Ce site web est dédié aux administrateurs de la Plate-forme lait et les utilisateurs. Il permet aux administrateurs de gérer l'ouverture/suppression des comptes utilisateurs, validation ou refus des réservation, déclaration des opérations de maintenance, métrologie, entre autres. Pour plus d'informations sur le fonctionnement du site, référez-vous au manuel d'utilisation disponible dans le dossier « documents ».

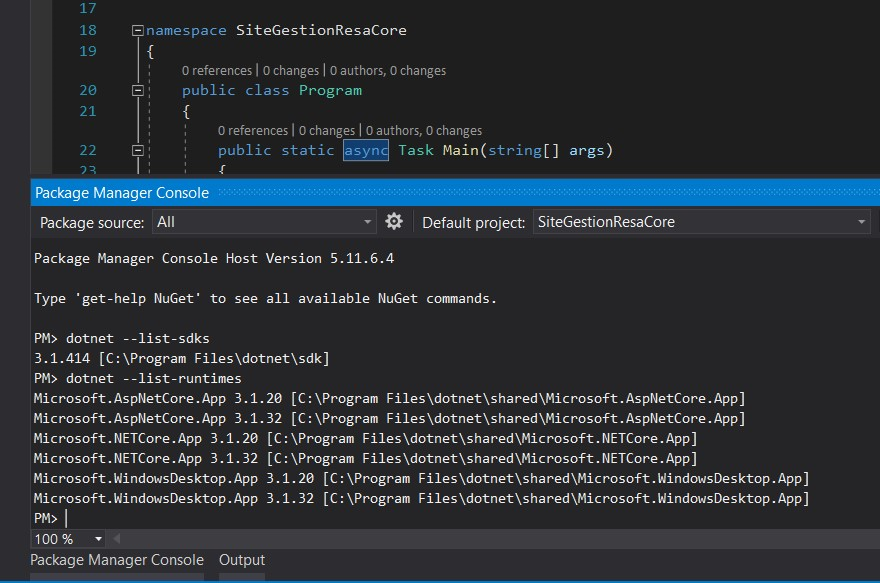
Vous allez devoir installer et configurer certains logiciels ou packages avant de pouvoir exploiter le logiciel, ci-dessous je vous donne le maximum d’informations. Cependant n’ayant jamais exploité ce logiciel ailleurs il se peut que vous retrouviez des problèmes. Dans ce cas, merci de me faire un retour du problème (anny.vivas@inrae.fr) pour que je puisse alimenter ce document.

**Installations pour le développement :**

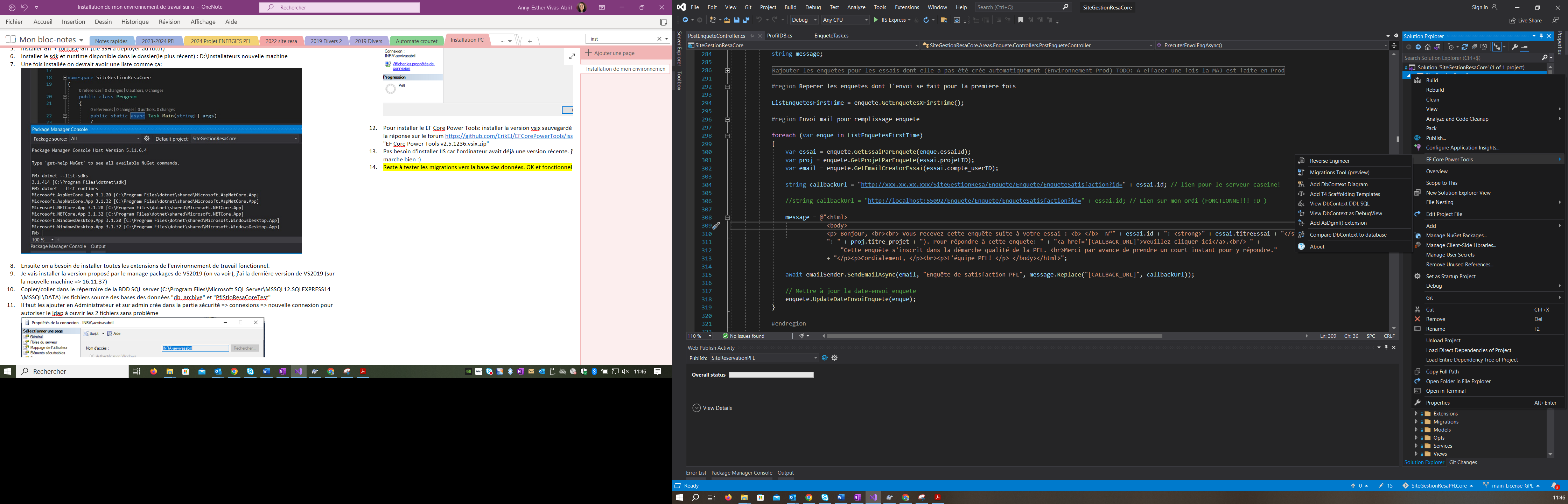
J’ai utilisé Visual studio 2019 pour la programmation du site. Ce logiciel sera à installer sur le poste pour pouvoir le faire fonctionner ou pour le modifier. <https://visualstudio.microsoft.com/fr/vs/older-downloads/>

Une fois VS installé alors vous allez devoir configurer votre environnement de travail. Pour faire marcher certaines fonctionnalités, il faudra installer certains packages ou NuGet. Voici les pas à suivre :

1. Installer tout d’abord le SDK (kit de développement) et le runtime ASP .NET CORE V3. Les fichiers d’installation sont disponibles dans ce lien : <https://dotnet.microsoft.com/fr-fr/download/dotnet/3.1>. Dans la liste montrée sur le site, sélectionnez la version 3.1.20, vous trouverez dedans les fichiers SDK 3.1414, Asp.net core 3.1.20, desktop .NET 3.1.20 et runtime .net 3.1.20. Une fois installés, vérifier que les packages sont bien installés (image ci-dessous). Taper : « dotnet - -list-runtimes » et « dotnet –list-runtimes » dans la package manager console.



