Azure Unit Tests - VS 2019

I. L'objectif de cet atelier est de créer avec VS un projet web forme qui utilise Entity Framework comme ORM pour effectuer des opérations sur la BD, et en parallèle nous avons créé un projet de type Unit Tests pour pouvoir lancer des tests à partir de Visual Studio.

Les prérequis pour la réalisation de cet atelier :

- Visual Studio version 2019.
- SQL Server.
- Des connaissances Entity FrameWork.

1. Lancez Visual Studio.

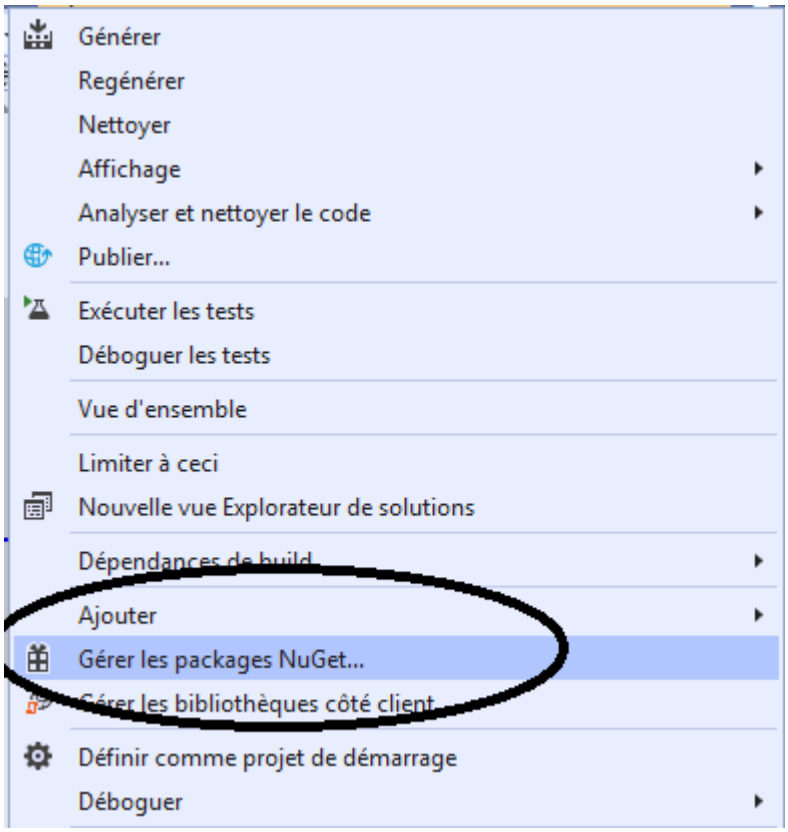
2. Créez un projet vide de type WebForme.

3. Puis cochez la case, Créer un projet Unit Test, soit au moment de création du projet, soit en rajoutant un nouveau projet après la création de la solution.

