

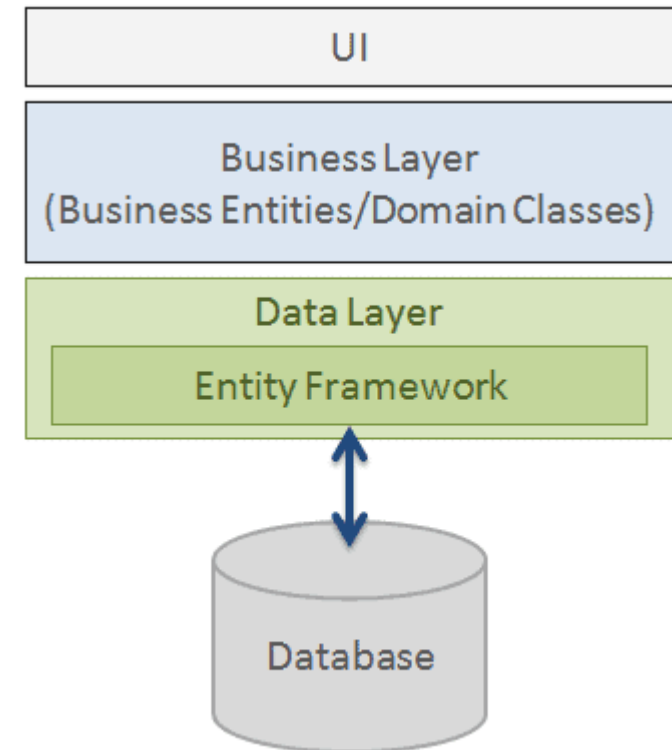
# Utilisation d'Entity Framework pour l'accès aux bases de données

---

MOHAMED AIROUCHE

# L'ORM Entity Framework

- L'utilisation d'une base de données au sein d'une application développée en programmation orientée objet nécessite une liaison entre la base de données et les classes du modèle représentant les tables et vice-versa.
- Les performances d'une application dépendent en partie de système de gestion de base de données relationnelle et des requêtes SQL écrites dans l'application. Parfois, ces derniers ne sont pas optimisés.
- Il existe des ORM (Object Relational mapping) qui sont des outils ou des techniques permettant de lier (**mapper**) des tables d'une base de données relationnelle avec des objets utilisés dans les programmes. Ces techniques ont fortement simplifiée le débogage des requêtes SQL.
- L'ORM fait un pont entre les objets manipulés par l'application Web et les lignes et les colonnes des données d'une base de données relationnelle.



© EntityFrameworkTutorial.net

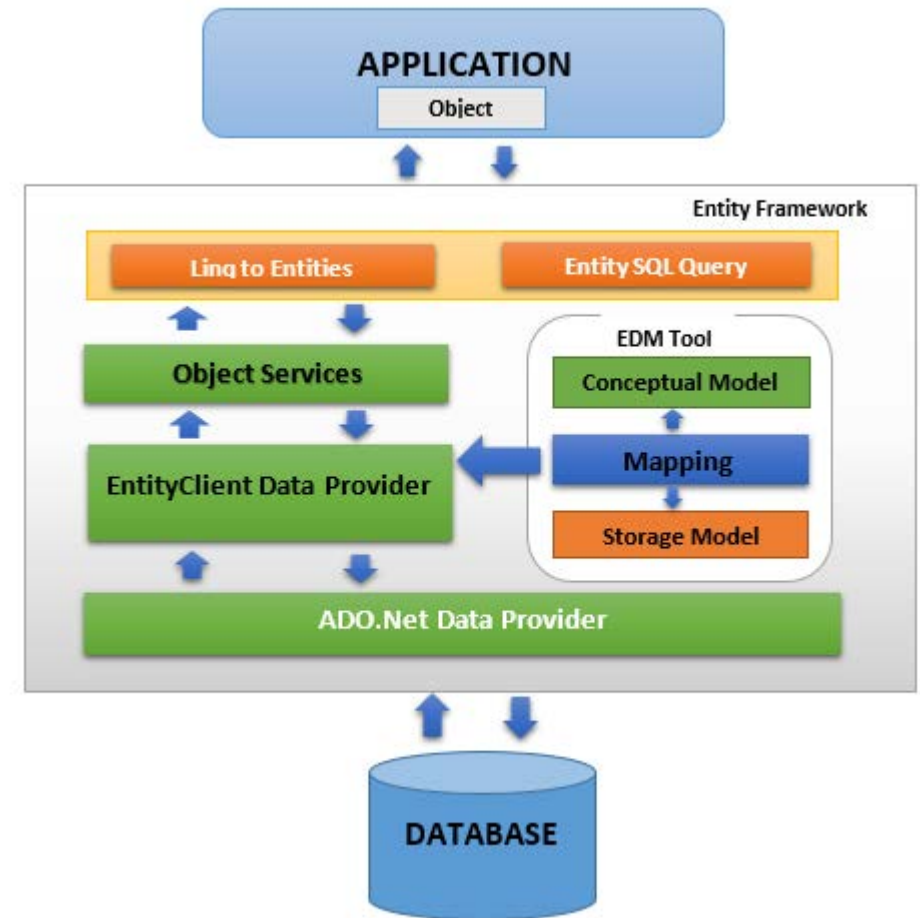
<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

# L'ORM Entity Framework

- ✓ **Entity Framework:** est une brique logicielle permettant de faciliter l'accès à une base de donnée.
- ✓ Ce Framework est un ORM(Object-Relational Mapping), c'est à dire un outil qui permet de travailler des objets C# plutôt qu'avec des requêtes en dur.
- ✓ Le développeur va ainsi gérer, ajouter ou supprimer des objets via l'ORM.
- ✓ L'ORM s'occuper de traduire automatiquement les opérations via des requêtes SQL.

## Architecture d'Entity Framework

Entity Framework est un outil open source, ORM (Object-Relational Mapper) pour accéder à la base de données dans une approche orientée objet utilisant la technologie .NET.



<https://www.c-sharpcorner.com/article/get-your-hands-on-entity-framework-code-first-workflow/>

# L'ORM Entity Framework

Entity Framework utilise 3 types d'approches : <https://www.entityframeworktutorial.net/choosing-development-approach-with-entity-framework.aspx>

## Database first (La base de données d'abord)

Dans l'approche Database first, nous concevons nos tables à l'aide d'un concepteur visuel ou à l'aide d'une requête SQL. Ensuite, Entity Framework génère une classe de domaine. C'est l'approche traditionnelle que de nombreux développeurs suivent depuis la version initiale d'EF 1 et de Visual Studio 2008.

