TP Gaming – .net 7 (Visual Studio 2022)

Table des matières

[1. Création du projet avec Visual Studio 2022 2](#_Toc127368723)

[2. Test de bon fonctionnement 4](#_Toc127368724)

[3. Connexion de Visual Studio à Azure 7](#_Toc127368725)

[4. Création du modèle « VirtualM » 8](#_Toc127368726)

[5. Mise à jour du DbContext 10](#_Toc127368727)

[6. Mise à jour de la base de données 11](#_Toc127368728)

[7. Consultation de la base de données 13](#_Toc127368729)

[8. Génération du controller et des views 14](#_Toc127368730)

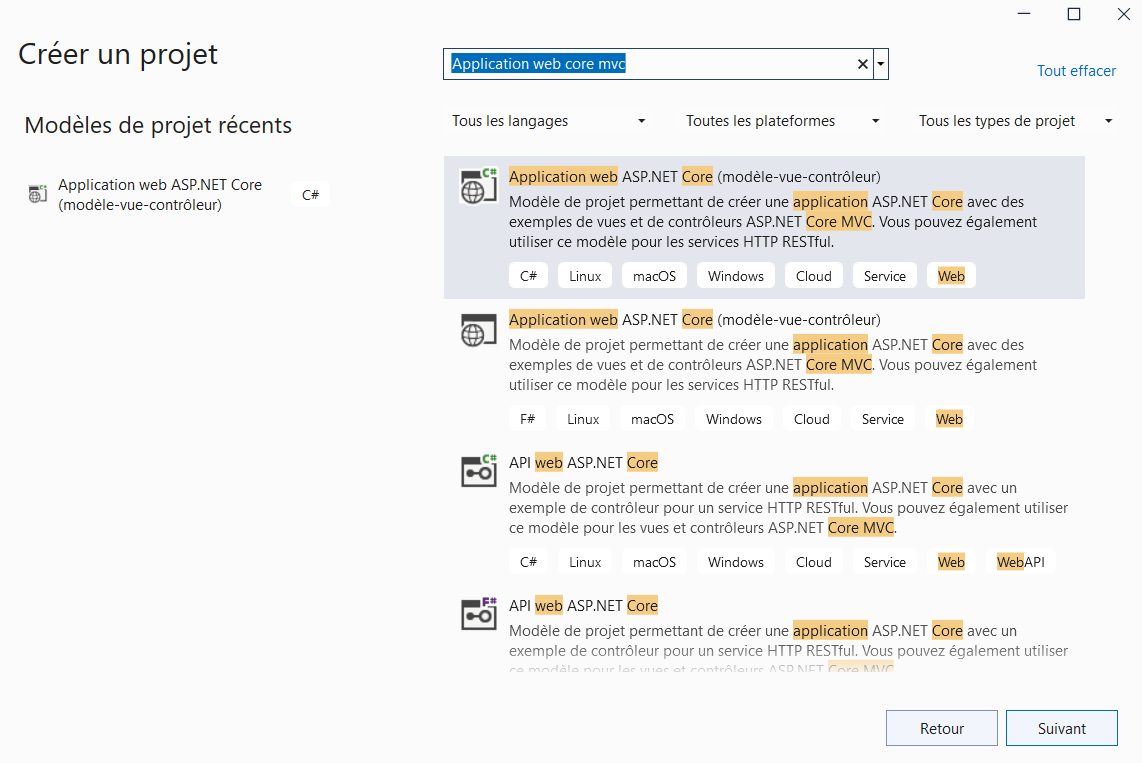
[9. Ajout d’un lien d’accès aux views de VirtualM 16](#_Toc127368731)

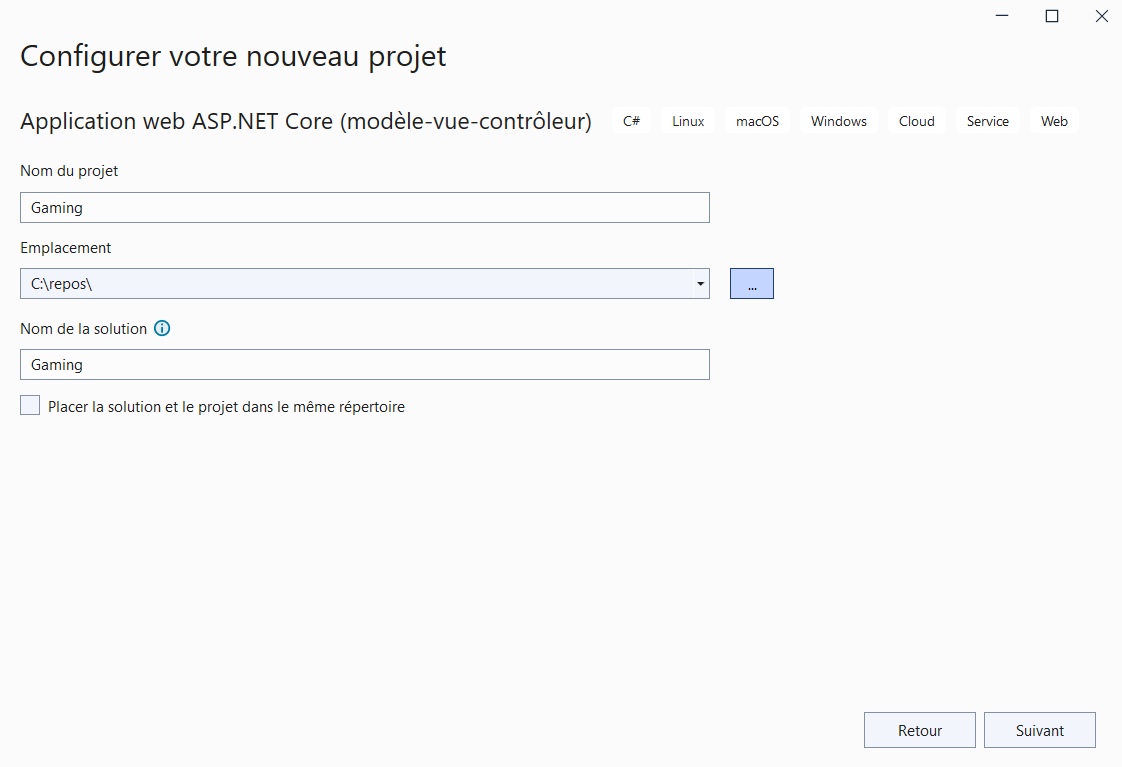
[10. Mise à jour de la vue de création d’une VM 17](#_Toc127368732)

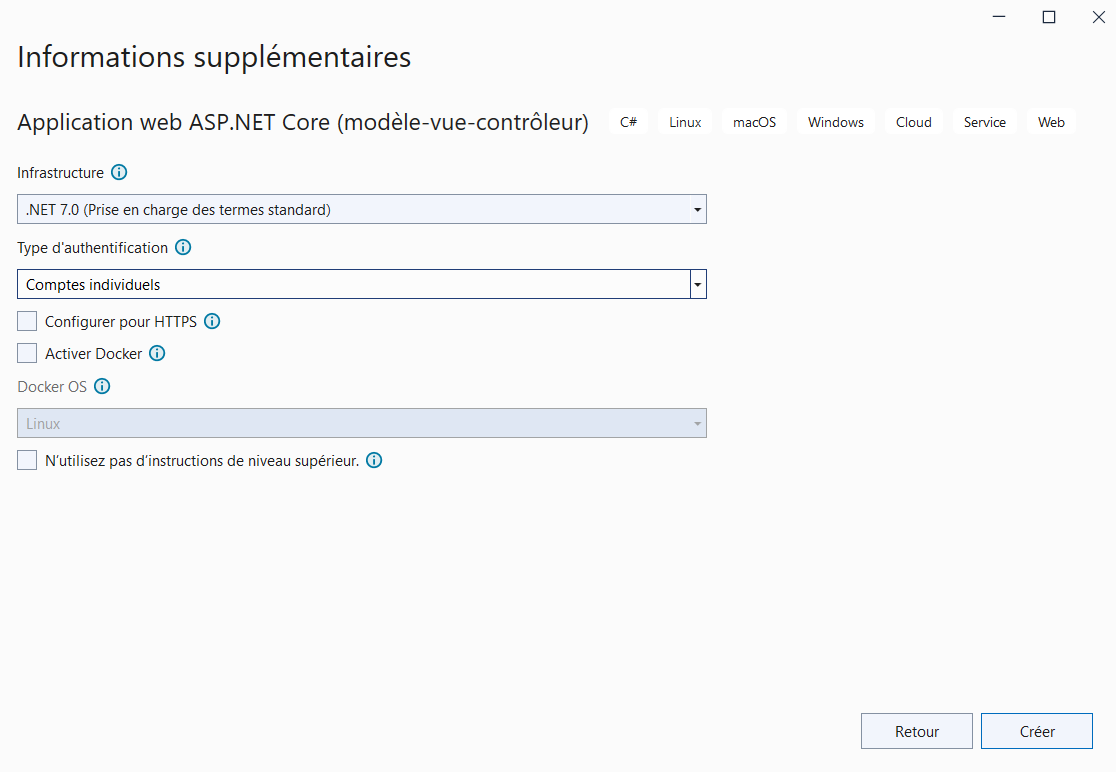
[11. Création d’un username pour l’utilisateur connecté 19](#_Toc127368733)