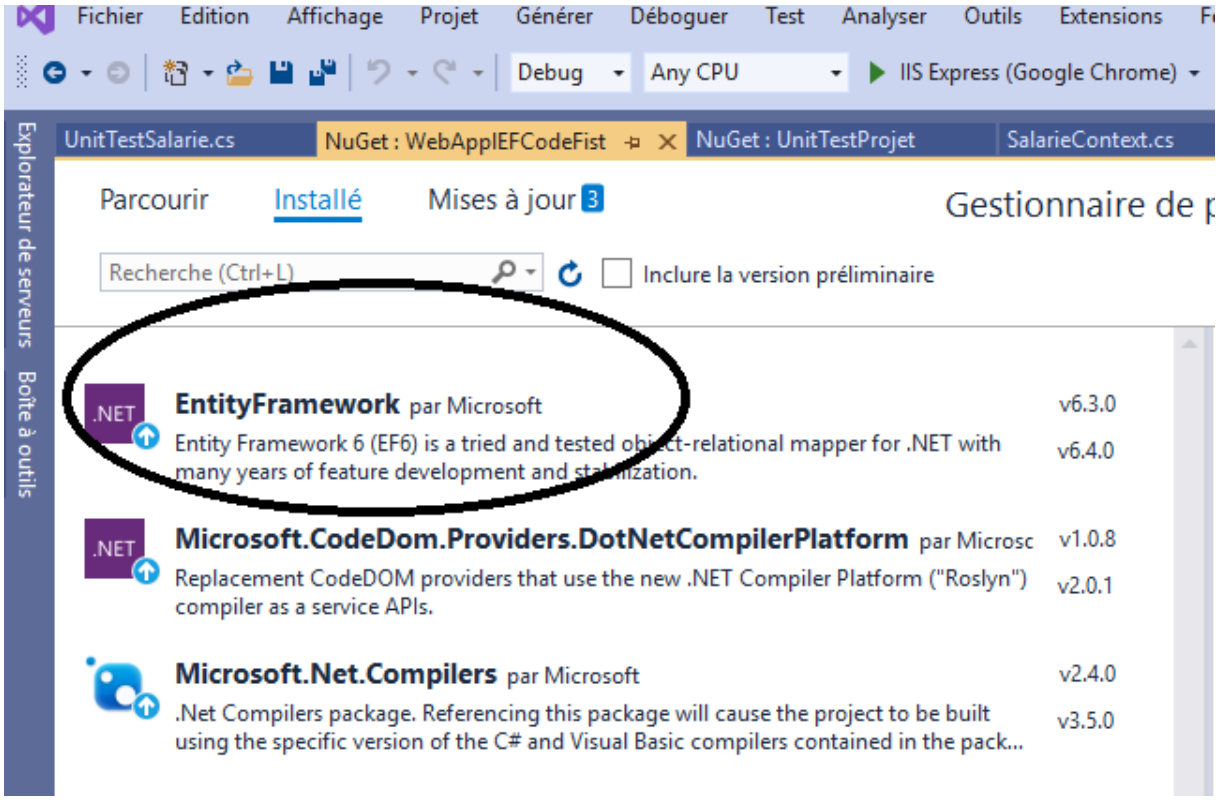
4. Une fois votre projet est créé, nous allons commencer par la création de deux classes, Salarie et Departement, ci-dessous le code :

```
public class Salarie
{
    public Int32 SalarieId { get; set; }
    public String Nom { get; set; }
    public String Prenom { get; set; }
    public String Fonctione { get; set; }
    public Int32 DepartementId { get; set; }

 }


public class Departement
{
    public Int32 DepartementId { get; set; }
    public String Description { get; set; }

    // Add Ville
    public String Ville { get; set; }

    public IList<Salarie> ListeSalarie { get; set; }
}
```

5. Avant de créer la classe DBContext, on va procéder au téléchargement d'Entity FrameWork à partir du gestionnaire du package Nuget de Visual Studio, cliquez bouton droit sur votre projet.



Puis :



Puis, cliquez sur installé.

6. Par la suite on va procéder à la création de la classe DBContext, la Classe DBContext doit contenir la chaine de connexion vers la DB (dans ce cas, SQLExpress).

```
public class SalarieContext : DbContext
{
       public SalarieContext():base("Data Source=.\\SQLExpress;Initial
       Catalog=MaBaseSalarie;Integrated Security=True")
         {

         }

         public DbSet<Salarie> Salaries { get; set; }
         public DbSet<Departement> Departements { get; set; }

}
```

NB : N'oubliez pas d'importer le NameSpace : using System.Data.Entity;

7. Après la création des deux classes, Salarie et Departement et la création de la classe DBcontext, nous allons créer une nouvelle classe SalarieCRUD avec les méthodes d'Ajout, de Suppression et de Recherche d'un salarie.