## Code first (Code d'abord)

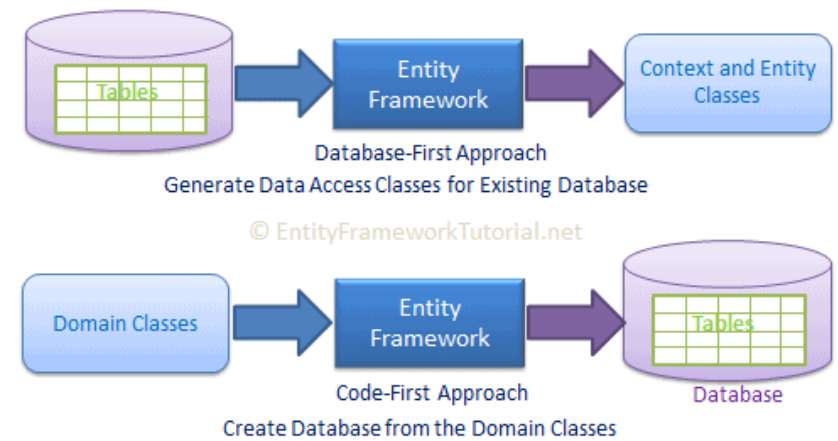
Dans l'approche Code first, nous créons notre classe de domaine, puis l'EF génère une base de données pour nous. Le Code First a été publié avec EF 4.1.1 et Visual Studio 2010.

## Model first (Modèle d'abord)

Dans l'approche Model first, nous utilisons le concepteur visuel de Visual Studio pour modéliser notre classe et ses associés. C'est comme un diagramme UML et sur cette base, EF génère une classe de domaine et une base de données. Il a été publié avec EF 4.0 et Visual Studio 2010.

<https://www.c-sharpcorner.com/article/get-your-hands-on-entity-framework-code-first-workflow/>

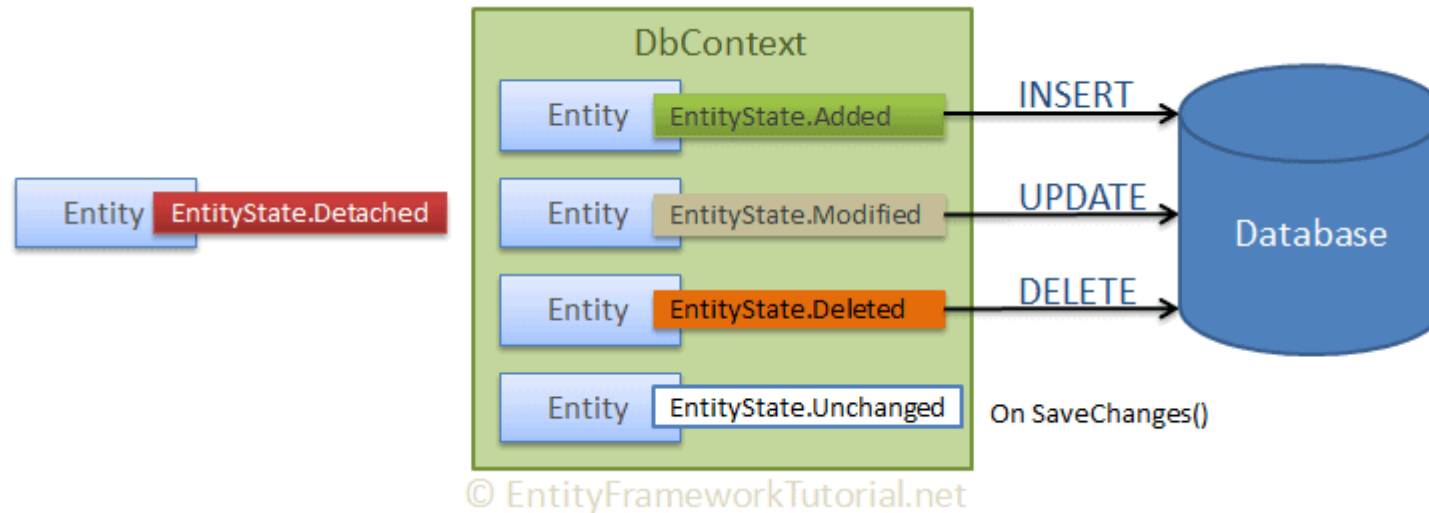
<https://stackoverflow.com/questions/37975075/entity-framework-clarification-about-model-objects-and-their-methods>



# L'ORM Entity Framework

Entity Framework se base sur les éléments suivants :

- ✓ La classe de contexte de la base de données. Ce contexte est une classe dérivée de la classe système [System.Data.Entity.DbContext]. Elle sert à définir les images objets des tables de la base de données. Elle est utilisée pour interroger ou enregistrer des données dans la base de donnée en effectuant les opérations CRUD (Créer, Lire, Mettre à jour, Supprimer).
- ✓ Les classes des entités Entity framework, qui sont des classes dans lesquelles on encapsule les lignes des différentes tables de la base de données.



<https://www.entityframeworktutorial.net/basics/entity-states.aspx>

# Exemple

---

## **Développement d'une application qui permet de gérer des employés :**

On souhaite créer une application qui permet de gérer des employés. Chaque employé est défini par son identifiant, son nom, son prénom, son sexe, son département et sa ville.

L'application doit permettre de :

- Afficher tous les employés.
- Afficher les détails d'un employé.
- Saisir et ajouter un nouvel employé.
- Éditer et modifier un employé.
- Supprimer un employé.

Les employés sont stockés dans une base de données locale SQL Server Express.

Dans cet exemple, l'application utilisera Entity Framework pour l'accès aux données de la base de données.