[12. Appels Azure 20](#_Toc127368734)

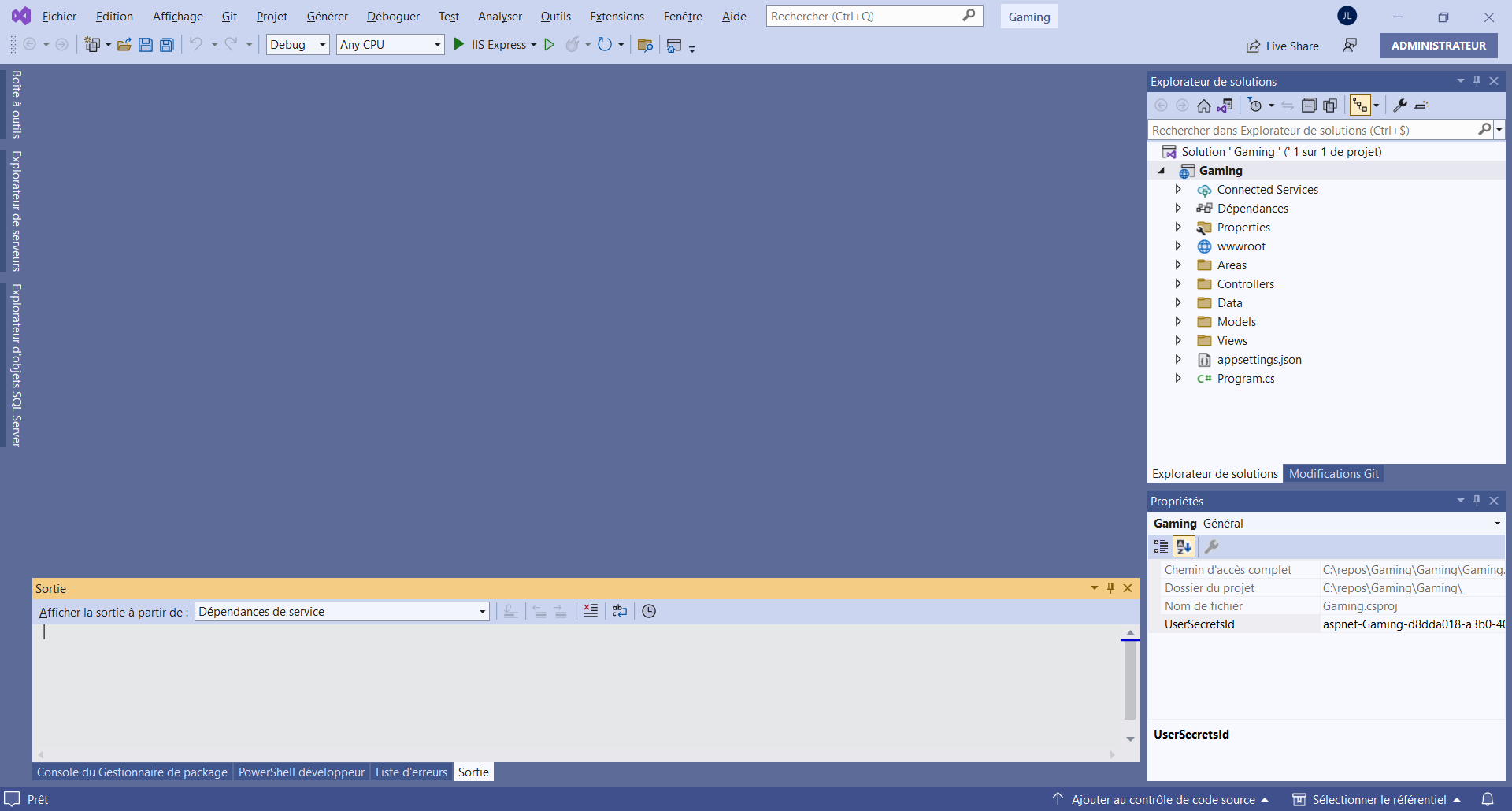
# Création du projet avec Visual Studio 2022