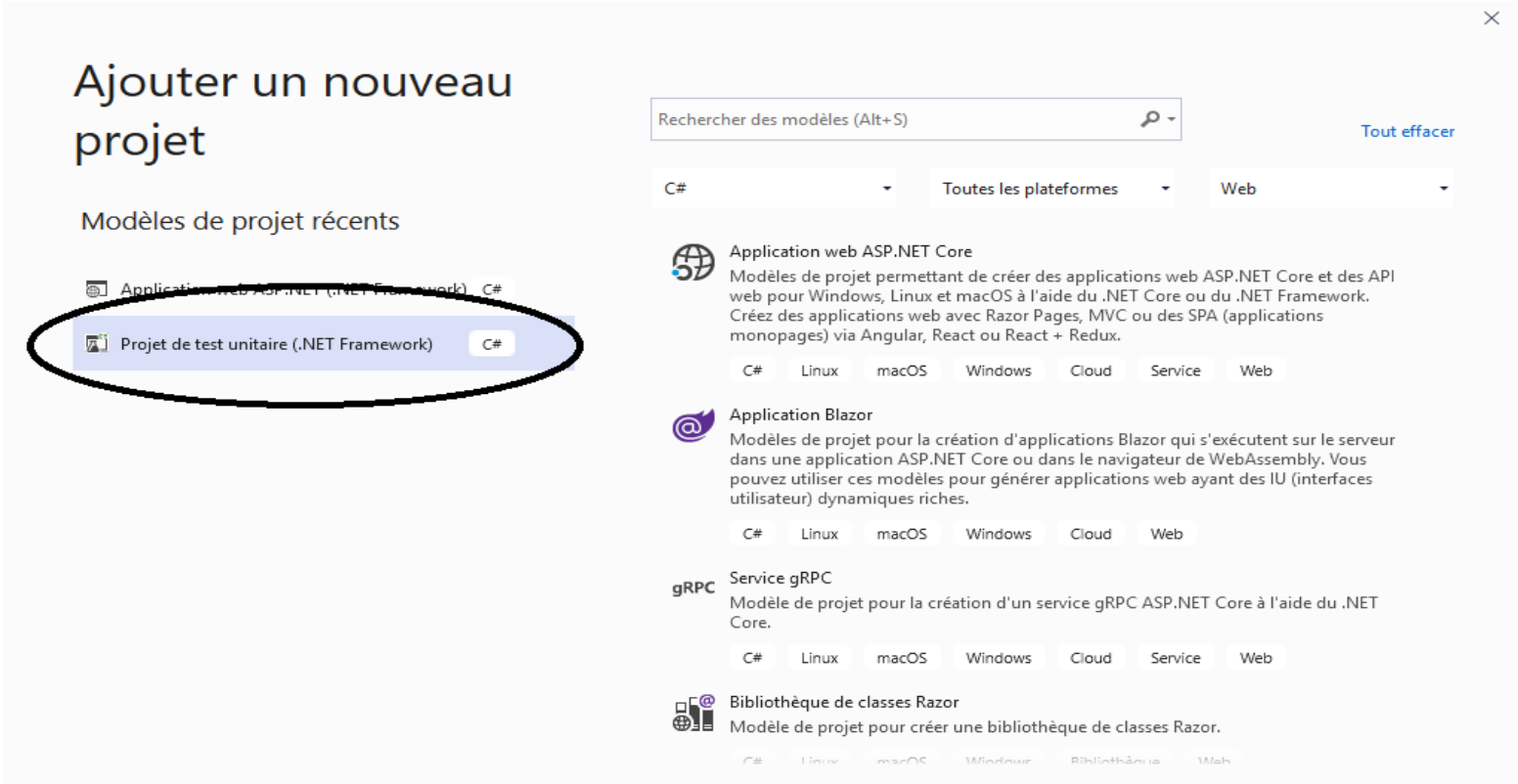
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplEFCodeFist.Models
{
    public class SalarieCRUD
    {

        SalarieContext cont = new SalarieContext();
        public void AddSalarie(Salarie Sal)
        {

            cont.Salaries.Add(Sal);
            cont.SaveChanges();
        }

        public void DelSalarie(Salarie Sal)
        {

            cont.Salaries.Remove(Sal);
            cont.SaveChanges();
        }
```

```
        public Salarie GetSalarie(String Prenom)
        {
            Salarie Sal = new Salarie();

