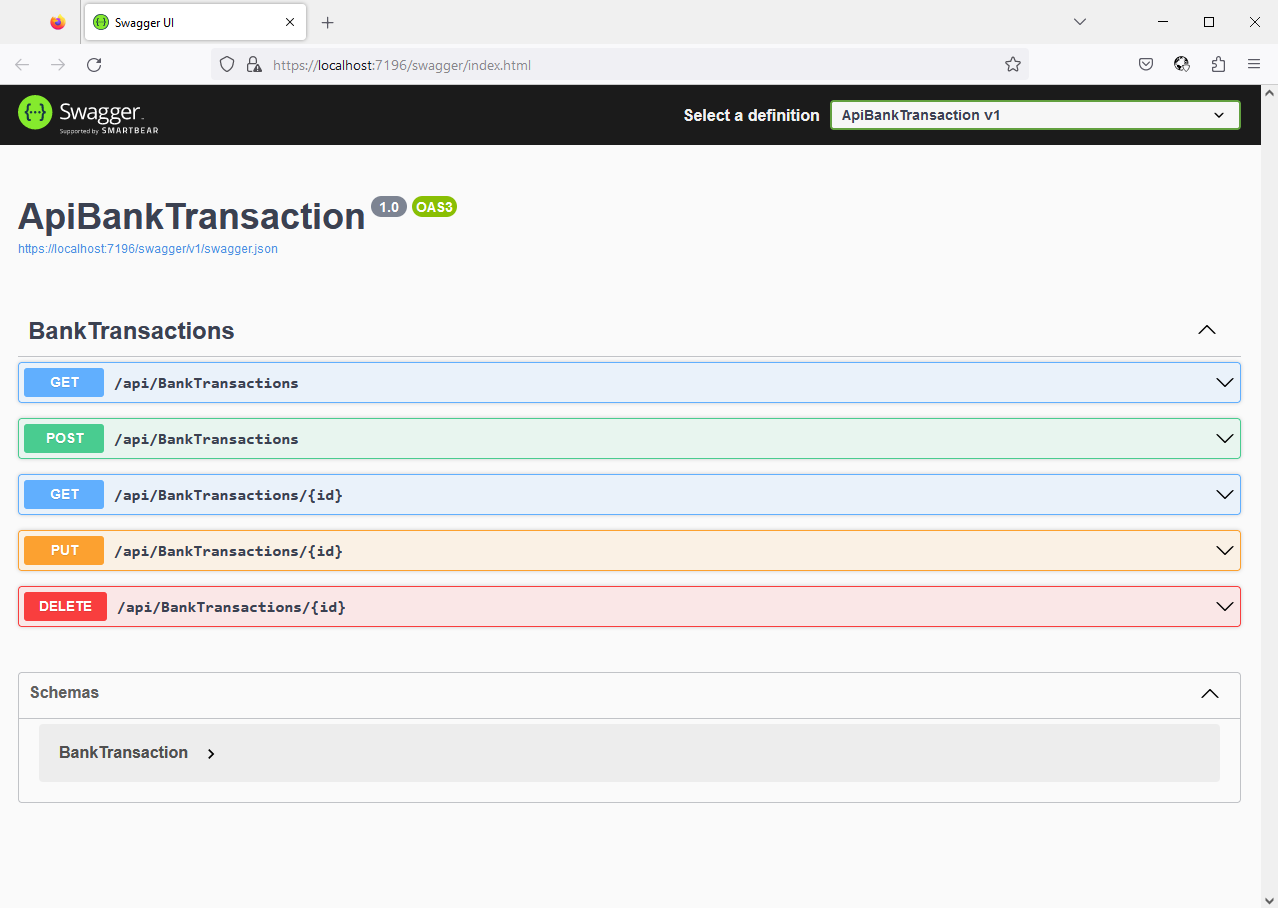
On valide et s’ouvre alors la page du contrôleur, si ce n’est pas le cas et que ça plante, vérifier que la solution compile bien (regénération du projet sans erreur), et que les packages NuGet sont tous biens installés.



On peut lancer le projet en s’assurant que ApiBankTransaction est bien défini comme projet de démarrage