1. Pour pouvoir gérer la base des données à partir de Visual studio, faire des migrations et du reverse engineering, il faudra installer l’extension EF Core Power Tools Version 12.5.1236 (disponible dans ce lien <https://github.com/ErikEJ/EFCorePowerTools/tree/archive/tools/VS2019Support> (renommer le fichier téléchargé en enlevant l’extension .zip) Cette extension vous permettra de faire des changements sur la BDD directement à partir du Visual studio et vous permettra aussi de faire l’inverse, c’est-à-dire, à partir d’une base des données, récupérer les classes pour pouvoir la gérer à partir de Visual studio (fonction Reverse Engineer). Point qui sera abordé plus loins dans le doc. Pour vérifier que EF core power tools et bien installé, clic droit sur le projet => EF core power tools



**2 base des données devront être instanciées sur SQL:**

- 1 base de données dédié à la sauvegarde des données provenant du site web (réservations des équipements, utilisateurs, etc.), elle sera créée à partir du code du site.

- 1 base de données qui dans le logiciel est nommé "dbArchive" mais qui peut être nommé différemment si nécessaire, c'est une base des données existante et qui devra être intégrée dans le site. Si vous n’avez pas de base des données pour la récupération des données, adaptez le code pour ignorer cette bdd.

SQL server management version 2014 devra être installé sur votre poste. Voici le lien d’installation pour télécharger les fichiers d’installation : <https://www.microsoft.com/fr-fr/download/details.aspx?id=42299>

Installation des logiciels SQL server et WebDeployIIS (logiciel qui permet d'exécuter l'application en tant que page web) :

1. Lancer l'installation du SQL server 2014, fichier nommé probablement "SQLEXPR\_x64\_FRA.exe". Installe le moteur de base de données.

2. Cliquer sur nouvelle installation autonome.

3. Sur sélection fonctionnalité : par défaut

4. Nom de l'instance SQLExpress14

5. Configuration du serveur : laisser par défaut. Mode d'authentification Windows.

6. Spécifier les administrateurs SQL Server : par défaut.

7. Une fois installée, lancer le 2émé fichier : « SQLManagementStudio\_x64\_FRA.exe ». Il installe l'IHM pour accès au moteur de BDD.

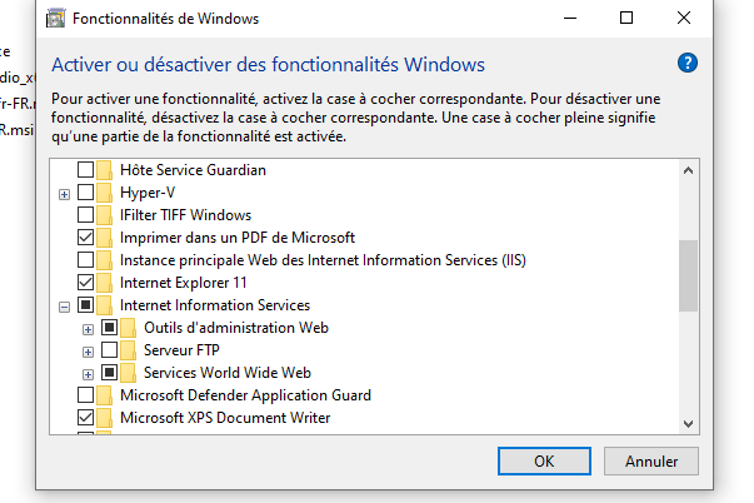
8. Sélectionner nouvelle installation autonome.

9. Type d’installation : cocher "Effectuer une nouvelle installation". Sélection fonctionnalités : par défaut.

10. localhost = nom de l'ordinateur. Privilégier le localhost pour standardiser les chaines de connexion.

Installation de WebDeploy IIS pour déploiement des sites web (disponible dans le dossier « WebDeploy »)

1. Installer tout par défaut. Ouvrir "Activer ou désactiver des fonctionnalités Windows" y activer "instance principale Web" y "Internet information Services"