            Sal = cont.Salaries.Where(s => s.Prenom ==
Prenom).FirstOrDefault();
            return Sal;
        }
    }
}
```

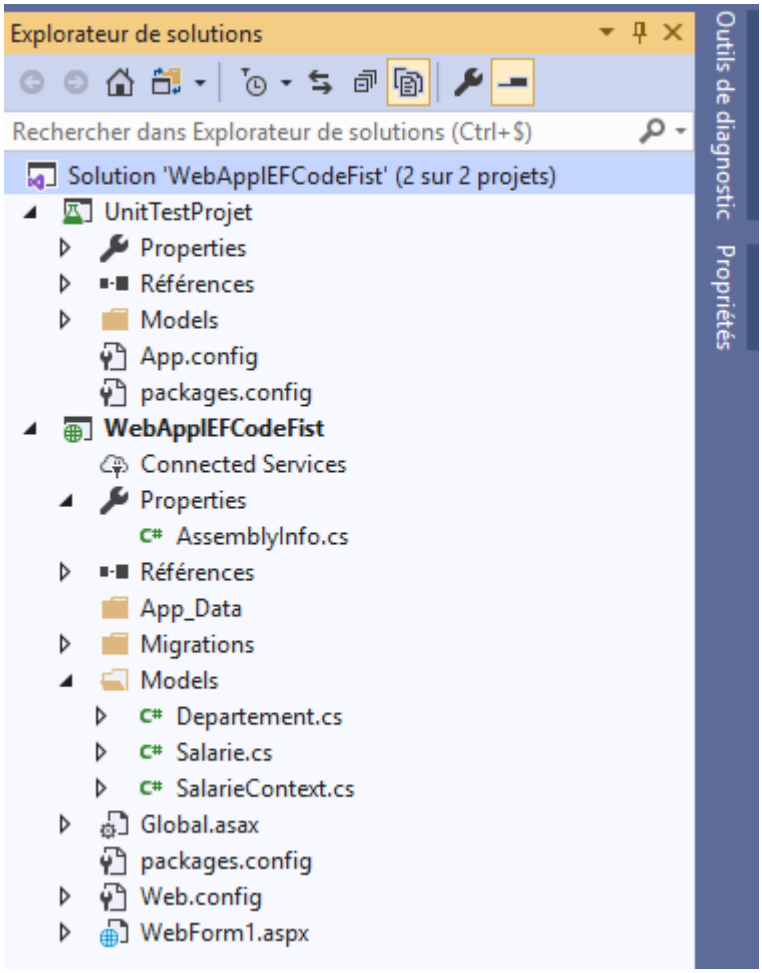
8. L'approche code first d'Entity Framework va créer les tables au niveau de la base de données d'une façon automatique à l'exécution de la première requête vers la DB.

Par la suite, nous allons procéder à la création du projet Unit Test, si vous n'avez pas coché la case de création de test unitaire au début, vous pouvez rajouter ce type de projet à ce moment, cliquez bouton droit sur votre solution :

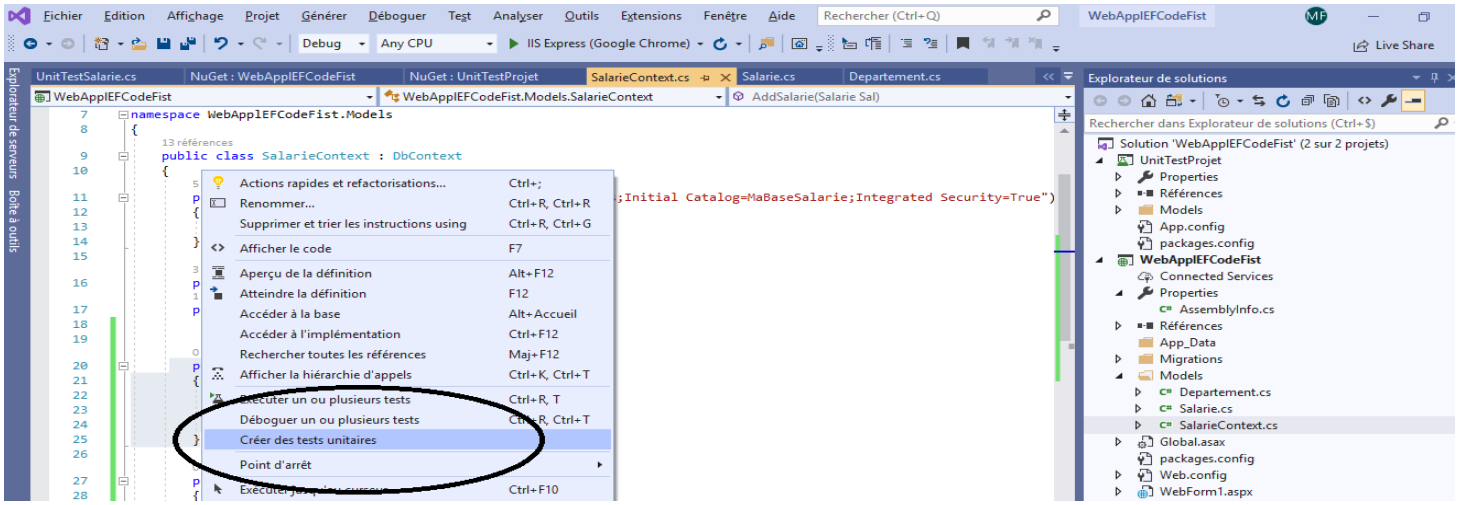
9. Choisissez Nouveau projet de type Test Unitaire (.Net Framework).



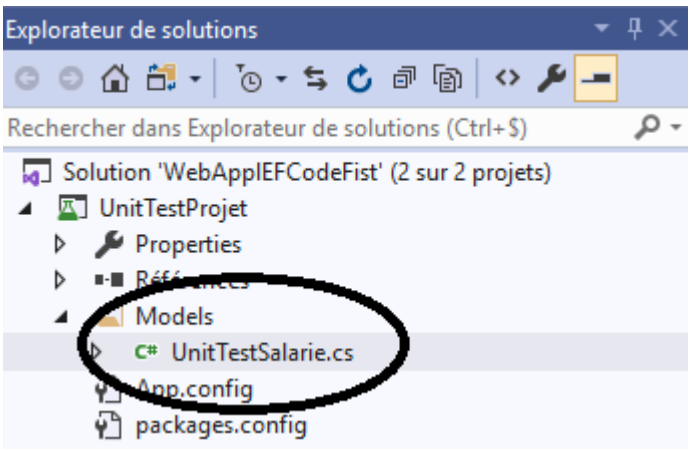
10. Un nouveau projet est rajouté à votre solution :



11. Allez maintenant à la méthode AddSalarie au niveau de la classe SalarieContext, Sélectionnez-là, puis cliquez bouton droit et choisissez « Créer des tests unitaires ».



12. Cela va vous créer un fichier UnitTestSalarie, avec une méthode « AddSalarieTest ».



13. Rajoutez le code suivant pour effectuer des tests sur les 3 méthodes, Ajout, Suppression et Récupération d'un salarié.