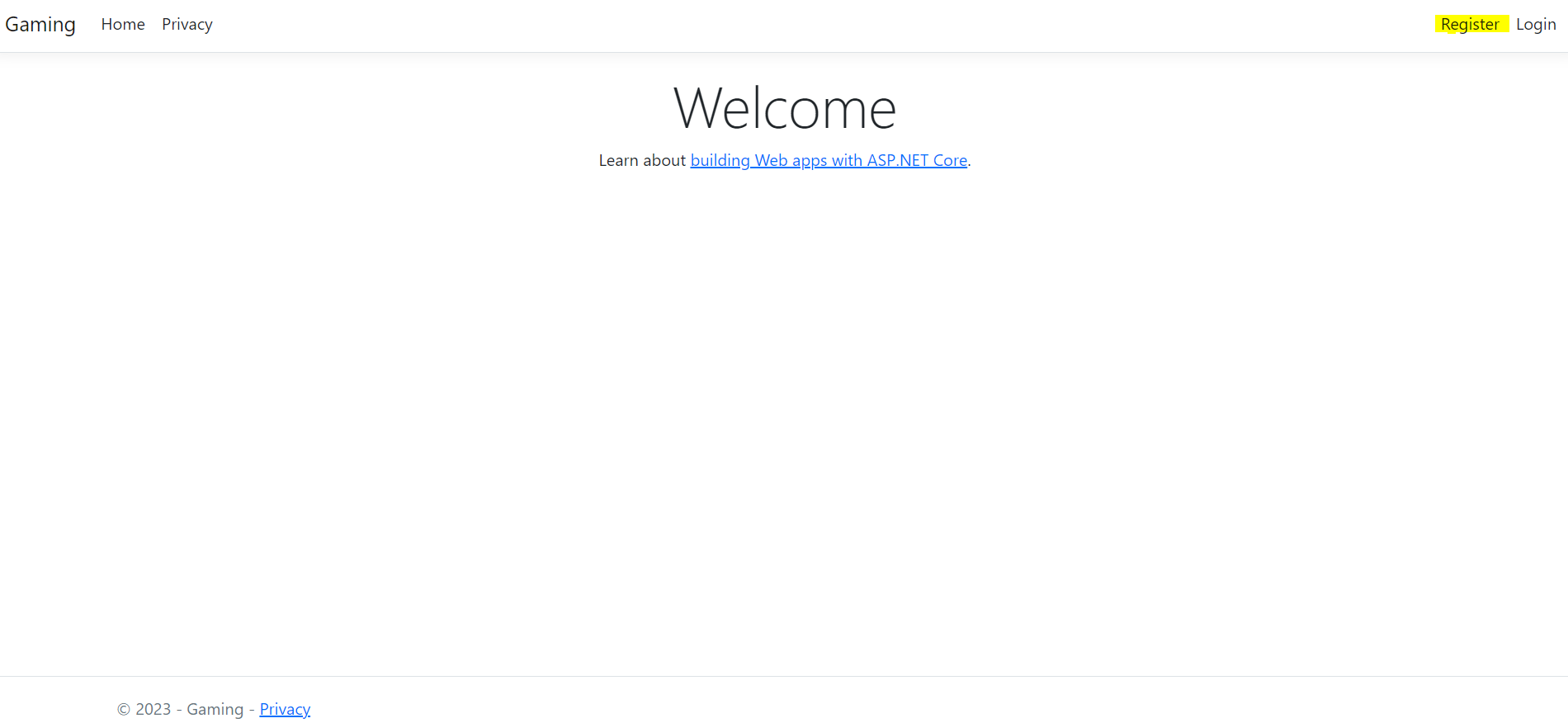


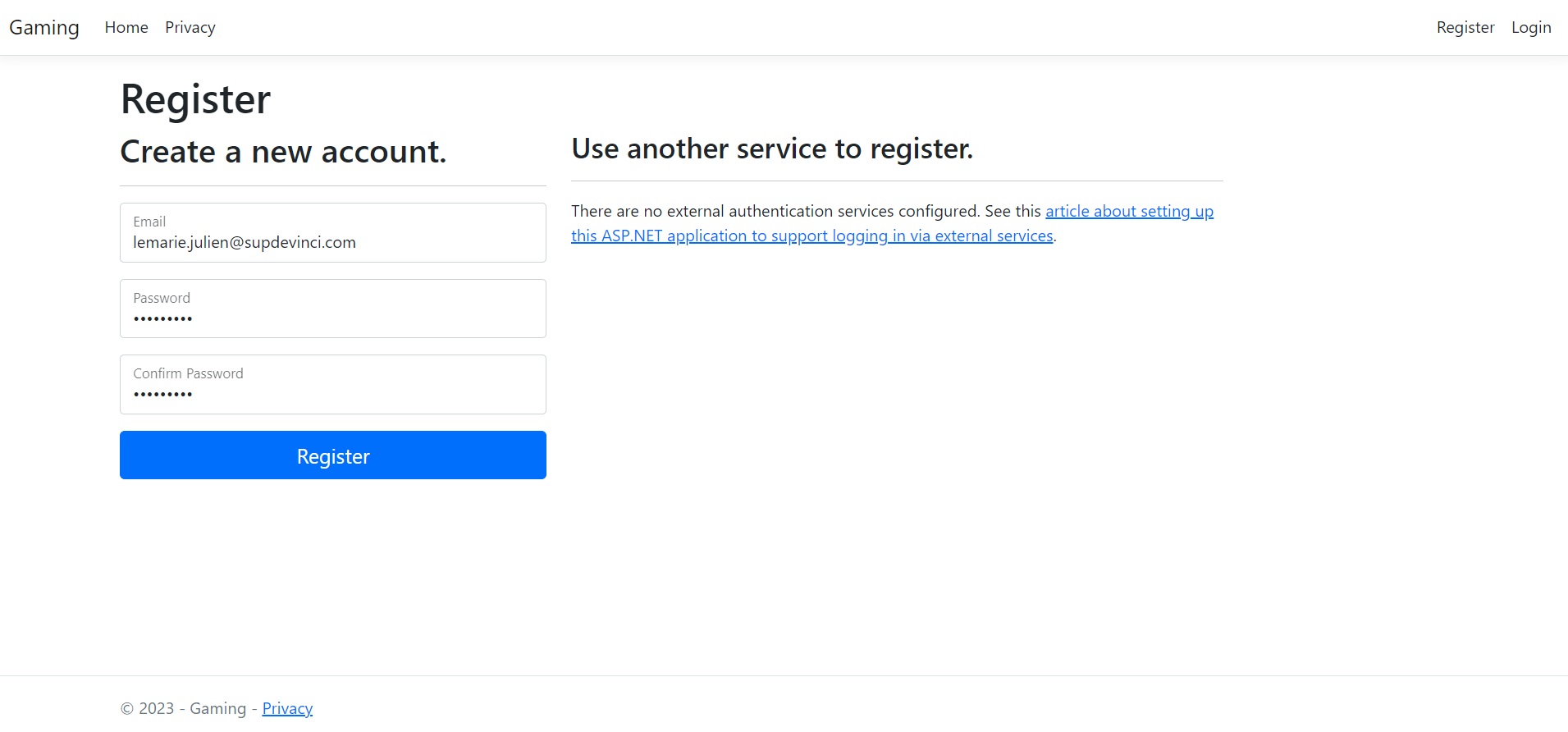


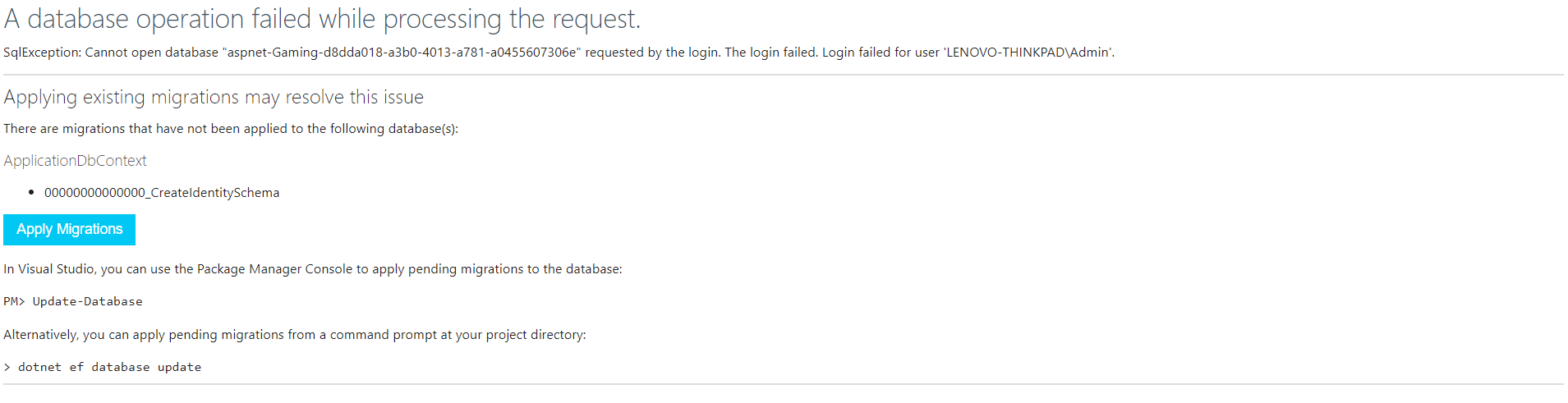
# Test de bon fonctionnement



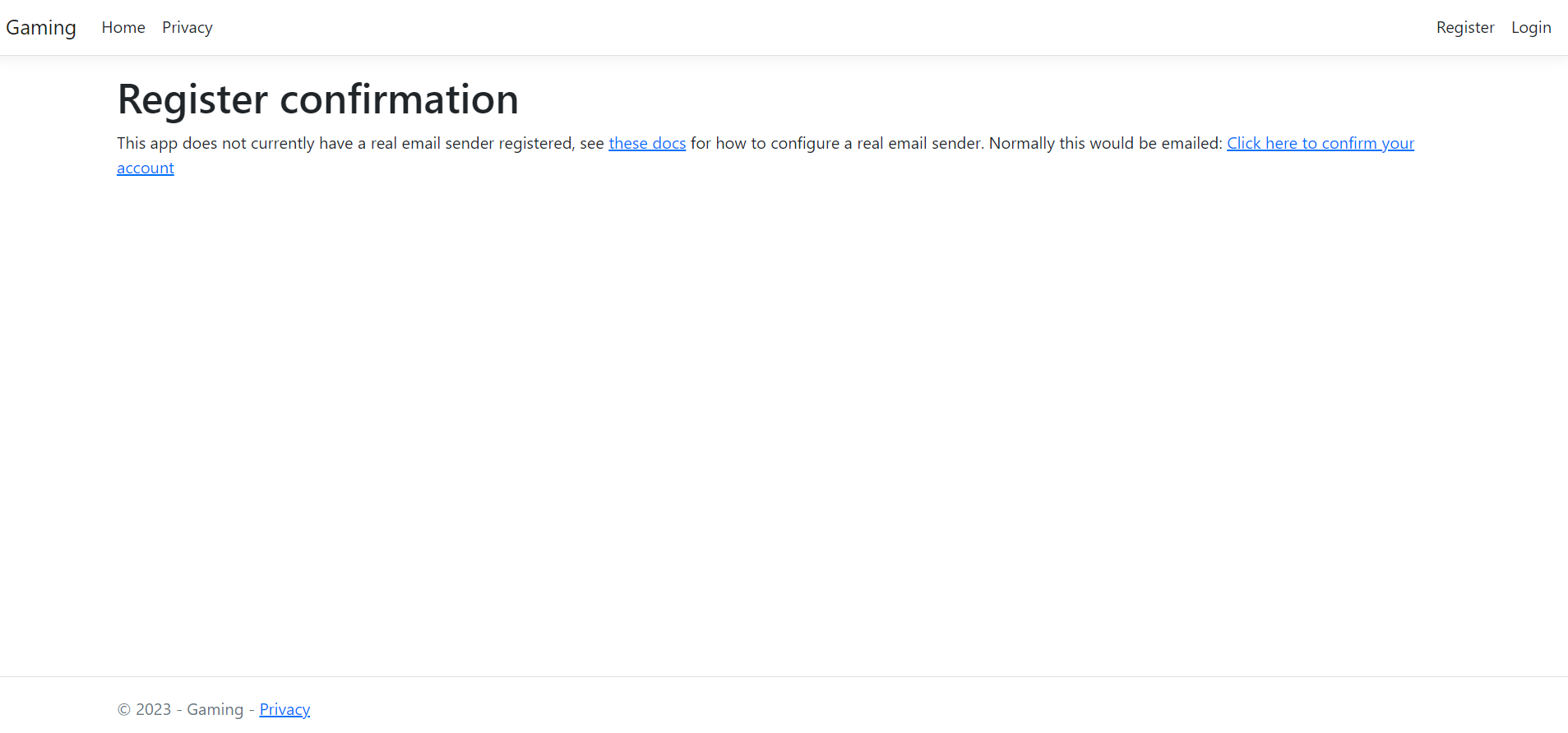
Lancer le projet en debug en cliquant sur l’icône verte : 