# Exemple

---

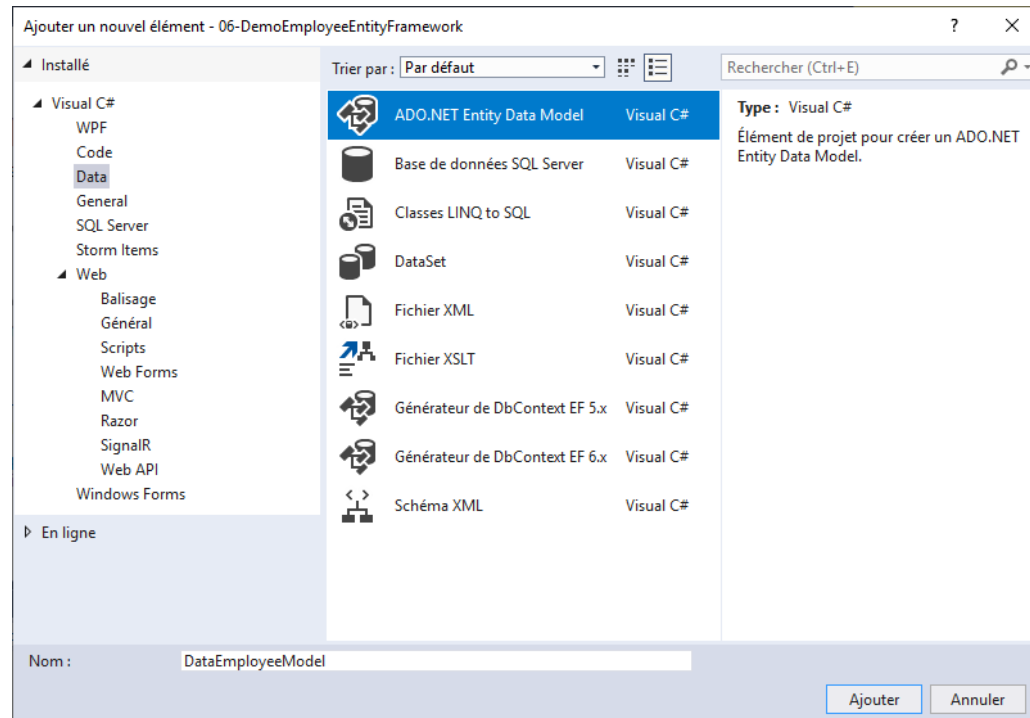
## Création d'une application ASP.NET MVC :

- Créez une application ASP.NET MVC : **06-DemoEmployeeEntityFramework**.
- Dans cet exemple nous utiliserons la base de donnée DemoEmployees et la table tbEmployee créés dans le cours précédent.
- L'approche Database first de l'ORM Entity Framework sera utilisée pour l'accès aux données de la base de données.

# Exemple

## Ajout d'un modèle de données Entity Framework :

- ✓ Pour ajouter un modèle de données en utilisant l'approche DB-First d'ORM Entity Framework, cliquez avec le bouton droit sur le dossier Models de votre projet et sélectionnez Ajouter - Nouvel élément. Cela ouvrira la fenêtre contextuelle Ajouter un nouvel élément comme indiqué ci-dessous.

