Créer un projet API wep asp .net core

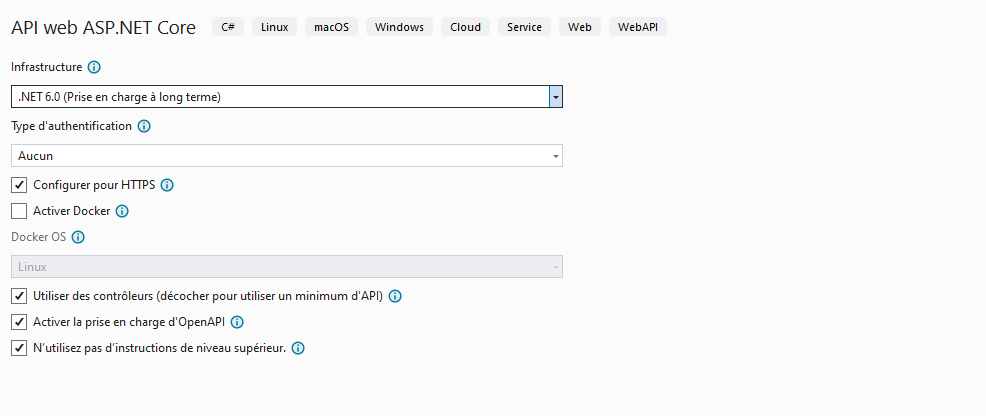
Installer les packet

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

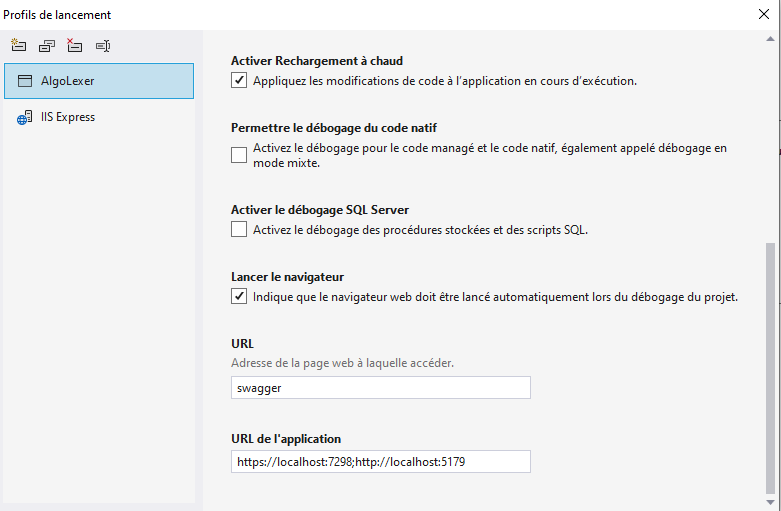


Une image contenant texte

Description générée automatiquement

A savoir

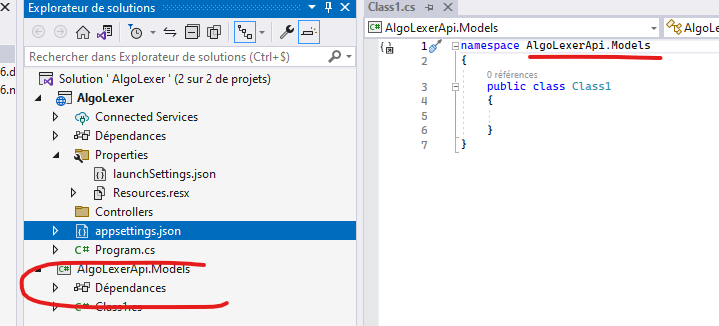
Si on veux changer le port de lancement (avec swagger)

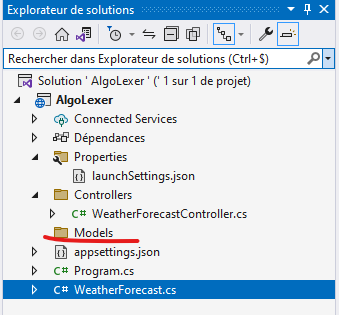


Une image contenant texte

Description générée automatiquement

Créer un dossier Models ou une bilbiotheque de classe Models ( si on crée une bibliotheque de classe la nommer avec le nom de l’application et ajouter . model ex : nomApplication.Models)





Crée dans la bibliotheque une classe Model

Pour en faire une classe générique

Ajouter l’attribut int ID et mettre un get et set  
On peut rendre la classe abstract