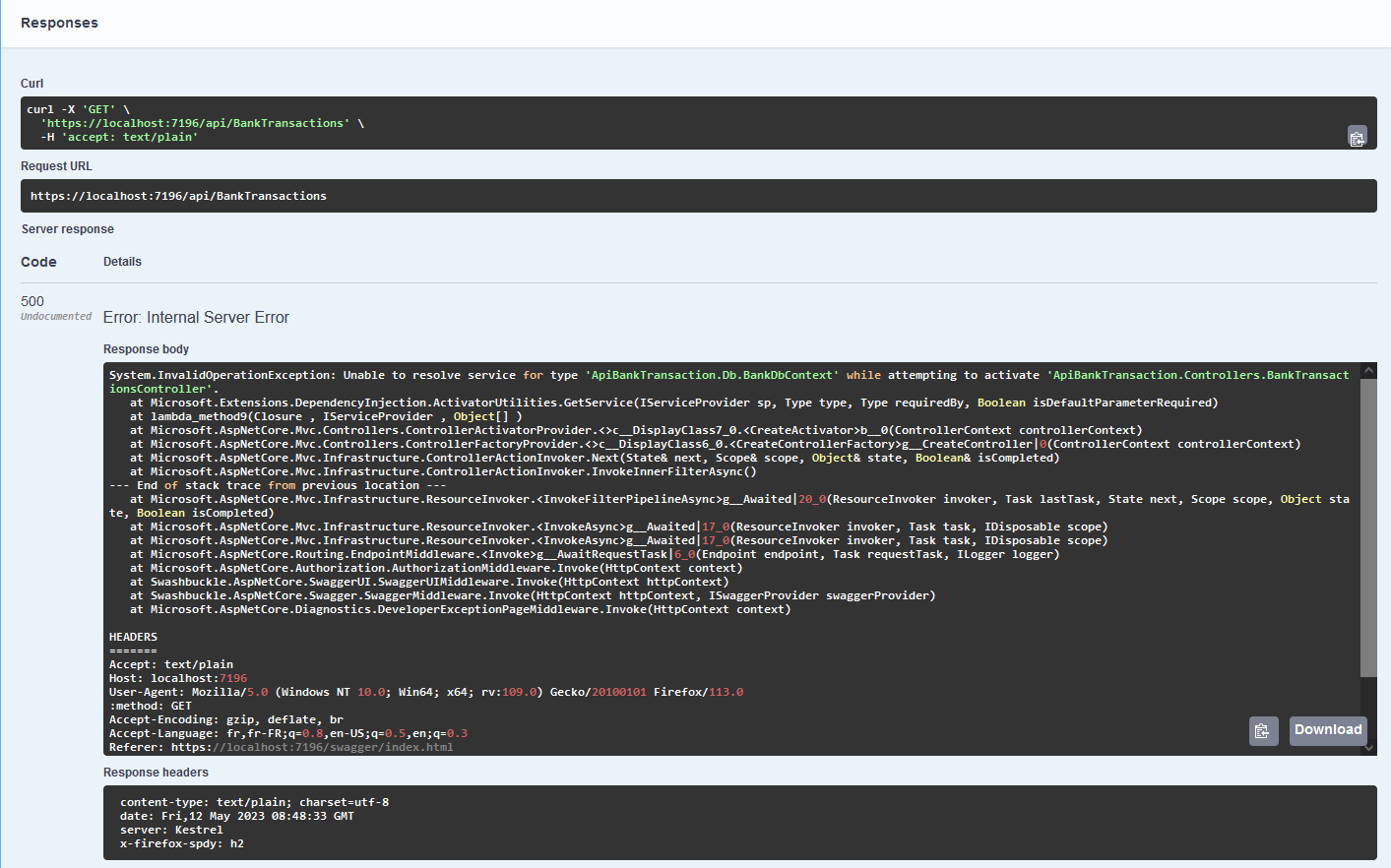


Lorsque l’on démarrage le projet Swagger s’ouvre

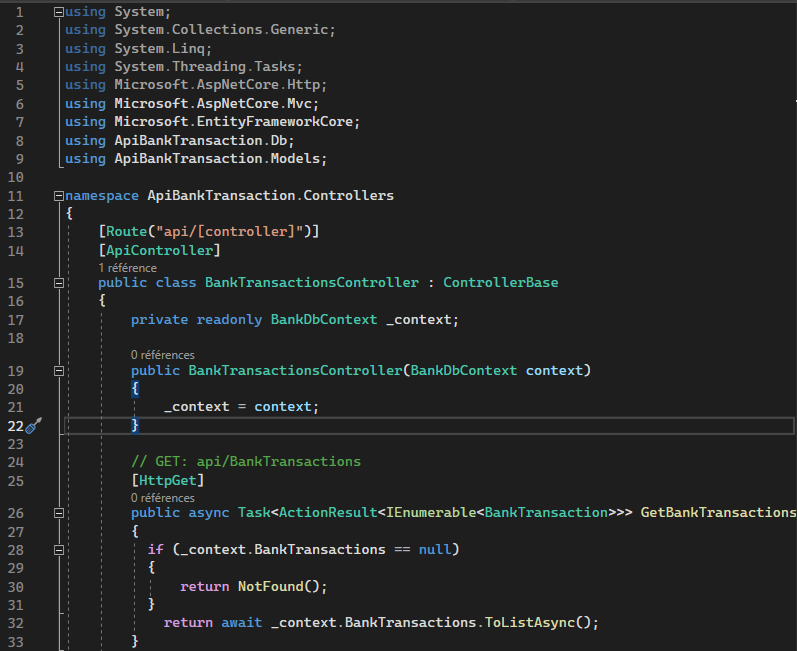


On teste alors la première méthode GET en cliquant ensuite sur Exécute

Apparaît alors le message d’erreur suivant :