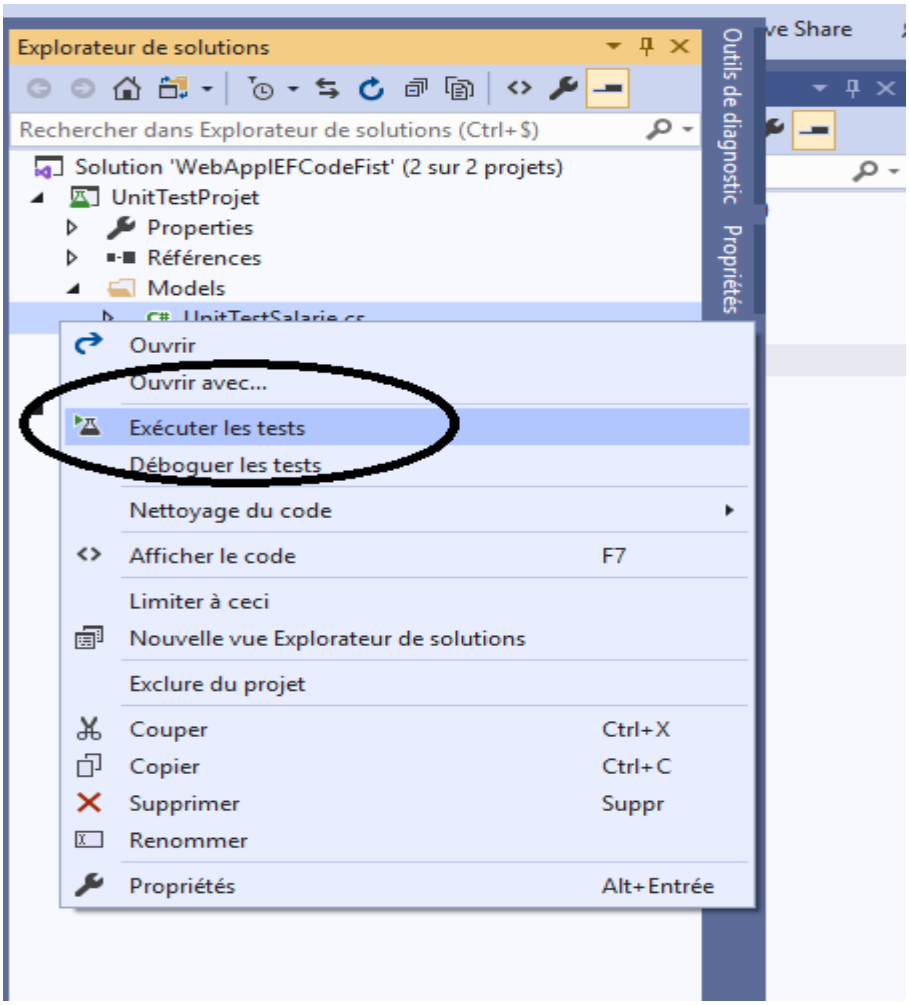
NB : Avant l'exécution de ce code, il faut avoir département avec un Id =1 au niveau de votre base de données.

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using WebApplEFCodeFist.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace WebApplEFCodeFist.Models.Tests
{
    [TestClass()]
    public class UnitTestSalarie
    {


        [TestMethod()]
        public void TestAddSalarie()
        {

            SalarieCRUD SalCrud = new SalarieCRUD();
            Salarie NewSalarie = new Salarie();
```

```
        NewSalarie.Nom = "Nom";
        NewSalarie.Prenom = "Prenom";
        NewSalarie.Fonctione = "Fonction";
        NewSalarie.DepartementId = 1;

        SalCrud.AddSalarie(NewSalarie);

        Salarie SalFind = SalCrud.GetSalarie("Prenom");

        Assert.AreEqual(SalFind.Nom.ToString(),"Nom");

        SalCrud.DelSalarie(NewSalarie);

    }

    [TestMethod]
    public void TestDelSalarie()
    {

        SalarieCRUD SalCrud = new SalarieCRUD();
        Salarie NewSalarie = new Salarie();
        NewSalarie.Nom = "Nom";
        NewSalarie.Prenom = "Prenom";