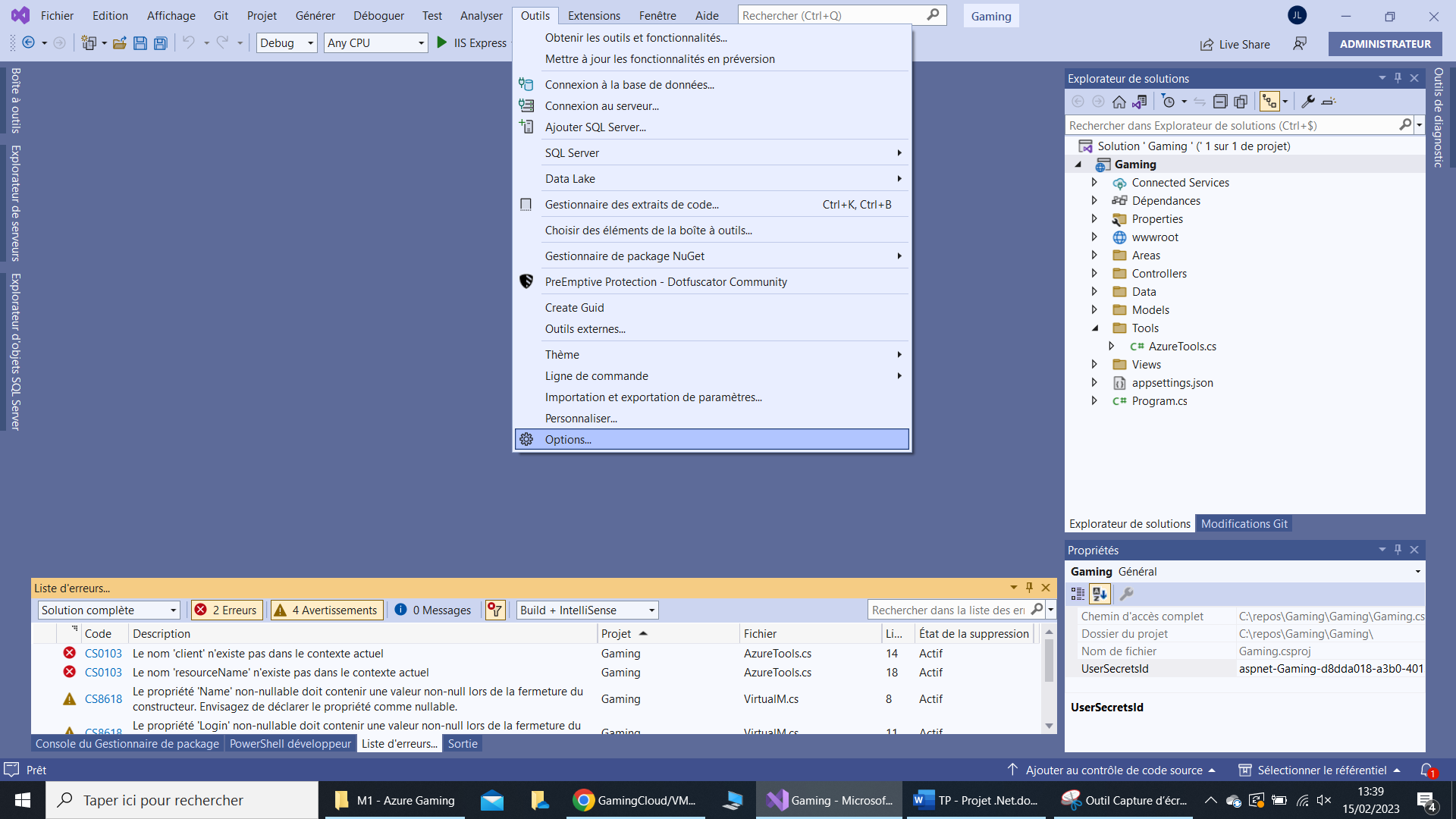


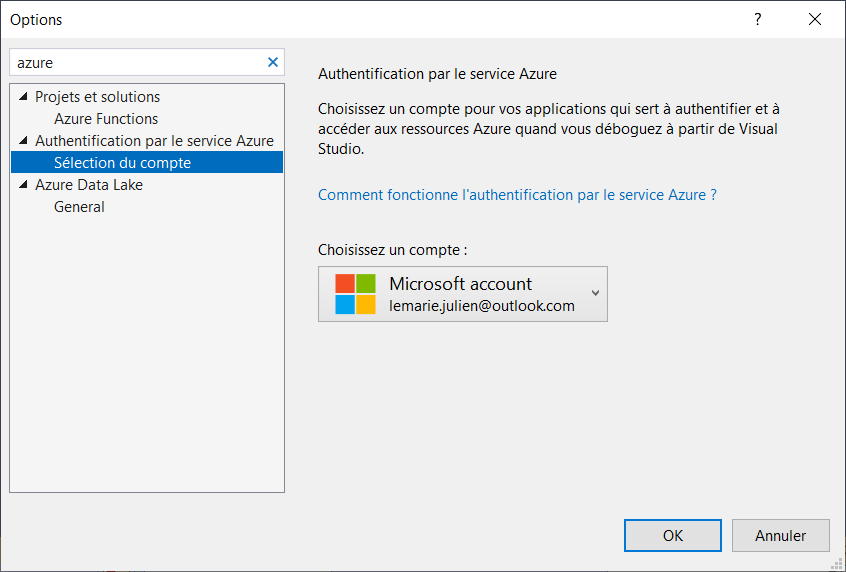
Cliquez sur « Apply Migrations » puis rafraichir la page.



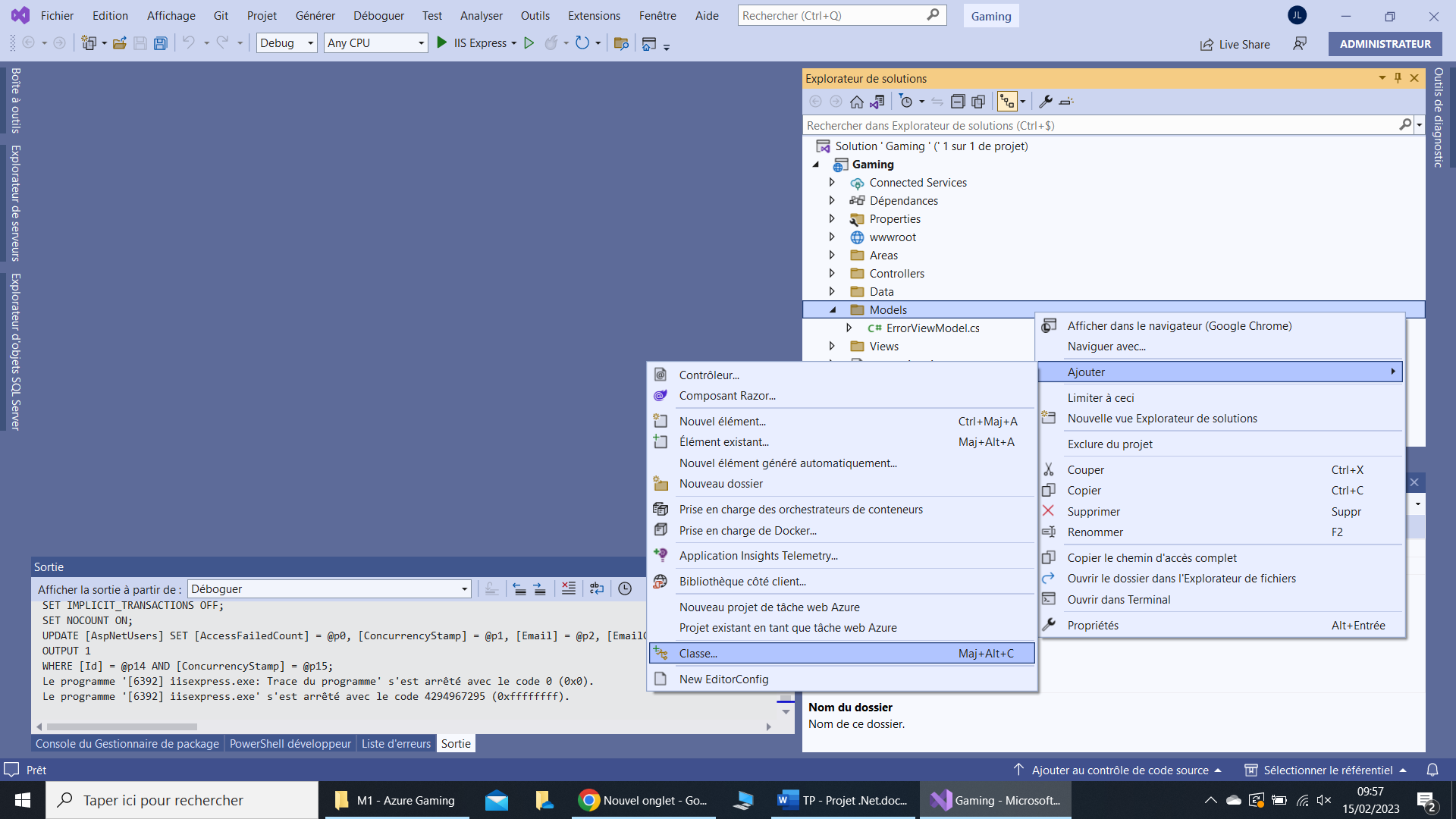
Cliquez sur le lien pour confirmer le compte.