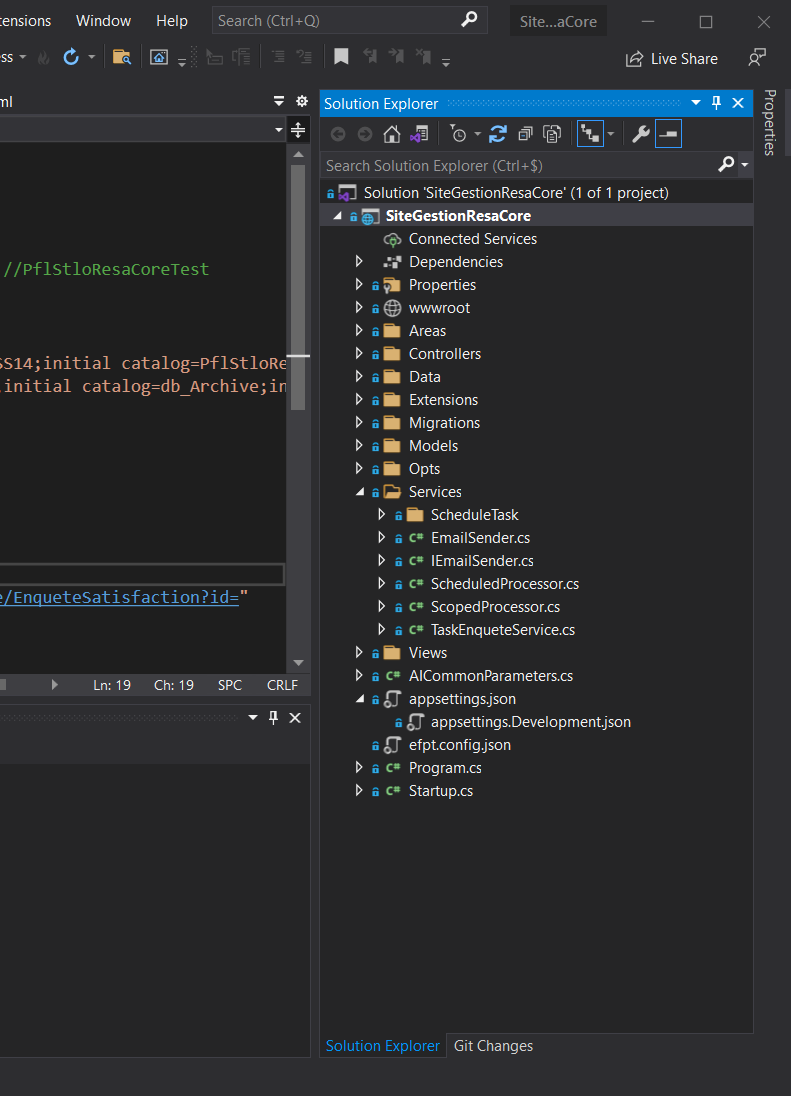
**Modifications à apporter dans le code avant de compiler :**

Une fois que vous avez installé SQL server et Visual studio, vous devrez configurer le code à votre environnement de travail, vous allez devoir modifier et faire quelques manipes avant de voir le site tourner sans erreur.

1. Configurations des fichiers. json :

Il s’agit des fichiers de configuration pour l’environnement de production (appsettings.json) et l’environnement de développement (appsettings.Development.json) qui devront être avec les informations pour votre environnement de travail.

Ces 2 fichiers sont disponibles dans le « solution explorer » sur Visual studio :



**Fichier « appsettings.Development.json » :**

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"ConnectionStrings": {

"SqlServer": "data source=localhost\\SQLEXPRESS14;initial catalog=PflStloResaCoreTest;integrated security=True;MultipleActiveResultSets=True",

"PcVue": "data source=localhost\\SQLEXPRESS14;initial catalog=db\_Archive;integrated security=True;MultipleActiveResultSets=True"

},

"MainAdmin": {

"email": "mail",

"mdp": "......",

"nom": "....",

"prenom": "...."

},

"LinkEnquete": {

"url": "http://localhost:55092/Enquete/Enquete/EnqueteSatisfaction?id="

}

}

PflStloResaCoreTest => Nom de la base des données test pour debug en local sur votre machine. A Changer pour l’adapter au nom de votre base des données

db\_Archive => Nom de la base des données où sont stockés les données provenant du système de supervision de la PFL. Cette base des données est gérée par PcVue (logiciel d’acquisition des données pour les équipements plate-forme). A changer et adapter selon le nom de votre base des données si vous en avez un système d’acquisition des données et sinon adapter le code pour ignorer cette partie.

MainAdmin => Saisir les informations de connexion au site web pour l’administrateur principal.

LinkEnquete => lien vers une enquête de satisfaction qui sera généré à la fin de chaque essai. Pas besoin de le modifier puisque le mot clé localhost prendra le nom du poste où vous travaillerez en développement.

**Fichier « appsettings.json » :**

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"AllowedHosts": "\*",

"Email": {

"From": ".......@inrae.fr",

"SMTP": "smtp.inrae.fr",

"Username": ".......@inrae.fr",

"Password": ""

},

"ConnectionStrings": {

"SqlServer": "data source=localhost\\SQLEXPRESS14;initial catalog=PflStloResaCore;integrated security=True;MultipleActiveResultSets=True",

"PcVue": "data source=localhost\\SQLEXPRESS14;initial catalog=db\_Archive;integrated security=True;MultipleActiveResultSets=True"

},

"MainAdmin": {

"email": "A completer",

"mdp": "A completer",

"nom": "A completer",

"prenom": "A completer"

},

"LinkEnquete": {

"url": "http://xxx.xx.xxx.xxx/Enquete/Enquete/EnqueteSatisfaction?id="