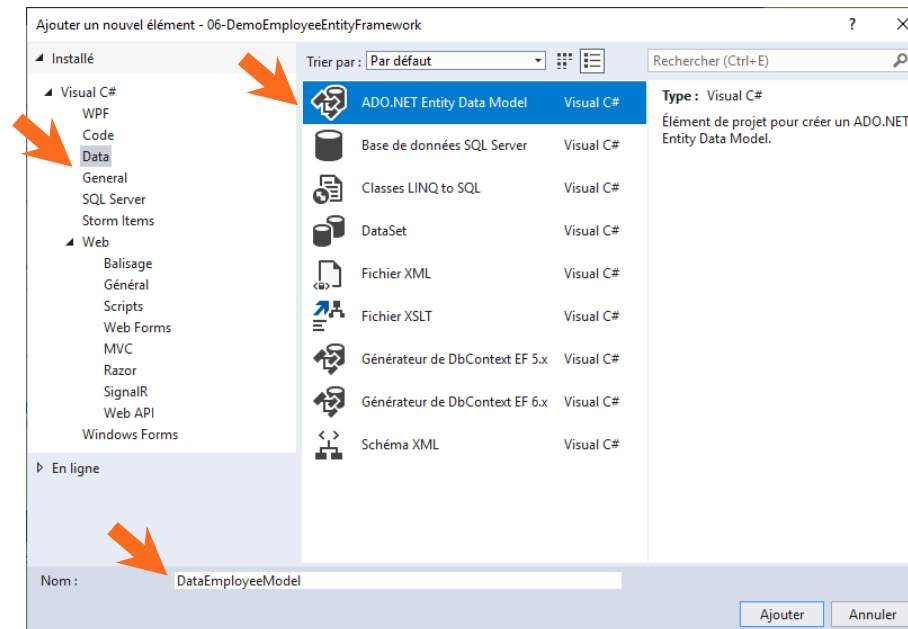




# Exemple

## Ajout d'un modèle de données Entity Framework :

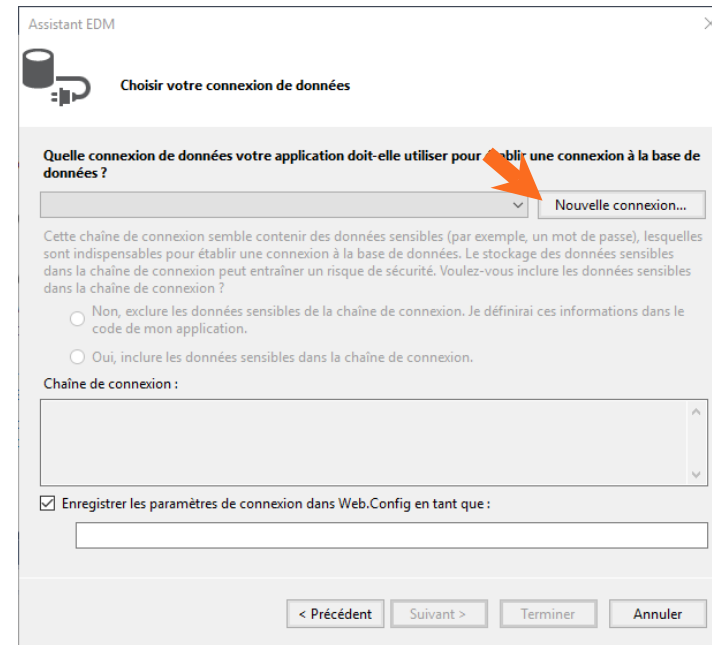
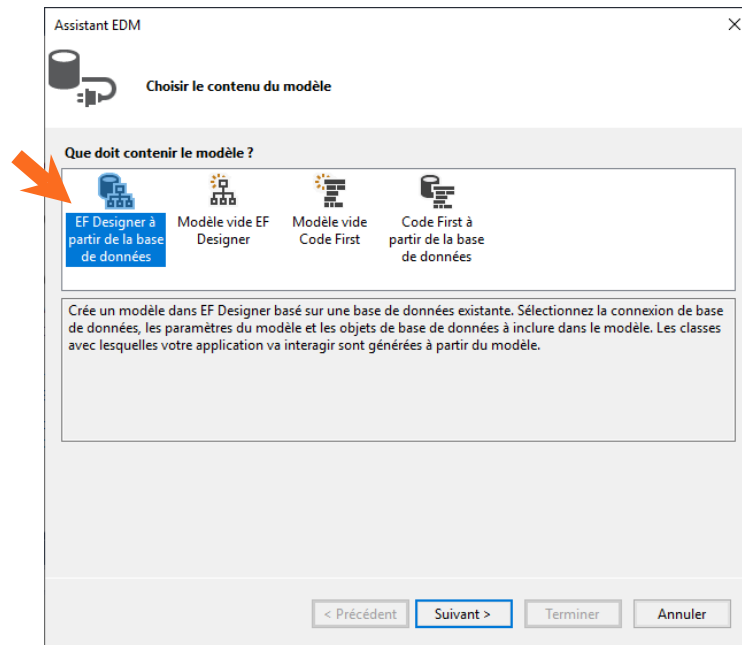
- ✓ Sélectionnez "Données" dans le volet gauche.
- ✓ Sélectionnez «ADO.NET Entity Data Model» dans le volet central.
- ✓ Dans la zone de texte Nom, entrez le nom d'un modèle de données **DataEmployeeModel** et cliquez sur Ajouter. Cela ouvrira l'assistant de modèle de données d'entité à l'aide duquel vous pouvez générer un modèle de données d'entité pour une base de données existante.



# Exemple

## Ajout d'un modèle de données Entity Framework :

- ✓ Dans l'assistant de modèle de données d'entité, sélectionnez l'option "EF Designer from database" et cliquez sur suivant.
- ✓ Sur l'écran suivant, cliquez sur le bouton "Nouvelle connexion".
- ✓ Choisir «Microsoft SQL Server» et cliquer sur OK ou Continuer.



# Exemple

- ✓ Dans la fenêtre "Propriétés de connexion", définissez le nom du serveur = (local) ou . Si vous utilisez «SQL Server» ou choisir le nom du serveur local : (localdb)\MSSQLLocalDB.
- ✓ Authentification = Authentification Windows.
- ✓ Sélectionnez ou entrez un nom de base de données "DemoEmployeeDB".

Propriétés de connexion

Entrez les informations pour vous connecter à la source de données sélectionnée ou cliquez sur "Modifier" pour sélectionner une autre source de données et/ou un autre fournisseur.

Source de données : Microsoft SQL Server (SqlClient) [Modifier...]

Nom du serveur : [Actualiser]

Connexion au serveur

Authentification : Authentification Windows

Nom d'utilisateur : [ ]

Mot de passe : [ ]

☐ Enregistrer mon mot de passe

Connexion à la base de données

☒ Sélectionner ou entrer un nom de base de données : [ ]

☐ Attacher un fichier de base de données : [Parcourir...]

Nom logique : [ ]

[Avancées...]

[Tester la connexion] [OK] [Annuler]

Propriétés de connexion

Entrez les informations pour vous connecter à la source de données sélectionnée ou cliquez sur "Modifier" pour sélectionner une autre source de données et/ou un autre fournisseur.

Source de données : Microsoft SQL Server (SqlClient) [Modifier...]

Nom du serveur : (localdb)\MSSQLLocalDB [Actualiser]

Connexion au serveur

Authentification : Authentification Windows

Nom d'utilisateur : [ ]

Mot de passe : [ ]

☐ Enregistrer mon mot de passe

Connexion à la base de données

☒ Sélectionner ou entrer un nom de base de données : [ ]

- DemoEmployeeDB
- EmployeeDB
- master
- model
- msdb
- tempdb

☐ Attacher un fichier de base de données : [Parcourir...]

Nom logique : [ ]

[Avancées...]

[Tester la connexion] [OK] [Annuler]

Propriétés de connexion

Entrez les informations pour vous connecter à la source de données sélectionnée ou cliquez sur "Modifier" pour sélectionner une autre source de données et/ou un autre fournisseur.

Source de données : Microsoft SQL Server (SqlClient) [Modifier...]

Nom du serveur : (localdb)\MSSQLLocalDB [Actualiser]

Connexion au serveur

Authentification : Authentification Windows

Nom d'utilisateur : [ ]

Mot de passe : [ ]

☐ Enregistrer mon mot de passe

Connexion à la base de données

☒ Sélectionner ou entrer un nom de base de données : DemoEmployeeDB [ ]

☐ Attacher un fichier de base de données : [Parcourir...]