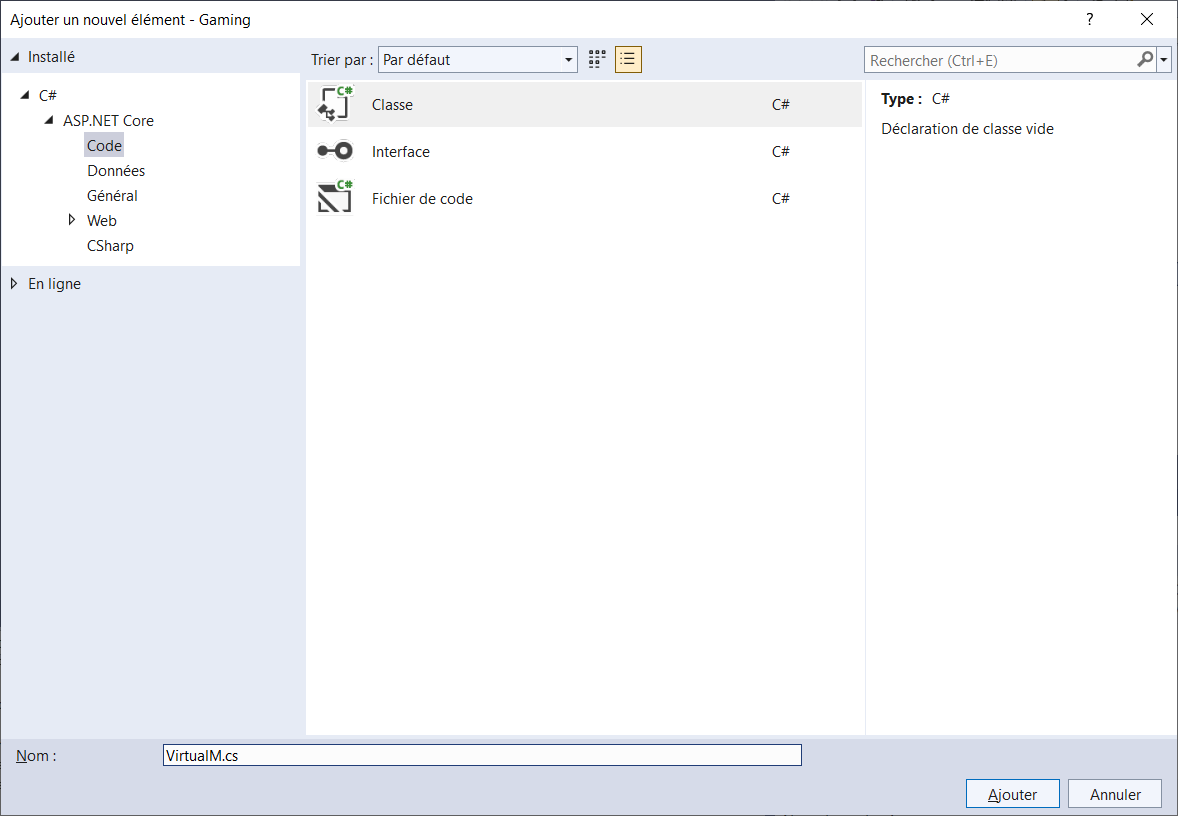
# Connexion de Visual Studio à Azure





# Création du modèle « VirtualM »





using System.ComponentModel.DataAnnotations;

namespace Gaming.Models

{

public class VirtualM

{

[Key]

public string Name { get; set; }

[Required]

public string Login { get; set; }

[Required]

[DataType(DataType.Password)]

public string Password { get; set; }

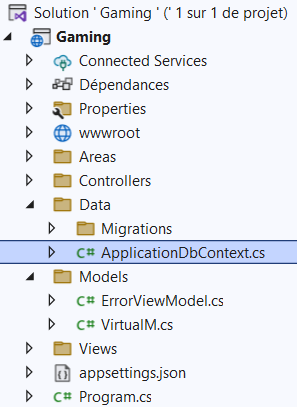
public string IP { get; set; }

public bool IsStarted { get; set; }

}

}

# Mise à jour du DbContext



using Gaming.Models;

using Microsoft.AspNetCore.Identity.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore;

namespace Gaming.Data

{

public class ApplicationDbContext : IdentityDbContext

{

public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)

: base(options)

{

}

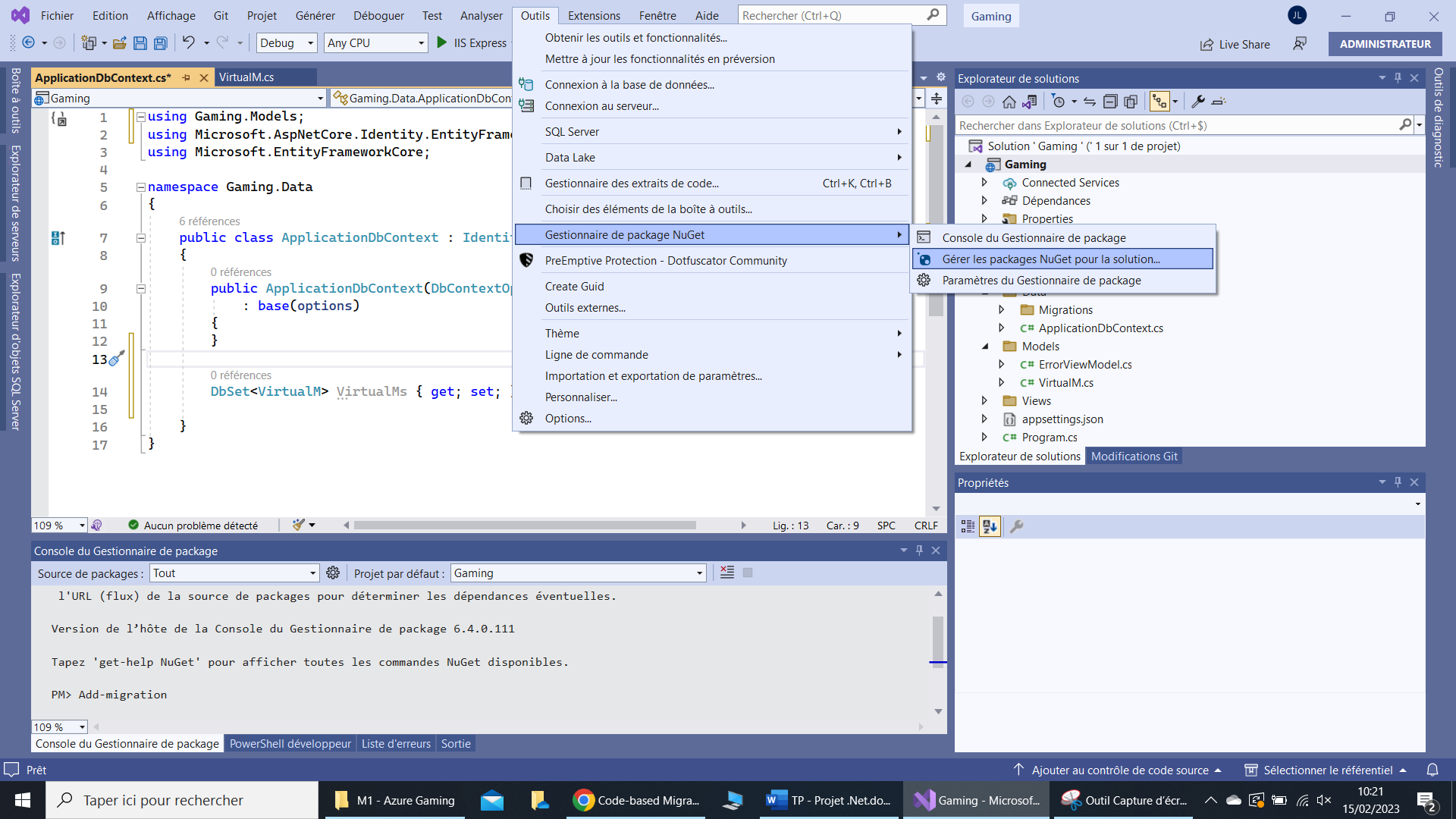
public DbSet<VirtualM> VirtualMs { get; set; }