}

}

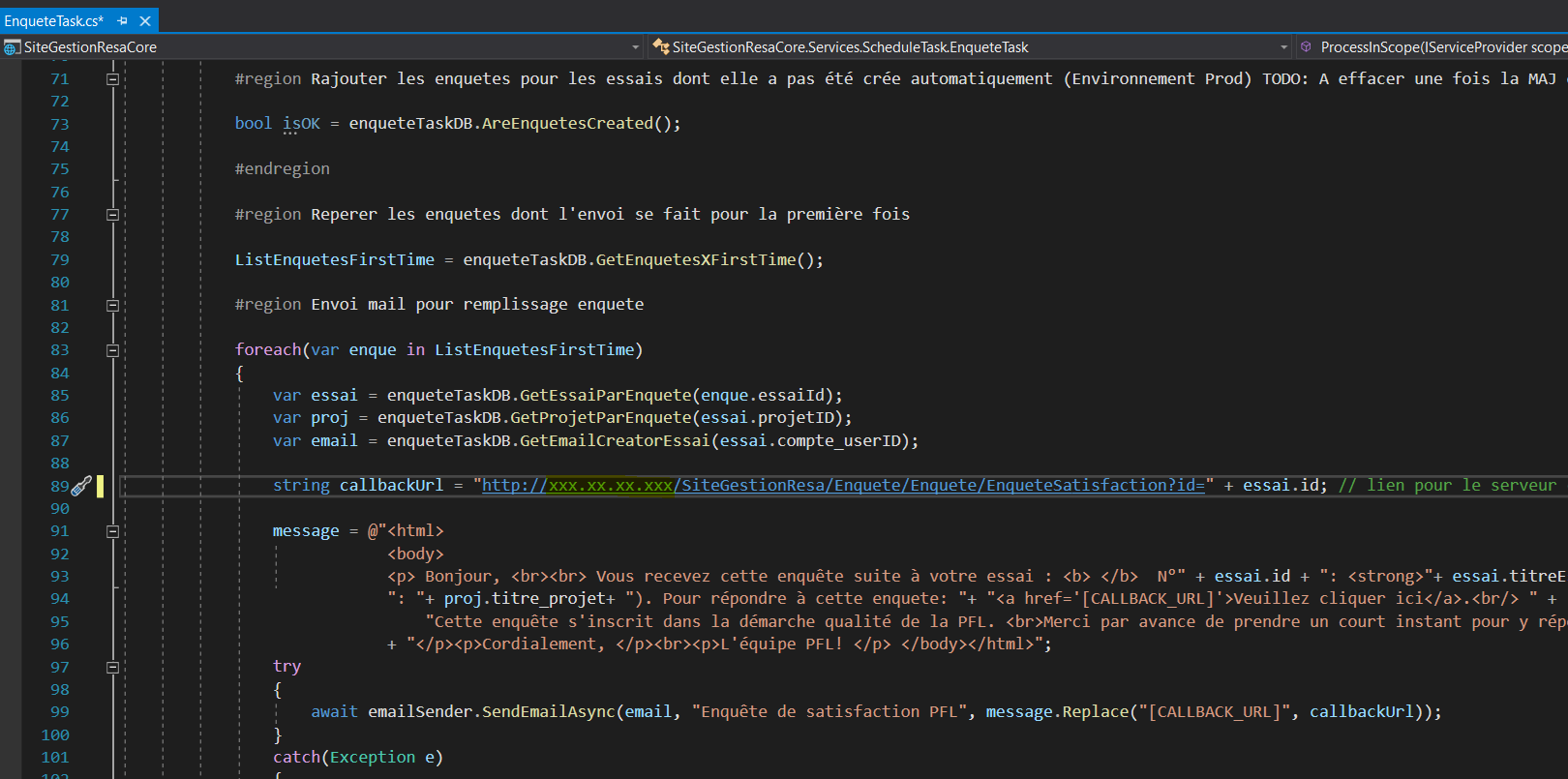
"AllowedHosts" => Saisir les informations de votre adresse mail applicative, cette adresse mail est à demander à votre gestionnaire de SI. Elle sera utilisée pour l’envoi des mails automatiques de notification.

"MainAdmin" => idem que sur le appsettings.developpement.json

"LinkEnquete" => ce lien sera envoyé à partir du serveur de déploiement, saisir l’adresse IP où vous allez déployer l’application.

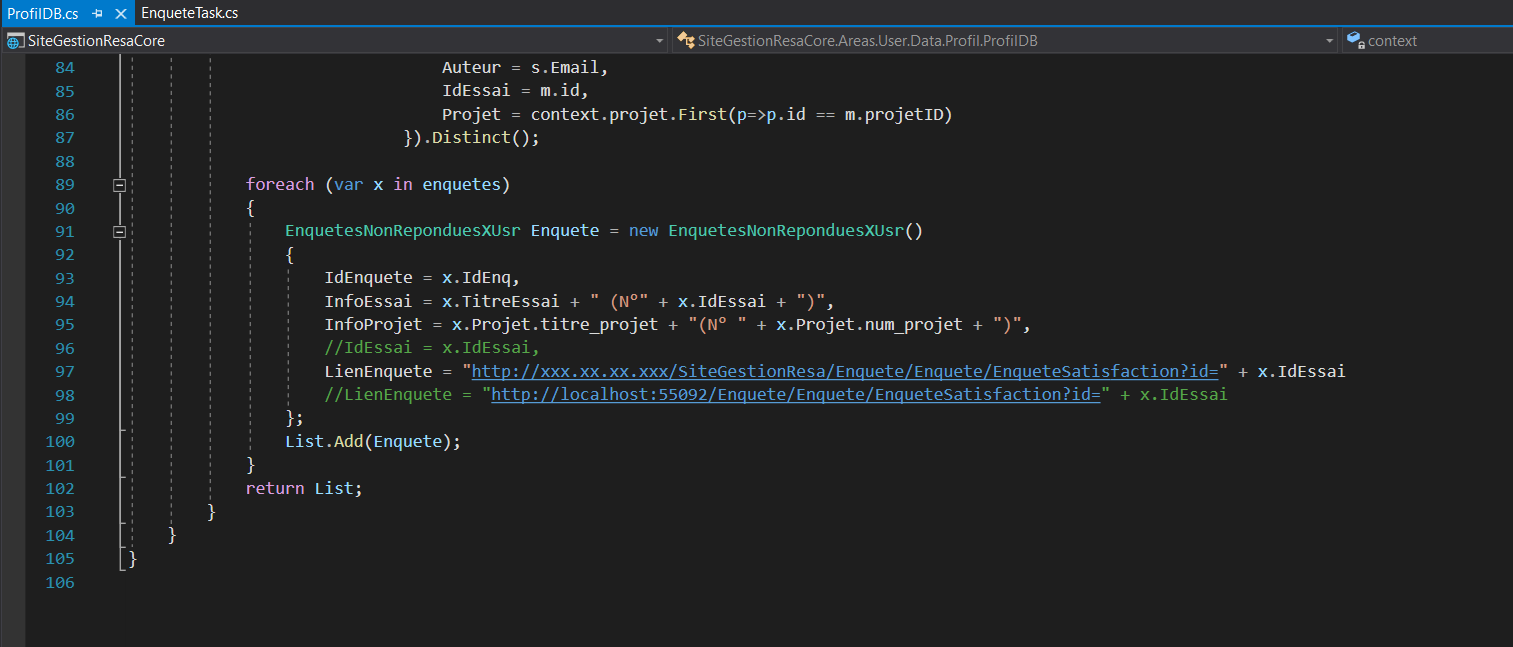
1. Fichier « EnqueteTask.cs » :