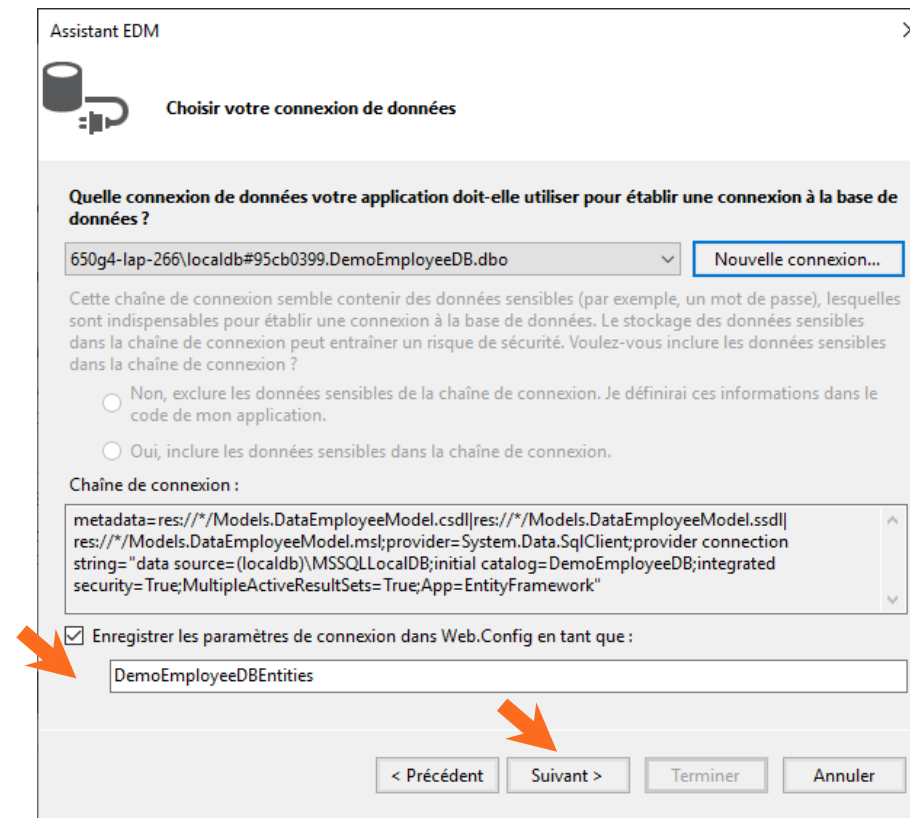
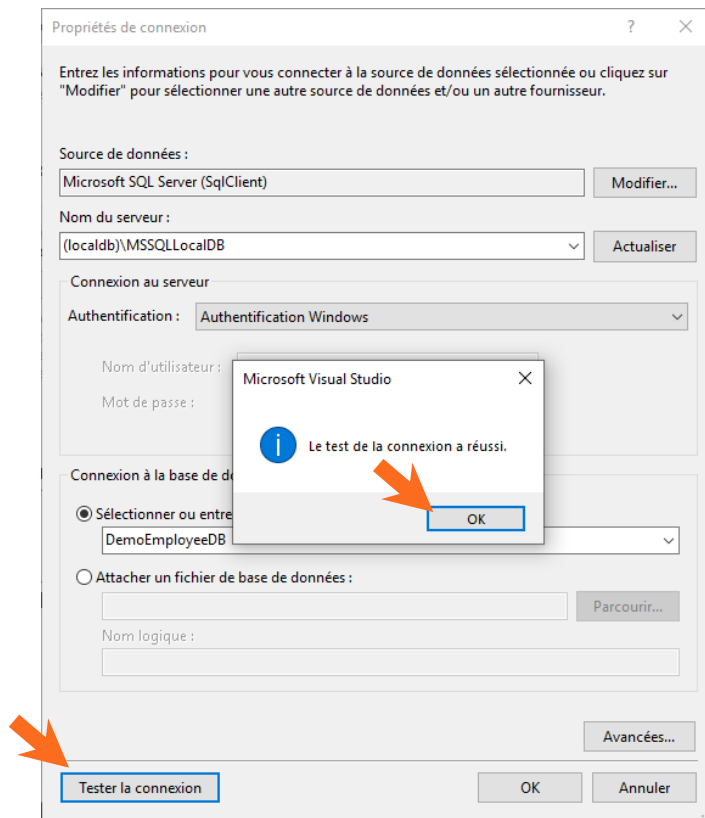
Nom logique : [ ]

[Avancées...]

[Tester la connexion] [OK] [Annuler]