Une image contenant texte

Description générée automatiquement

Lors de la création d’une classe pour en faire un model  
On fera hérité cette classe de model

Une image contenant texte

Description générée automatiquement

Générer les annotations si besoin

Une image contenant texte

Description générée automatiquement

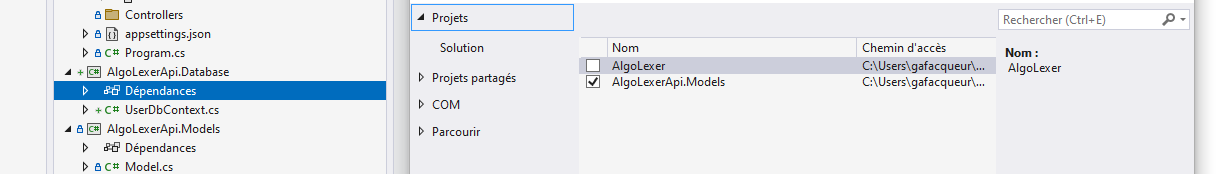
Créer un bibliotheque de classe pour la partie database

Et ajouter la classe UserDbContext

Une image contenant texte

Description générée automatiquement

Faire heriter la classe de db context (ne pas oublier de faire la dépendance de Model dans Database)



Une image contenant texte

Description générée automatiquement

Implémenter Onconfiguring

Une image contenant texte

Description générée automatiquement

Installer le packet nug get Entity pour tout les projet créer entre temps

Une image contenant texte

Description générée automatiquement

Entrer la chaine de connexion

Une image contenant texte

Description générée automatiquement

Puis fairre la migration depuis la console de package

Attention a bien sélectionner le bon projet

Puis faire la commande Update-Database

Une image contenant texte

Description générée automatiquement

On peut a present accéder a la base de donné depuis l’explorateur de serveur

Une image contenant texte

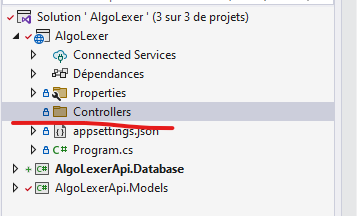
Description générée automatiquement

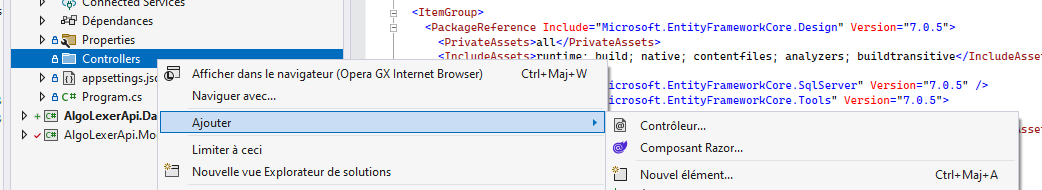
Ajouter les dépendance dans le projet

Une image contenant texte

Description générée automatiquement

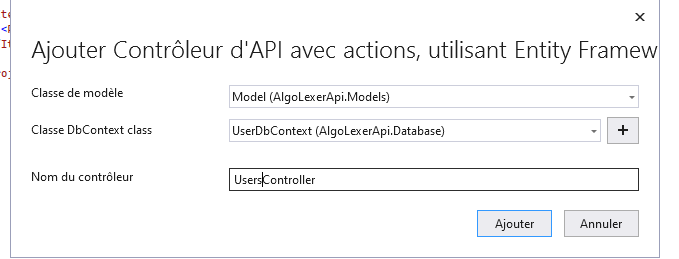
Creation d’un crontroller dans le dossier controller du projet