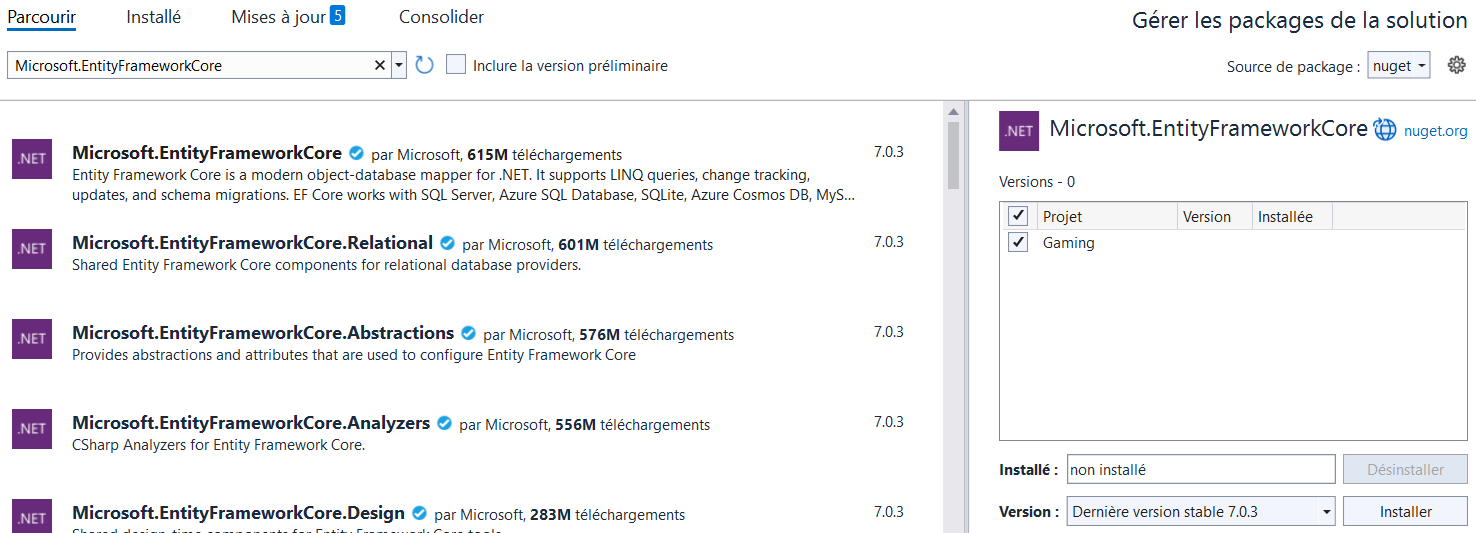
}

}

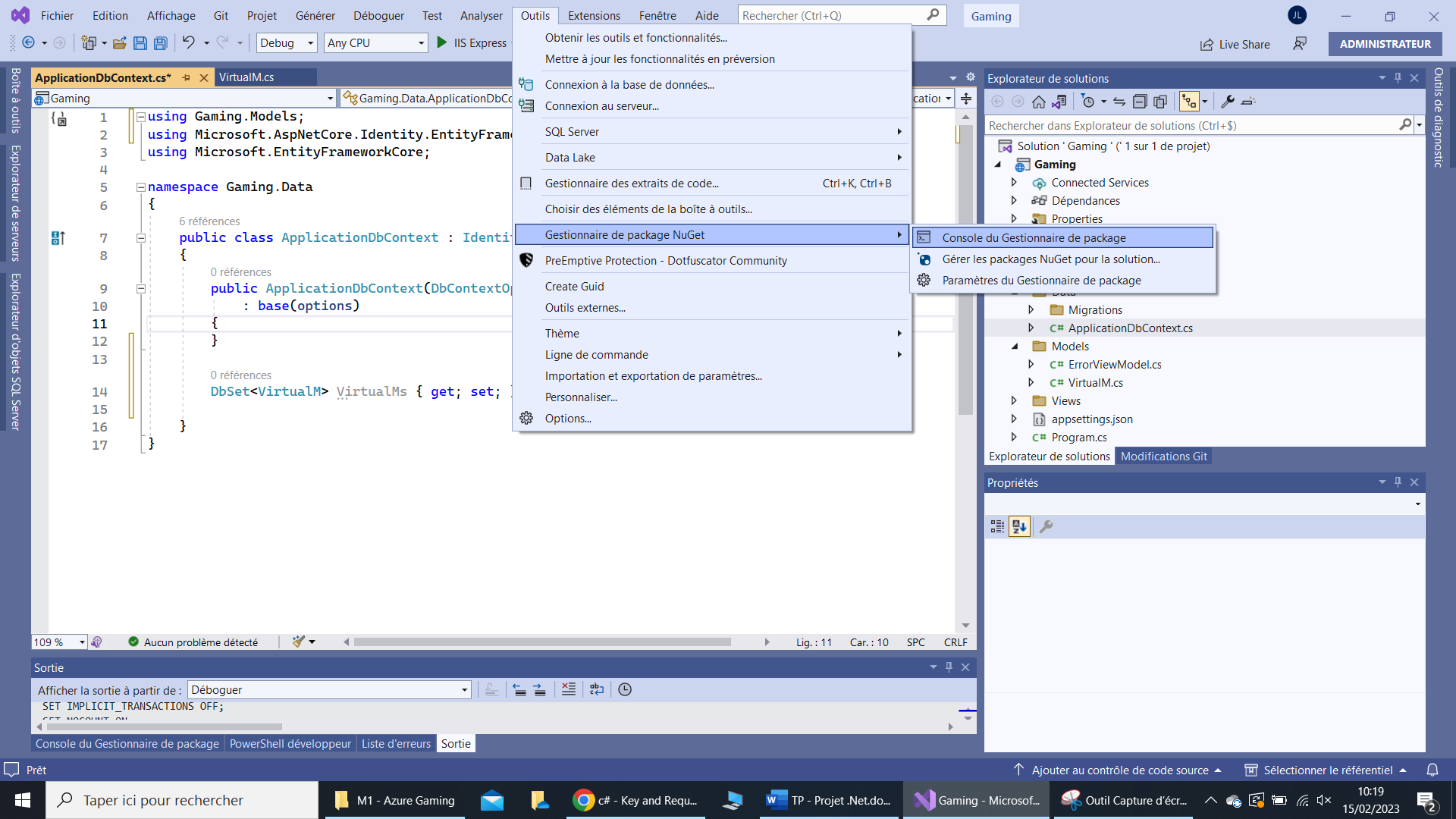
# Mise à jour de la base de données

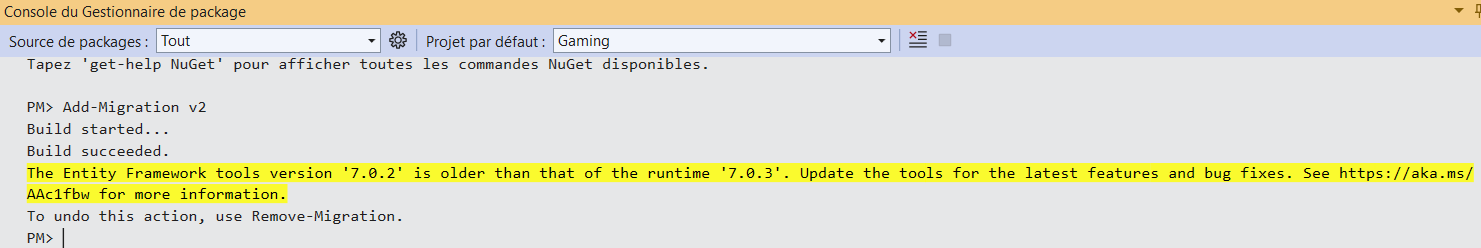
Ajout du package NuGet « Microsoft.EntityFrameworkCore » :



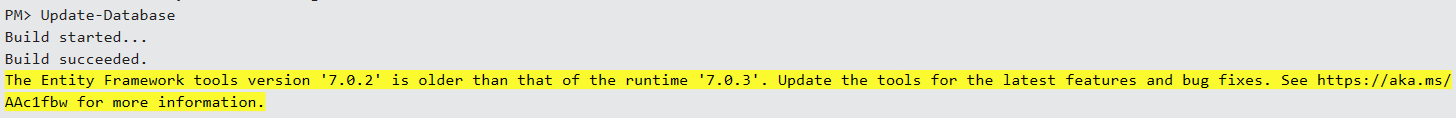


Exécution de la commande « Add-Migration v2 » :