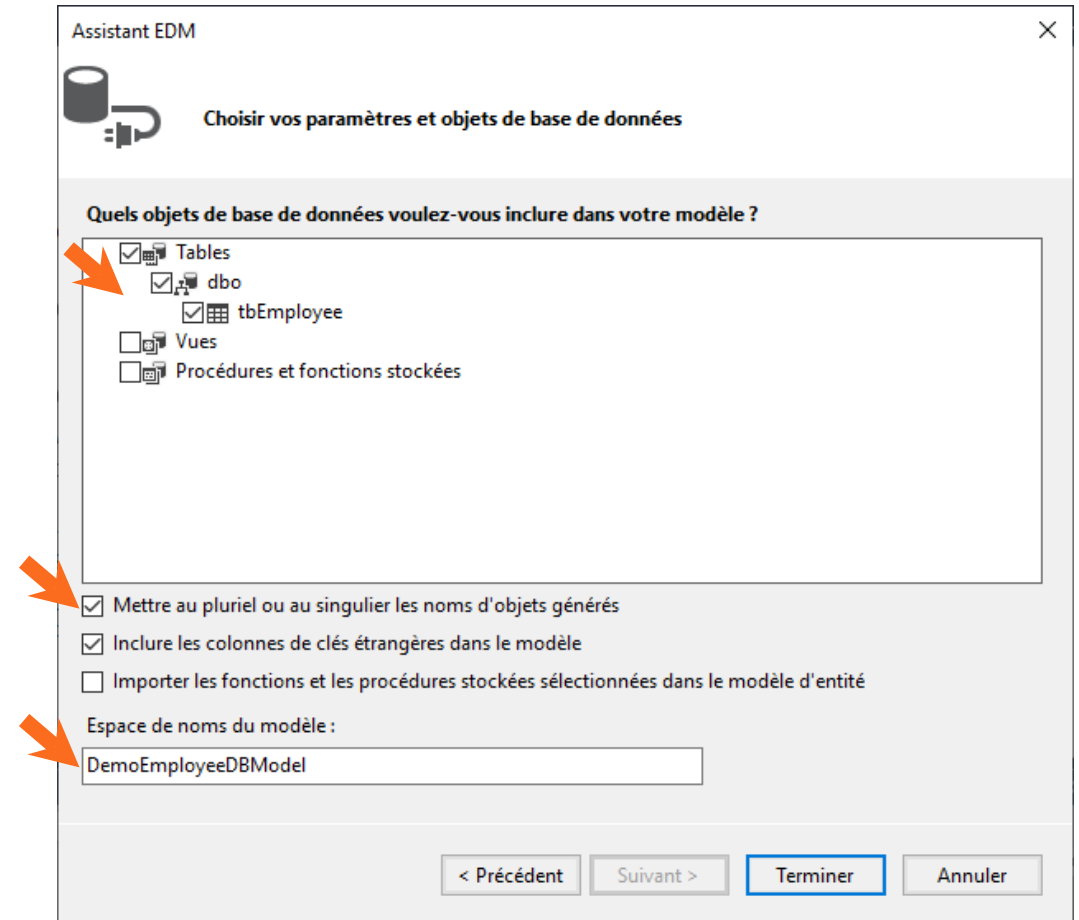
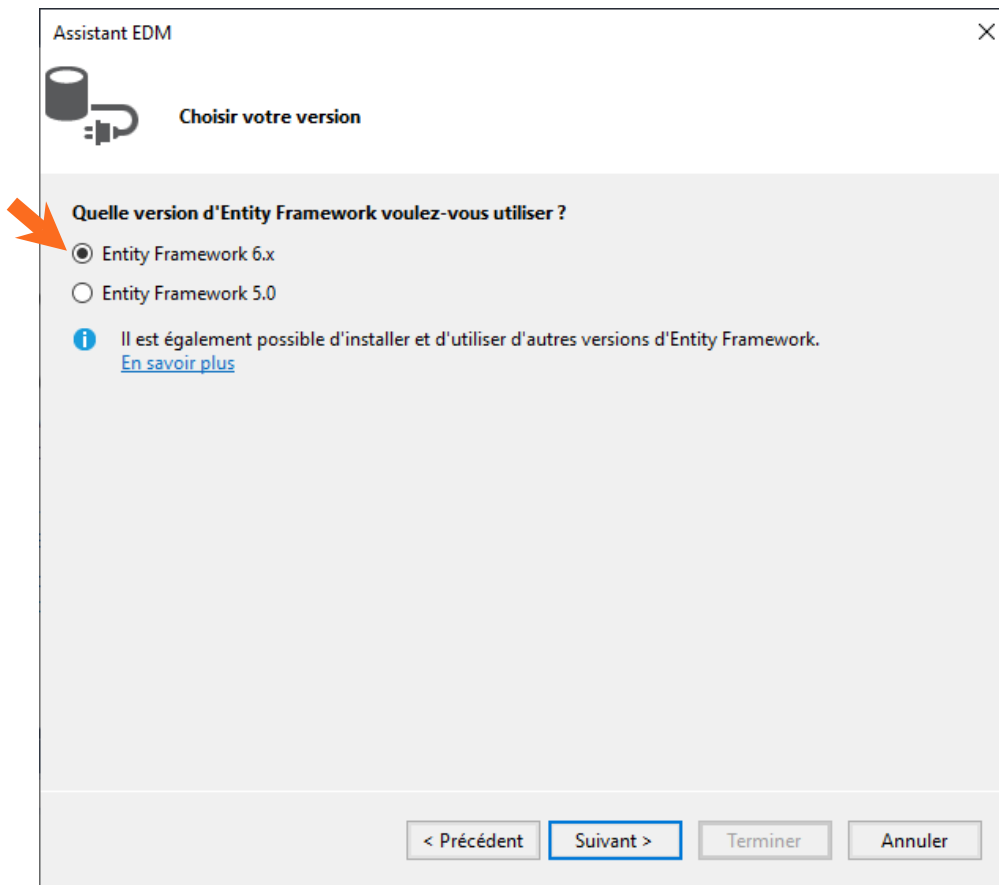
# Exemple

Testez la connexion. Puis cliquez sur Ok, on obtient une fenêtre qui nous demande si on désire enregistrer les paramètres de connexion dans le fichier App.Config. On garde la même configuration.



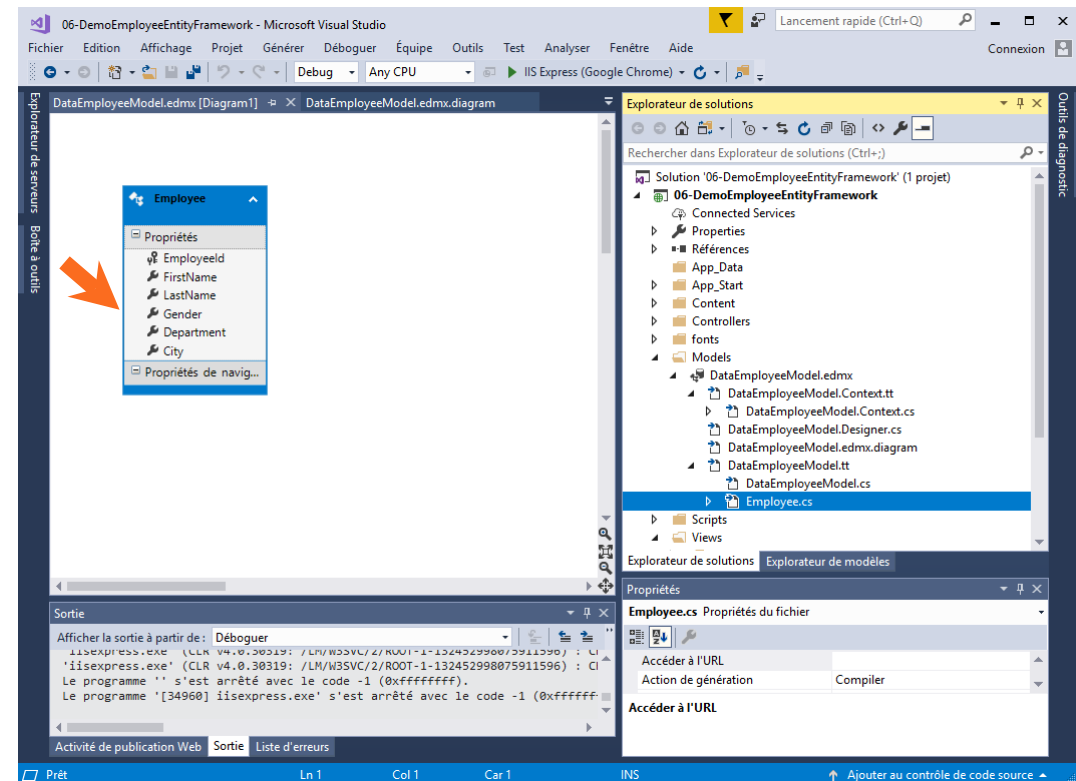
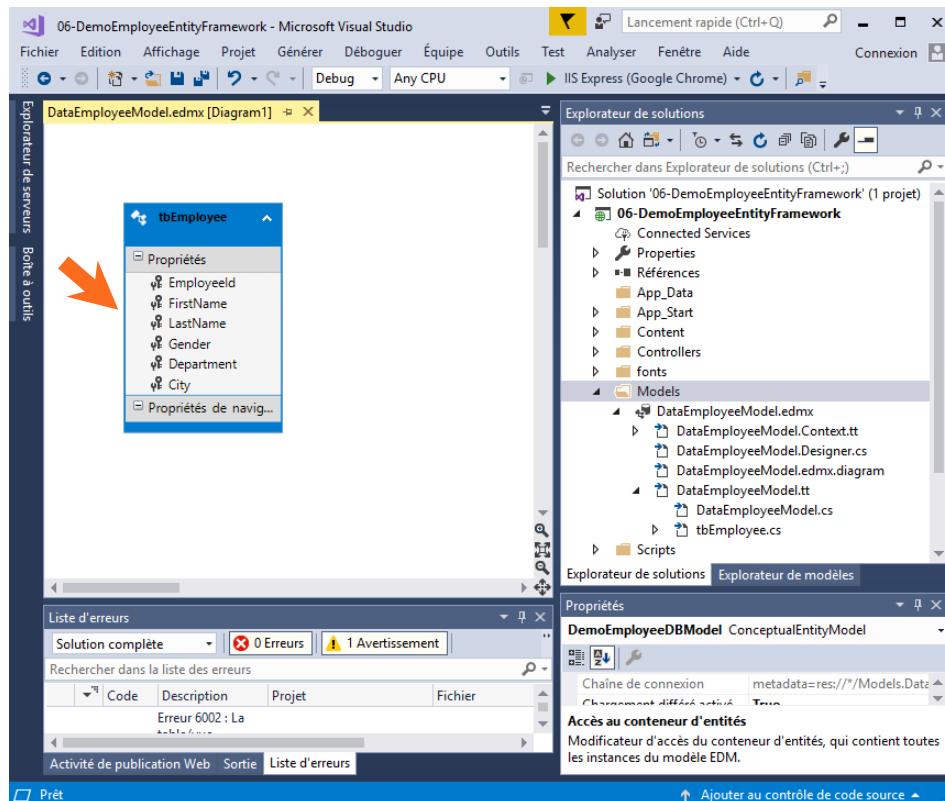
# Exemple