Sur ce fichier, saisissez l’adresse IP du serveur où l’application sera déployée. Ligne 89 et 127.



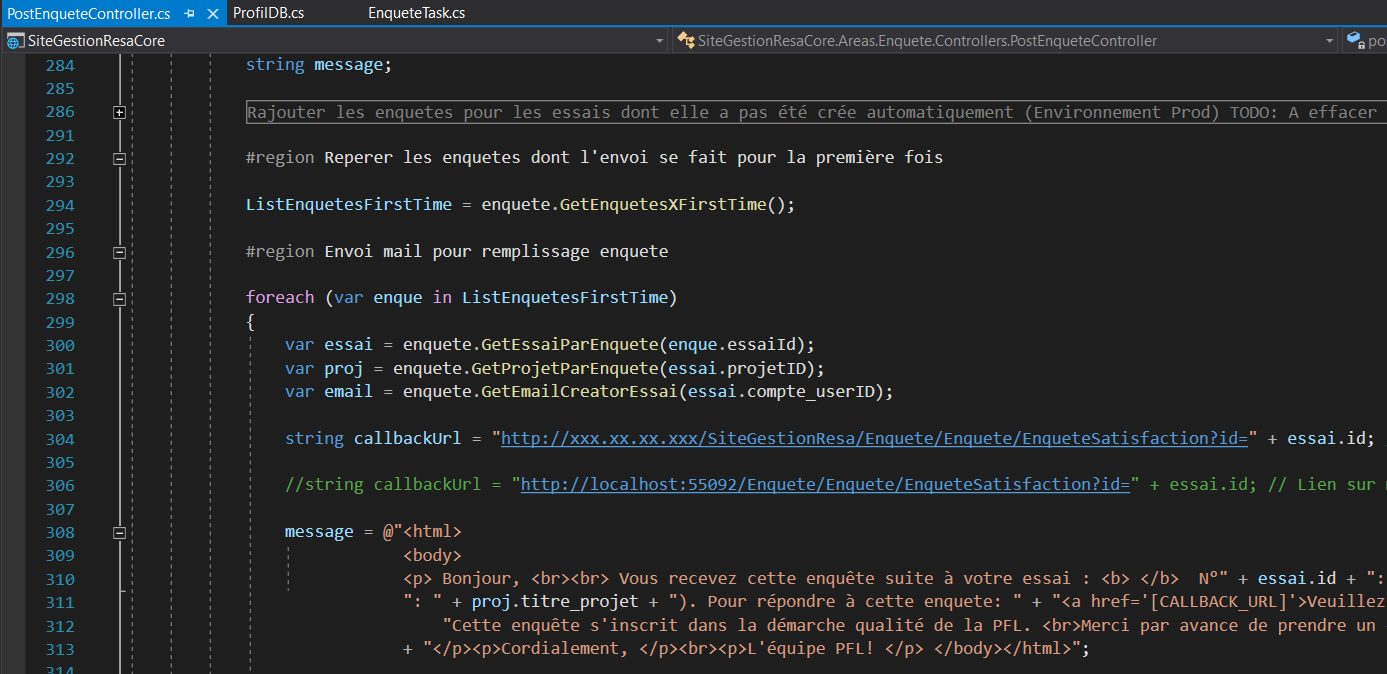
1. Fichier « ProfilDB.cs » :

Sur ce fichier, saisissez l’adresse IP du serveur où l’application sera déployée. Ligne 97.