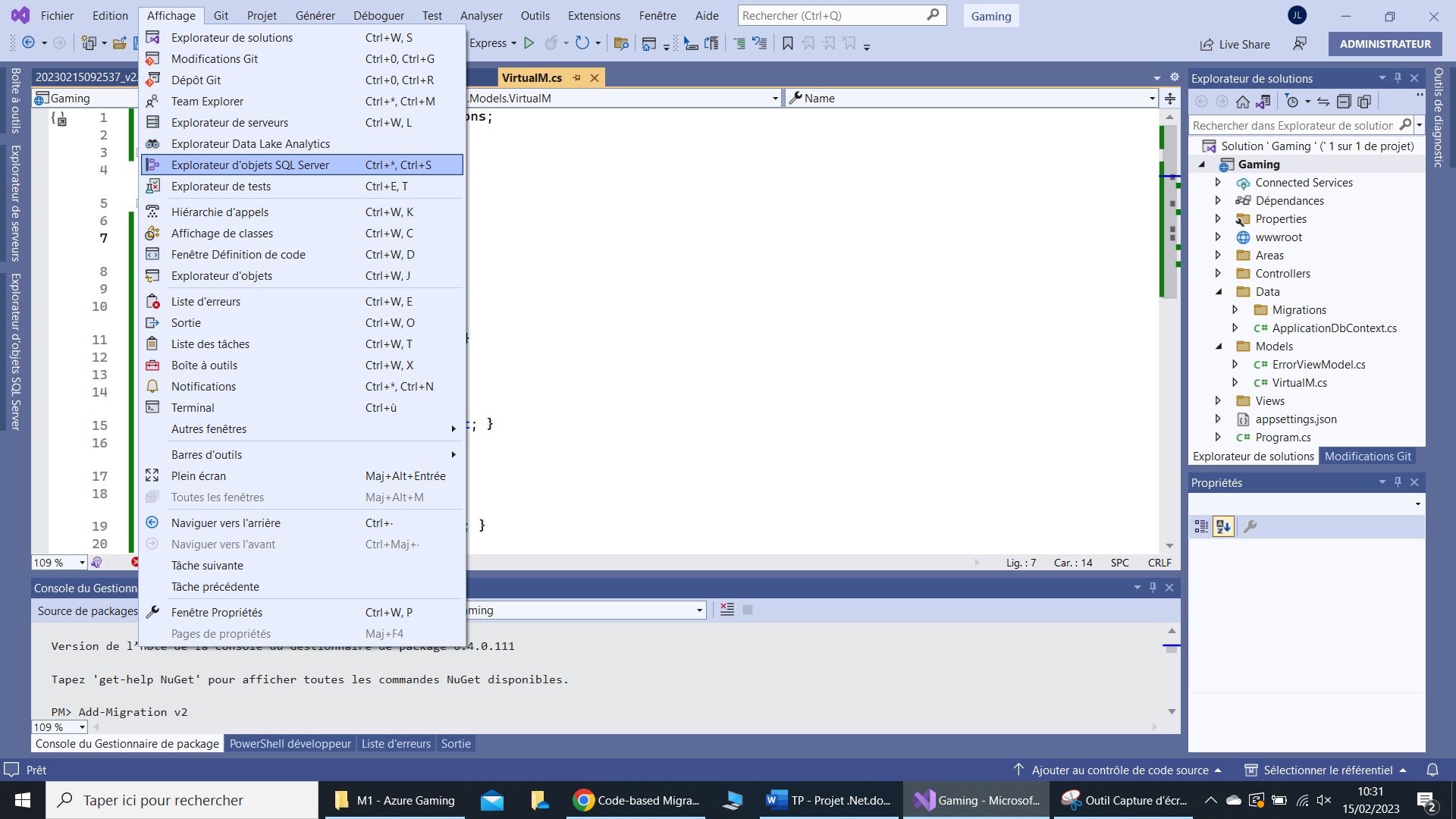
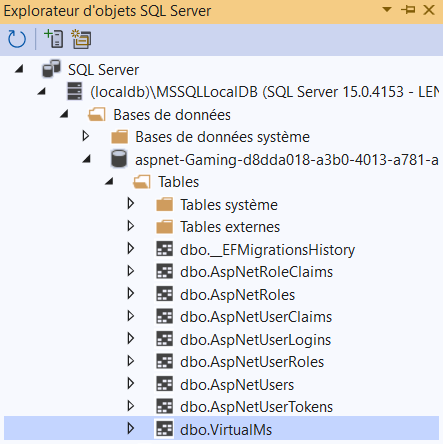




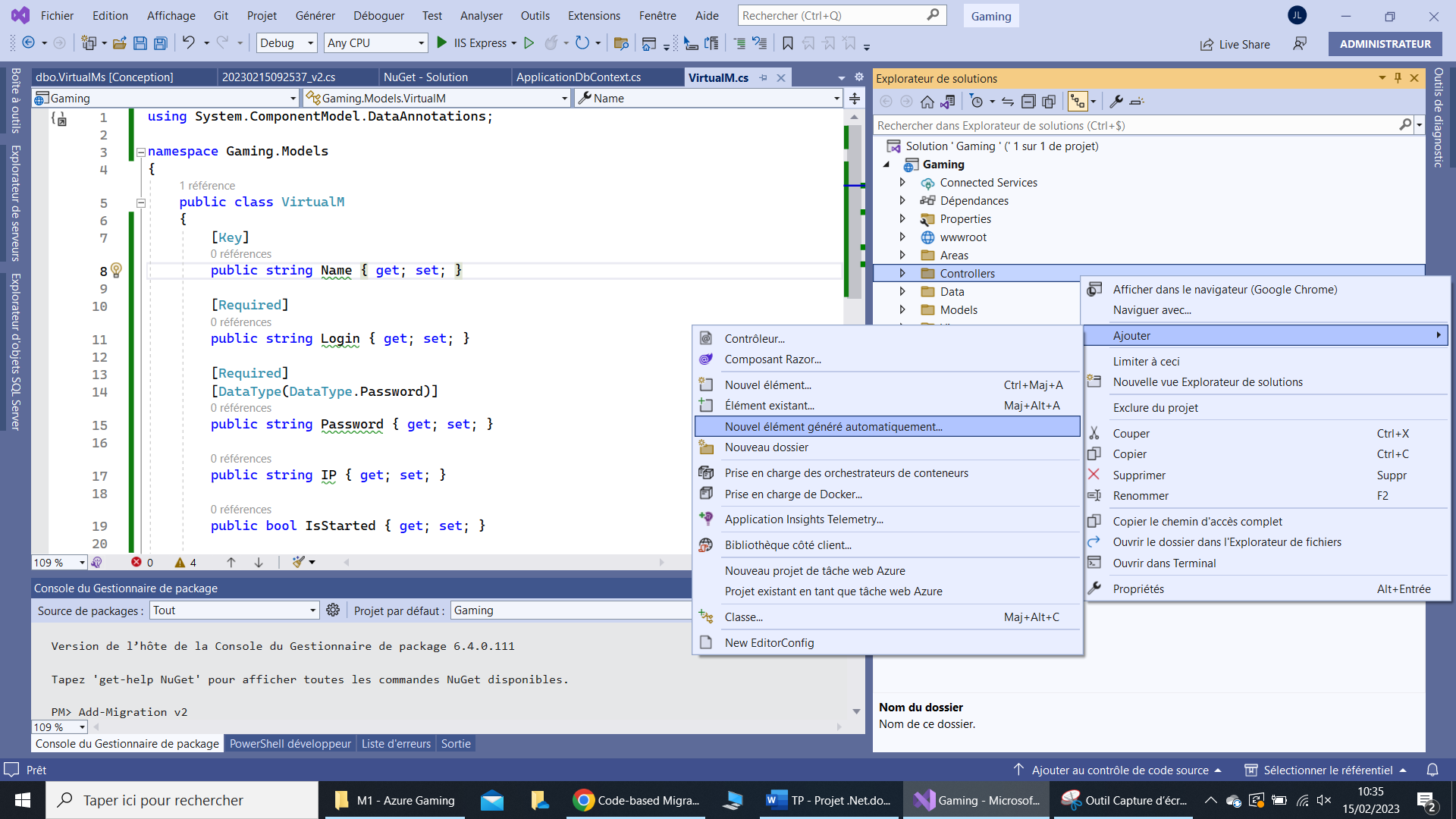
Exécution de la commande « Update-Database » :