Sélectionnez Entity Framework 6.0. Puis, sélectionnez la table tbEmployee.



# Exemple

Le modèle de données DataEmployeeModel est créé. Vérifiez votre modèle et apportez les modifications au besoin. Dans notre exemple, on sélectionne EmployeeID comme id de l'entité et on modifie le nom de la classe de tbEmployee à Employee.



# Exemple

---

- ✓ La classe d'entité générée : **Employee**

```
using System;
using System.Collections.Generic;

public partial class Employee
{
    public int EmployeeId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Gender { get; set; }
    public string Department { get; set; }
    public string City { get; set; }
}
```

- ✓ La classe de contexte de la base de données générée : **DemoEmployeeDBEntities**

```
using System;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;

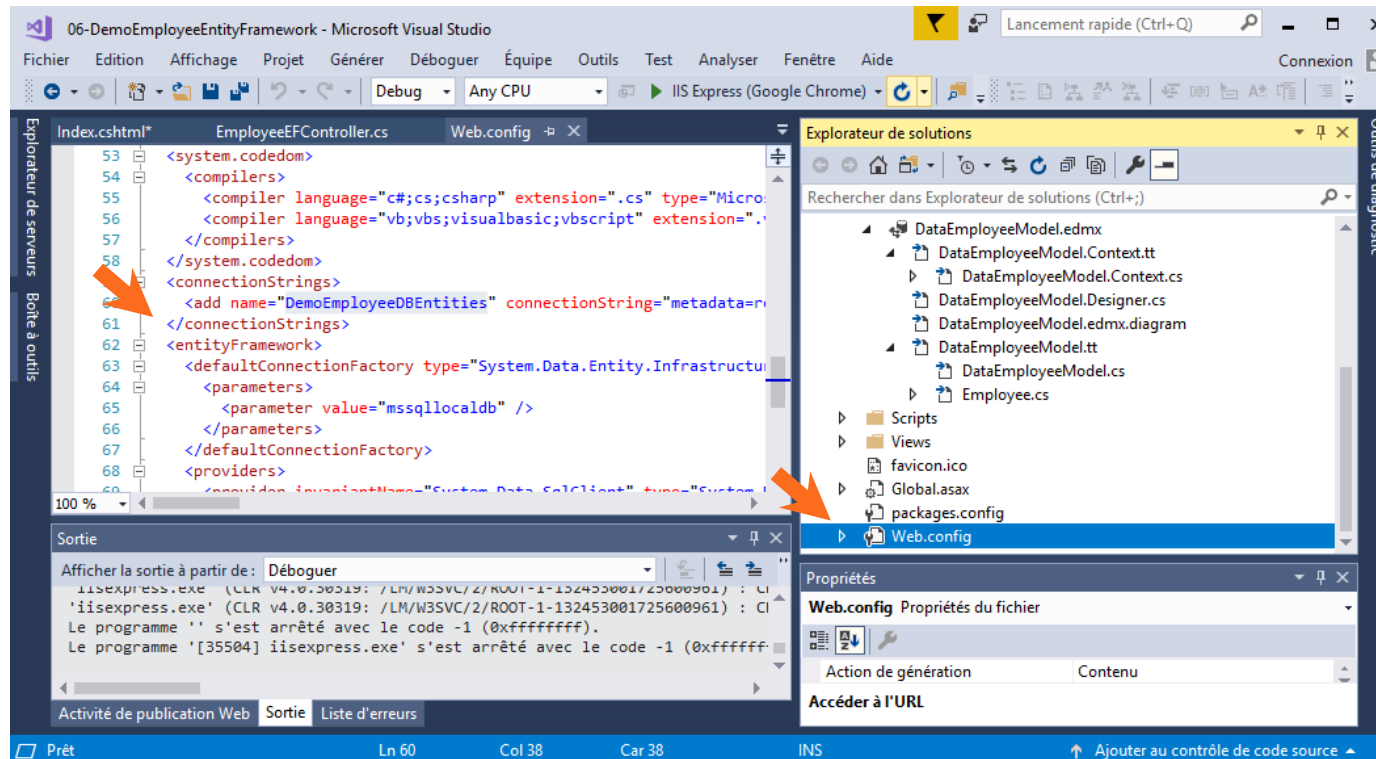
public partial class DemoEmployeeDBEntities : DbContext
{
    public DemoEmployeeDBEntities()
        : base("name=DemoEmployeeDBEntities")
    {
    }

    protected override void OnModelCreating(DbModelBuilder modelBuilder)
    {
        throw new UnintentionalCodeFirstException();
    }

    public virtual DbSet<Employee> Employees { get; set; }
}
```

# Exemple

Chaine de connexion nommée «DemoEmployeeDBEntities» est ajoutée dans le fichier Web.config. Elle pointe vers la BD qui contient la table tbEmployee.