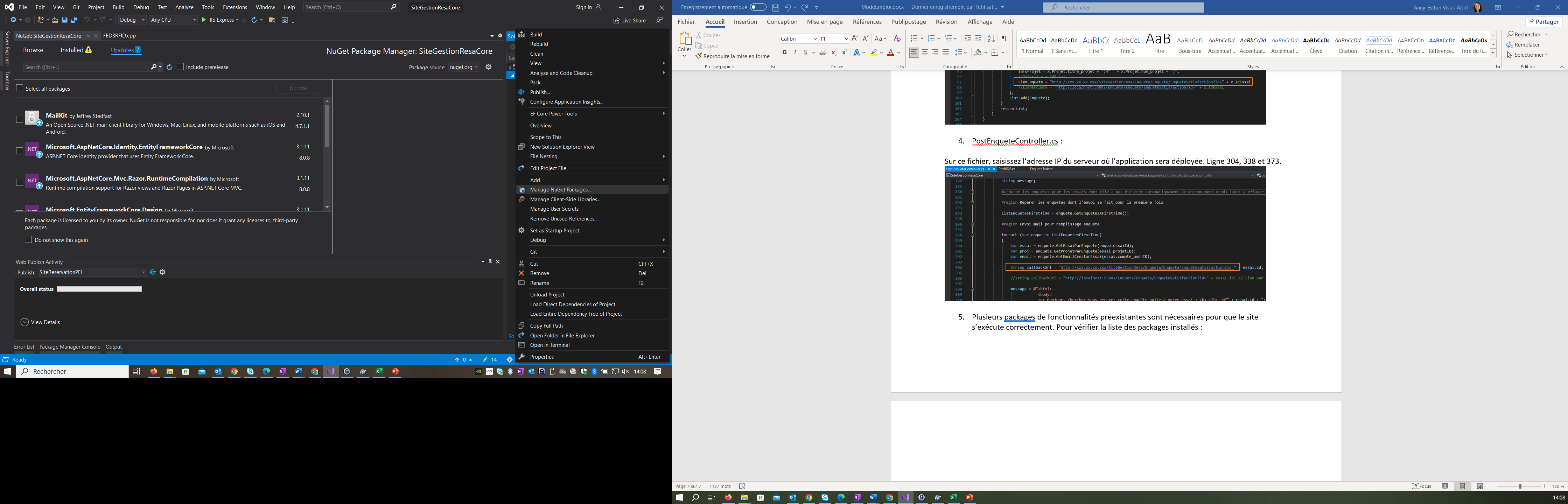


1. PostEnqueteController.cs :

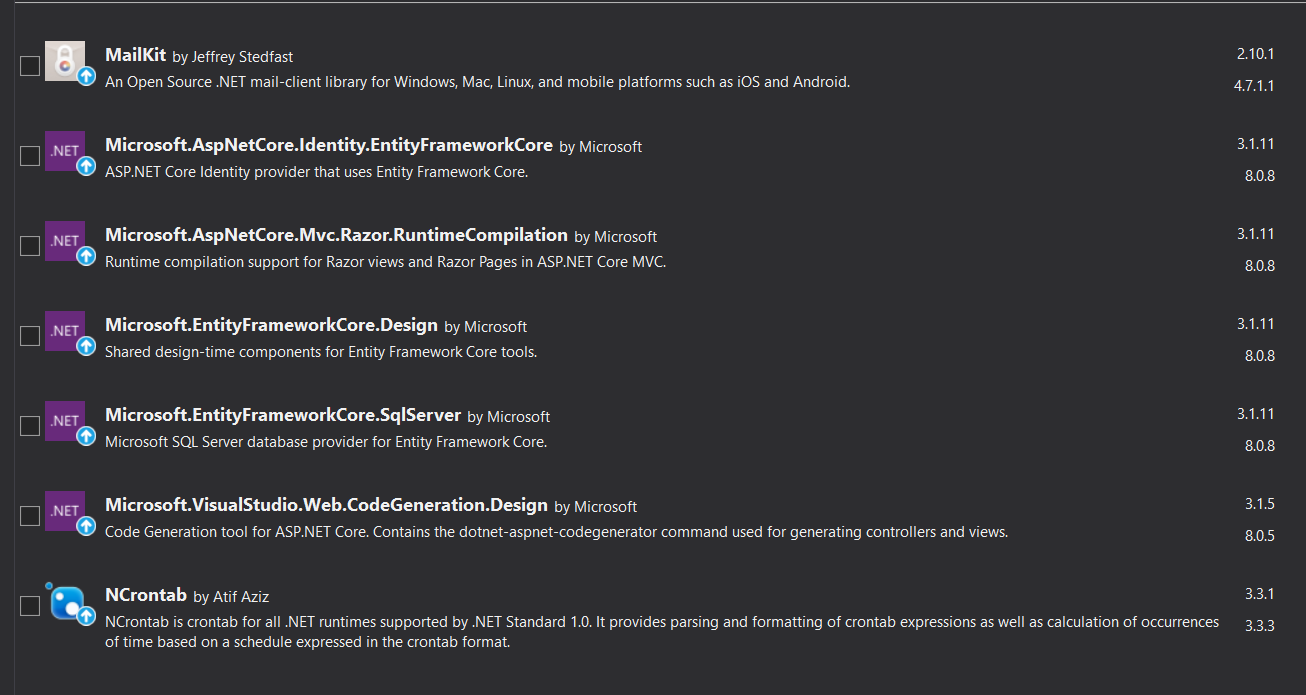
Sur ce fichier, saisissez l’adresse IP du serveur où l’application sera déployée. Ligne 304, 338 et 373.



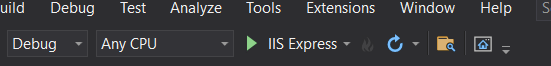
1. Plusieurs packages de fonctionnalités préexistantes sont nécessaires pour que le site s’exécute correctement. Pour vérifier la liste des packages installés :



Clic droit sur le projet « SiteGestionResaCore » et cliquer sur « Manage NuGet Packages ». Vous devriez avoir la liste des packages ci-dessous : attention, restez sur les versions proposées qui sont compatibles avec la version du .Net Core 3



**« Nettoyage » du code avant de compiler pour la toute première fois :**

Avant de lancer le debug sur IISExpress 

Vous allez devoir inclure les données que vous voulez écrire au démarrage de votre site. Les données qui sont mis en dur dans le code permettent de démarrer le site avec les informations importantes au démarrage et au fur et mesure que vous allez vous approprier du site vous alimenterez avec des autres données.

**Remplacer les données injectées par défaut sur la BDD :**

Si vous voulez mieux comprendre la structure de la BDD, clic droit sur le projet « SiteGestionResaCore » => EF Core Power Tools => Add DbContext Diagram, l’extension vous génere un diagramme de classes de la base de données que vous permettra de comprendre la rélation des differentes classes.