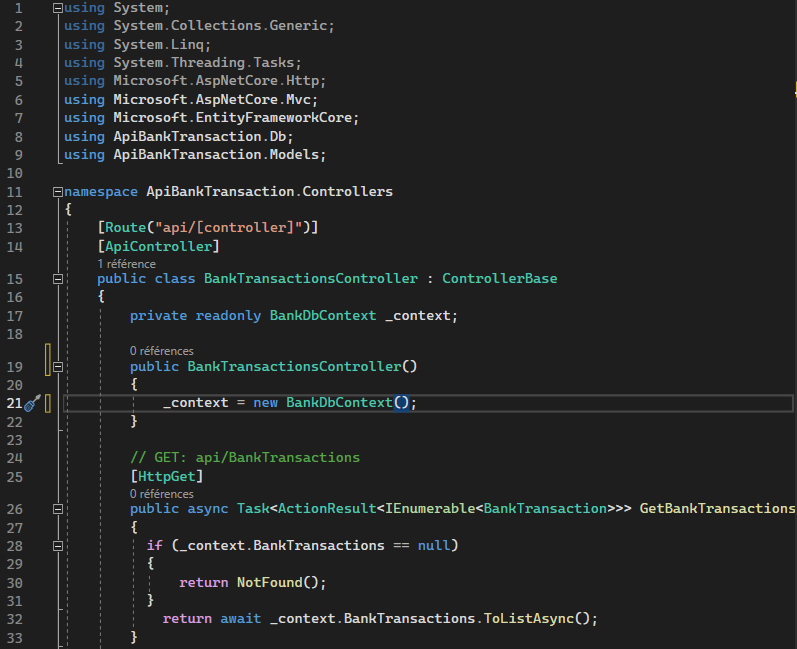
Il faut alors faire la modification suivante dans le controller :



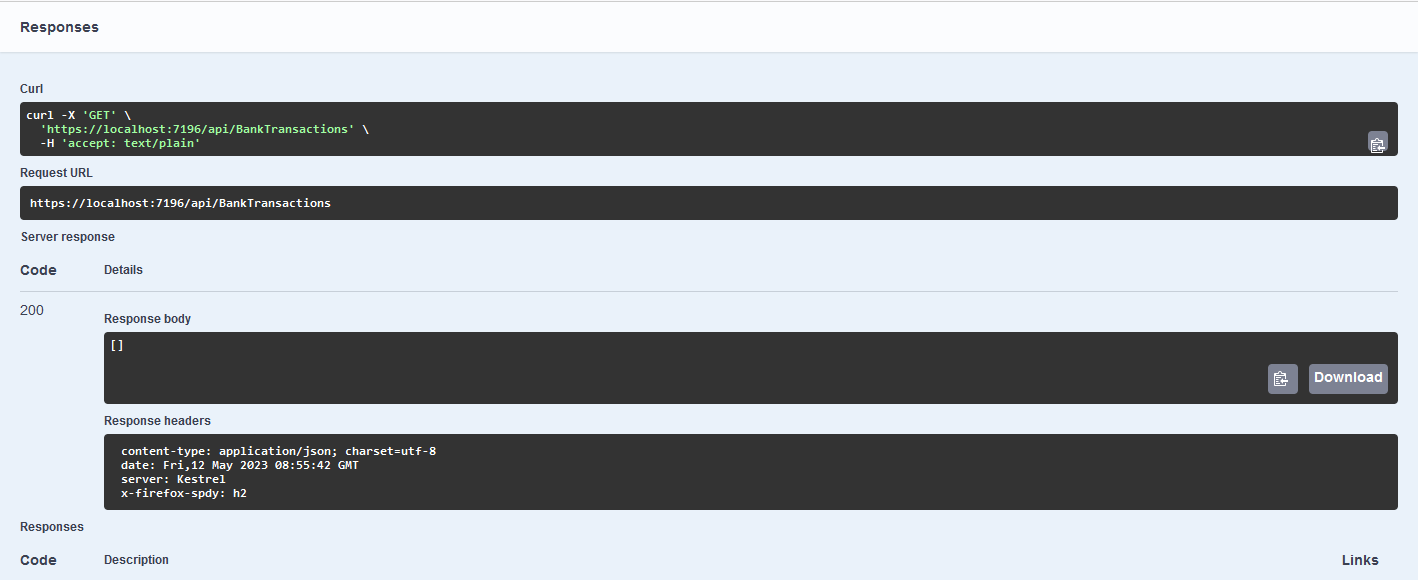
On supprime l’injection de dépendance « BankDbContext context » du constructeur « BankTransactionsController »

On instancie ensuite le « \_context » avec « \_context = new BankDbContext »

On a alors :



Quand on relance l’exécution du projet on obtient alors :



Renvoi un code 200 signifiant que tout s’est bien déroulé, même si, comme on peut le voir dans le « Response body », il n’y a encore aucun retour puisque que l’on n’a encore entré aucunes données.

On peut alors faire appel à la première méthode POST pour entrer nos premières données