        NewSalarie.Fonctione = "Fonction";
        NewSalarie.DepartementId = 1;

        SalCrud.AddSalarie(NewSalarie);

        Salarie SalFind = SalCrud.GetSalarie("Prenom");

        Assert.AreEqual(SalFind.Nom.ToString(), "Nom");

        SalCrud.DelSalarie(NewSalarie);


    }

    [TestMethod]
    public void TestGetSalarie()
    {

        SalarieCRUD SalCrud = new SalarieCRUD();
        Salarie NewSalarie = new Salarie();
        NewSalarie.Nom = "Nom";
        NewSalarie.Prenom = "Prenom";
        NewSalarie.Fonctione = "Fonction";
        NewSalarie.DepartementId = 1;

        SalCrud.AddSalarie(NewSalarie);
```

```
            Salarie SalFind = SalCrud.GetSalarie("Prenom");

            Assert.AreEqual(SalFind.Nom.ToString(), "Nom");

            SalCrud.DelSalarie(NewSalarie);


        }
    }
}
```

14. Maintenant nous pouvons lancer le test pour voir son déroulement et le résultat, pour le faire, cliquer bouton droit sur le fichier « UnitTestSalarie », puis choisissez, « Exécuter les tests ».



15. A l'exécution des tests, cela affiche la fenêtre « Test Explorer » et vous pouvez voir l'exécution de chaque méthode et le résultat après l'exécution :