Dans le fichier GestionResaContext :

Certaines données sont injectées dans la base des données directement dans le code pour faciliter la tâche. Dans votre cas, vous allez devoir rajouter vos propres données pour préremplir votre BDD. Prenez le temps de bien comprendre ce fichier avant de faire une migration.

**Fonctionnalité EF Core Power Tools : Reverse Engineer**

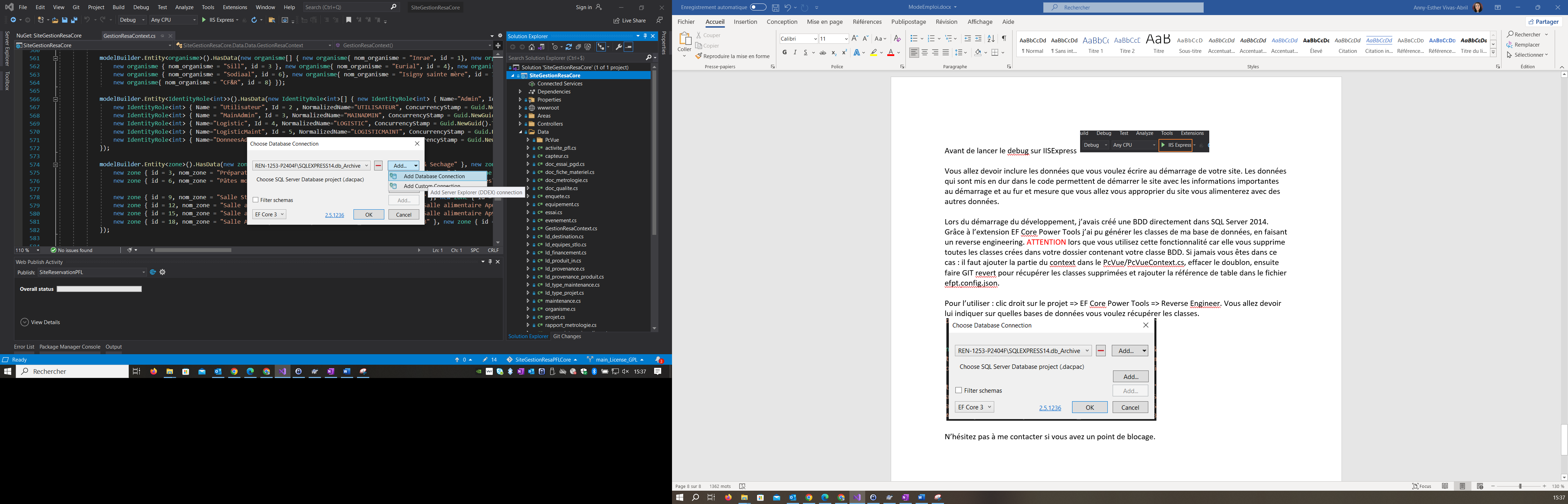
Cette fonctionnalité permet de créer toutes les classes nécessaires pour accéder/modifier les tables des données directement dans notre code à partir d’une base des données géré par un autre logiciel ou crée avant le développement.

Lors du démarrage du développement, j’avais créé une BDD directement dans SQL Server 2014. Grâce à l’extension EF Core Power Tools j’ai pu générer les classes de ma base de données, en faisant un reverse engineer. ATTENTION : lors que vous utilisez cette fonctionnalité car elle vous supprime toutes les classes crées dans votre dossier contenant votre classe BDD. Si jamais vous êtes dans ce cas : il faut ajouter la partie du context dans le PcVue/PcVueContext.cs, effacer le doublon, ensuite faire GIT revert pour récupérer les classes supprimées et rajouter la référence de table dans le fichier efpt.config.json.

J’utilise cette fonctionnalité très rarement, quand j’ai besoin de récupérer une nouvelle table des données sur ma BDD d’acquisition des données qui est géré par un autre logiciel. Récemment j’ai essayé de coder directement dans le code les tables des données sans passer par le reverse engineer et tout se passe bien.

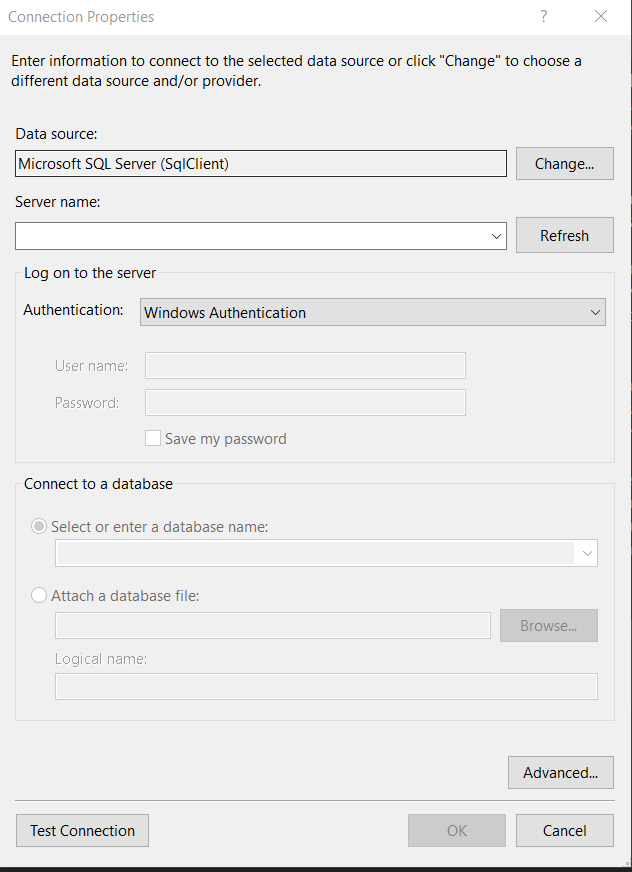
Cependant, je vous explique comme l’utiliser :