Une image contenant texte

Description générée automatiquement



Une image contenant texte

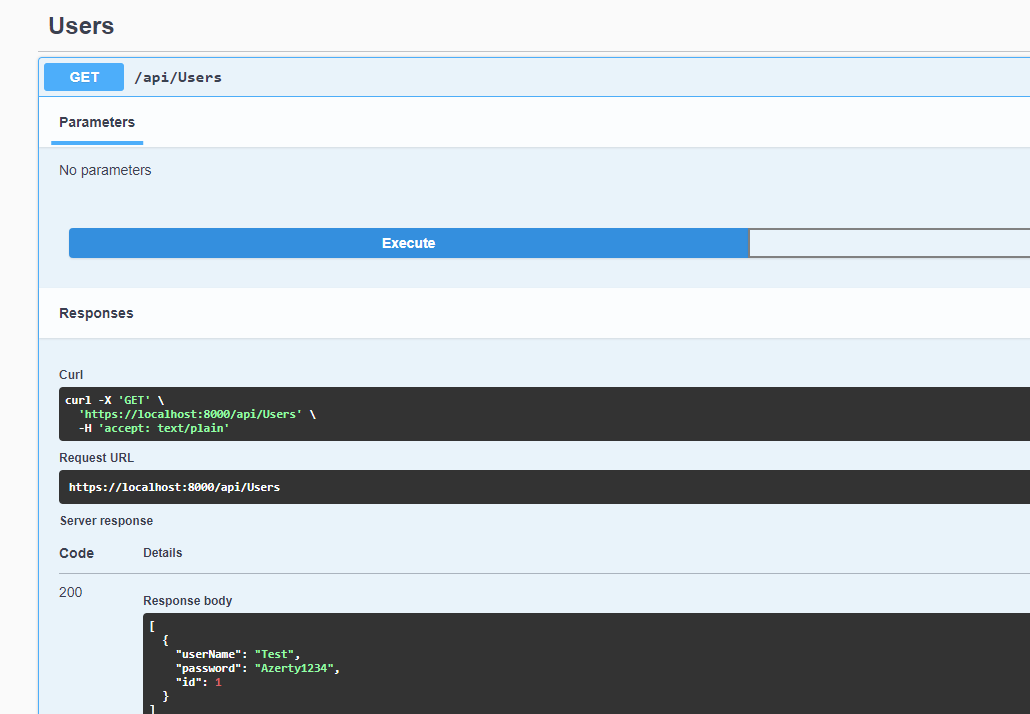
Description générée automatiquement

((Charger la base de donné dans le controlleur (bug a regler)))

Une image contenant texte

Description générée automatiquement

A ce stade Nous avons le controller et le model  
Il nous manque la vue, nous passons par Swagger UI



Design patern MVC

Une image contenant diagramme

Description générée automatiquement

----EXEMPLE DE Création d’un controlleur vide

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

Créer un constructeur et importer le dbcontext

Une image contenant texte

Description générée automatiquement

Creation de la methode d’authentification

Une image contenant texte

Description générée automatiquement

Teste sur PostMan

Une image contenant texte, capture d’écran, ordinateur

Description générée automatiquement

Chiffrement du mot de passe

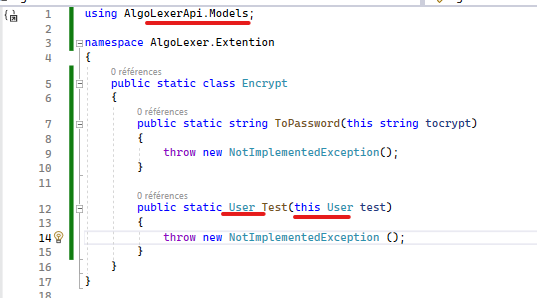
Création ‘une classe d’extention

Créer un dosier et une classe qu’on rendra static

Une image contenant texte

Description générée automatiquement

Exemple sur une classe personalisé



Installer Bcrypt.NET-Next (on peut l’installer que dans le projet ou la classe de chiffrement se situe)

Une image contenant texte

Description générée automatiquement

Methode de hash

Une image contenant texte

Description générée automatiquement

Methode qui verifie la chaine de connection

Une image contenant texte

Description générée automatiquement

On va l’utilisé dans le controlleur UserControl dans la methode PostUser

Une image contenant texte

Description générée automatiquement

A ce stade on peut ajouter un utilisateur et le mot de passe est transmi a la bdd de façon crypté

On implemente un viewModel pour pouvoir rendre les page visible a des utilisateur sans mot de passe

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

On peut mettre la Regex dans les ressource du projet

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Modifier ensuite le paramettre de PutUser dans UserController

Une image contenant texte, logiciel, Page web, Police

Description générée automatiquement

Separation des models et view models dans des dossiers

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Attention a bien réadapter le code si nécessaire

Une image contenant texte, ligne, Police, Tracé

Description générée automatiquement

Pour palier au probleme de l’afichage du mot de passe avec la methode Get  
On créer un autre view model sur le même principe que pour update

Une image contenant texte, logiciel, Police, capture d’écran

Description générée automatiquement

Puis modifier le verbe hhtpGet dans le userController