Générer la solution.

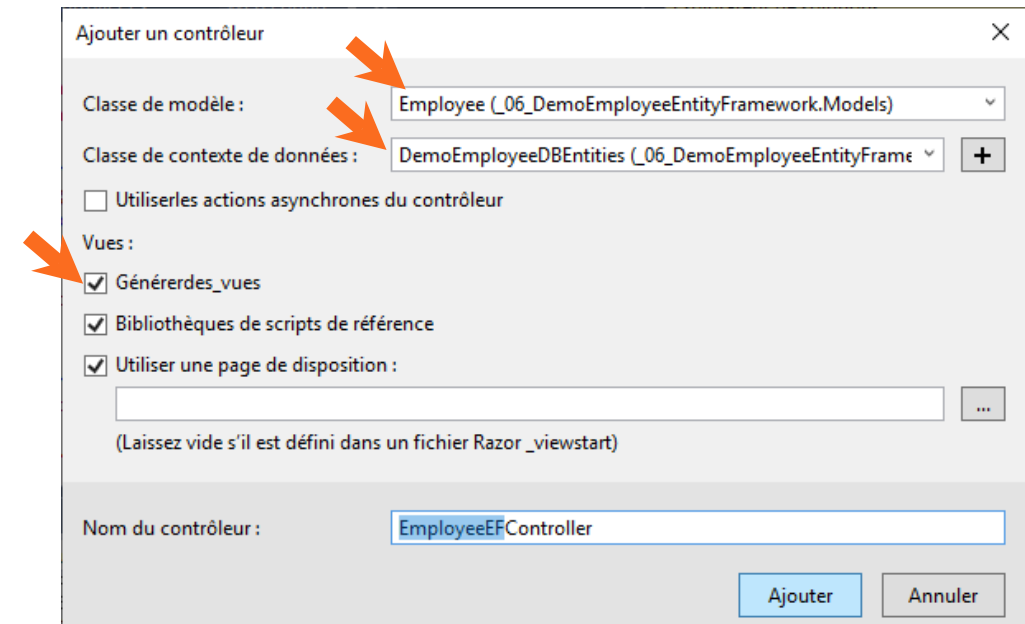
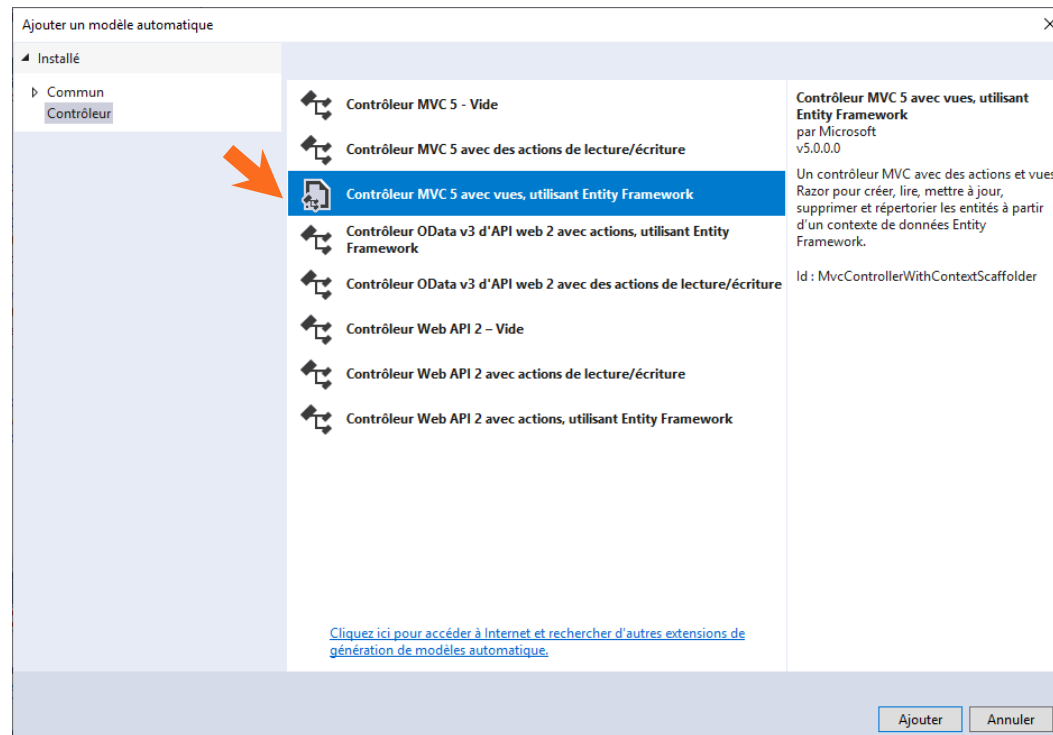


# Exemple

## Ajout d'un contrôleur à l'application :

Faire un clic droit sur le dossier «Controllers» du projet >> Ajouter >> Contrôleur.

Sélectionner le modèle Contrôleur MVC 5 avec vues, utilisant Entity Framework.



# Exemple

---

Visual studio 2017 vas créer automatiquement les méthodes d'actions suivantes :

- Index
- Create (GET, POST)
- Edit (GET, POST)
- Delete (GET, POST)
- Details

De plus, pour chaque méthode d'action de contrôleur Visual studio 2017 ajoute une vue. Au final, on doit se trouver avec les vues suivantes :

- Index.cshtml
- Create.cshtml
- Edit.cshtml
- Delete.cshtml
- Details.cshtml