Clic droit sur le projet => EF Core Power Tools => Reverse Engineer. Vous allez devoir lui indiquer sur quelles bases de données vous voulez récupérer les classes. Add => Add Database Connection



Vous allez pouvoir récupérer votre nom du serveur SQL server 2014. Pour l’obtenir, lancez SQL et copiez le nom du serveur

Dans la fenêtre « Connection properties » :



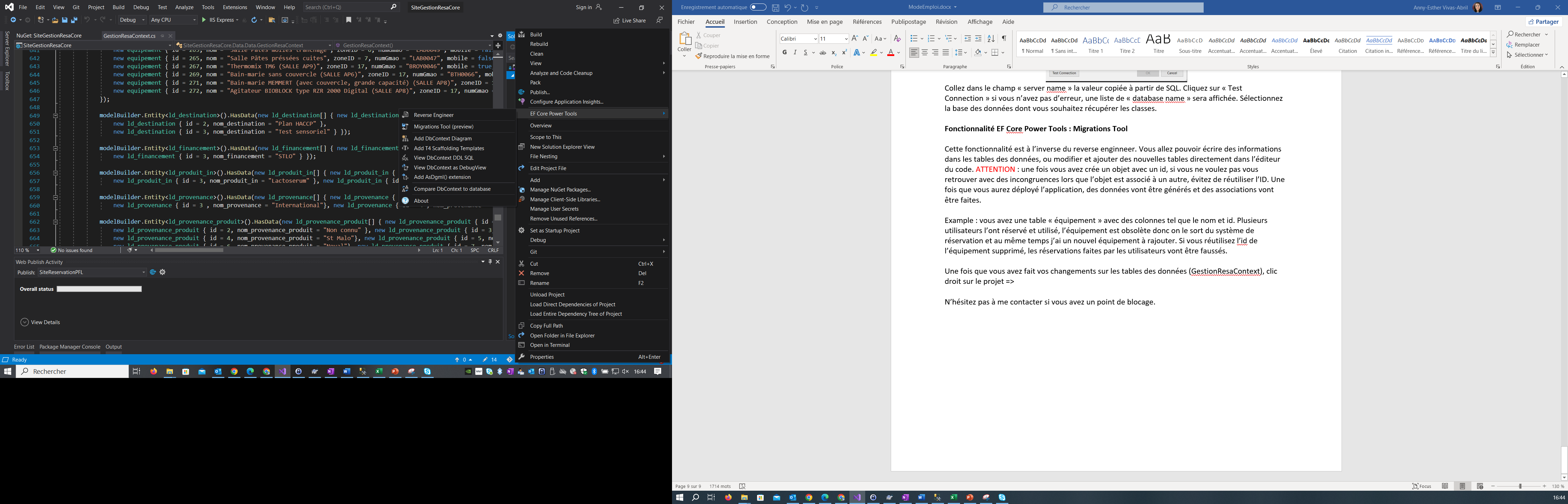
Collez dans le champ « server name » la valeur copiée à partir de SQL. Cliquez sur « Test Connection » si vous n’avez pas d’erreur, une liste de « database name » sera affichée. Sélectionnez la base des données dont vous souhaitez récupérer les classes.

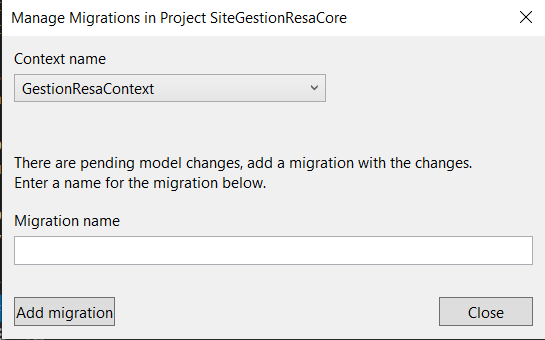
**Fonctionnalité EF Core Power Tools : Migrations Tool**

Cette fonctionnalité est à l’inverse du reverse enginneer. Vous allez pouvoir écrire des informations dans les tables des données, ou modifier et ajouter des nouvelles tables directement dans l’éditeur du code. ATTENTION : une fois vous avez crée un objet avec un id, si vous ne voulez pas vous retrouver avec des incongruences lors que l’objet est associé à un autre, évitez de réutiliser l’ID. Une fois que vous aurez déployé l’application, des données vont être générés et des associations vont être faites.

Example : vous avez une table « équipement » avec des colonnes tel que le nom et id. Plusieurs utilisateurs l’ont réservé et utilisé, l’équipement est obsolète donc on le sort du système de réservation et au même temps j’ai un nouvel équipement à rajouter. Si vous réutilisez l’id de l’équipement supprimé, l’historique des réservations faites par les utilisateurs va être faussée.

Une fois que vous avez fait vos changements sur les tables des données (GestionResaContext), clic droit sur le projet => EF Core Power Tools => Migrations Tool





Donnez un nom à votre migration et cliquez sur « Add Migration », ceci peut prendre quelques secondes selon la migration faite.

Un fichier contenant les changements sur la base des données est créé et rajouté dans un dossier nommé « Migrations ». Pour que la migration soit chargée sur la BDD, lancez le debug.

Pour mieux comprendre le fonctionnement du site avant de vous lancer sur l’installation et prise en main, vous trouverez dans le dossier « Documents » le manuel d’utilisation du site. N’hésitez pas à me contacter si vous avez un point de blocage.