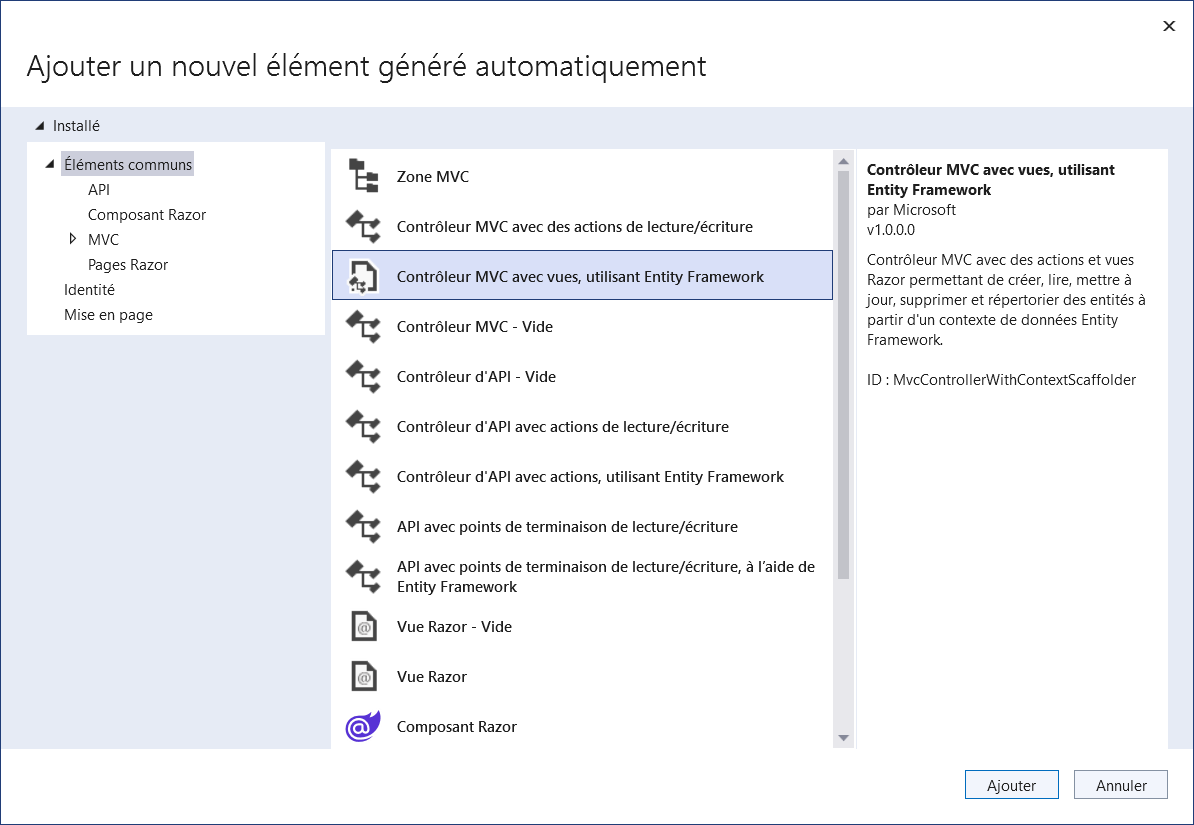


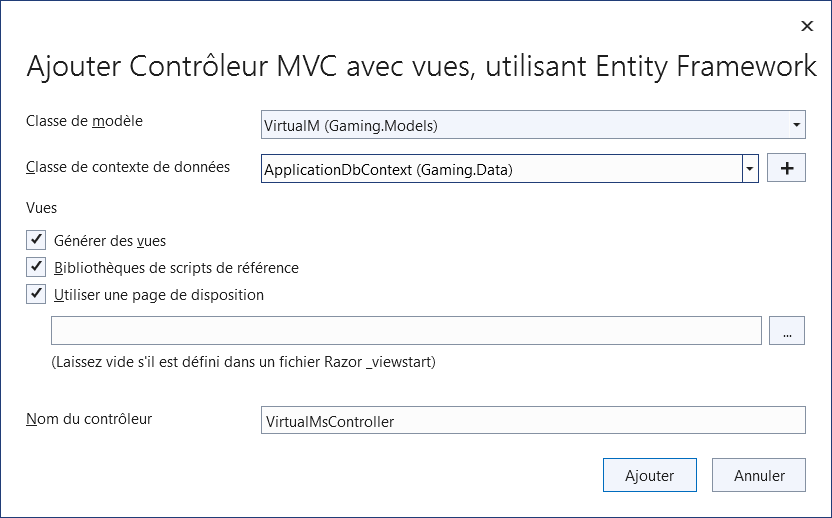
# Consultation de la base de données

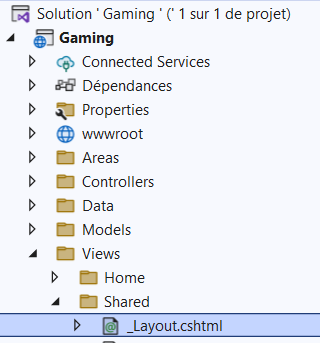
# Génération du controller et des views



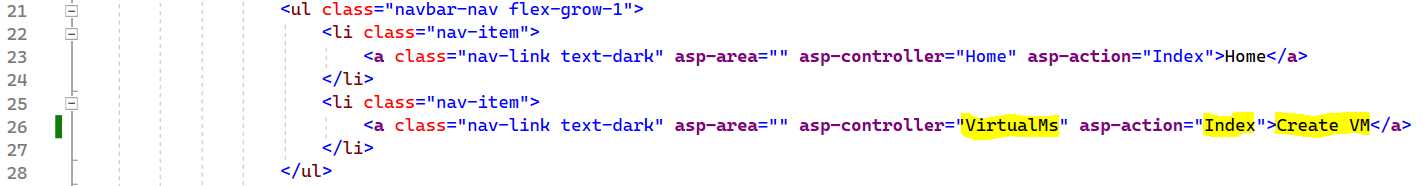




# Ajout d’un lien d’accès aux views de VirtualM

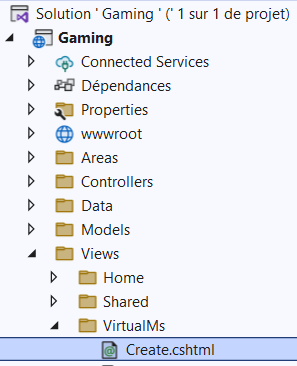


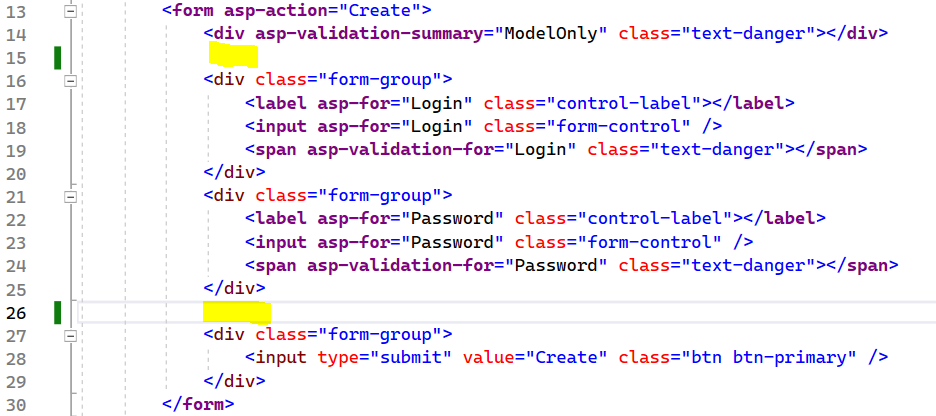
Remplacement du lien « Privacy » :



# Mise à jour de la vue de création d’une VM

Suppression dans la vue des propriétés « Name », « IP » et « IsStarted » :





Mise à jour de la méthode Create du Controller « VirtualMsController.cs » :

* Suppression des propriétés dans la signature de la méthode (partie « Bind »)
* Suppression des propriétés du ModelState afin de permettre sa validation (utilisation du « ModelState.Remove() »)
* Ajout de valeurs temporaires pour remplir l’objet « virtualM »

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("Login,Password")] VirtualM virtualM)

{

ModelState.Remove(nameof(virtualM.Name));

ModelState.Remove(nameof(virtualM.IP));

ModelState.Remove(nameof(virtualM.IsStarted));

if (ModelState.IsValid)

{

// Données temporaire en attendant la méthode de création d'une VM

virtualM.Name = "UserName";

virtualM.IP = "192.168.1.1";

virtualM.IsStarted = true;

\_context.Add(virtualM);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(virtualM);

}

# Création d’un username pour l’utilisateur connecté

Afin de pouvoir nommer les ressources Azure d’un utilisateur, nous allons créer une méthode privée dans le controller « VirtualMs » :

private string GetUserName()

{

string user = string.Empty;

if (HttpContext.User.Identity != null &&

HttpContext.User.Identity.Name != null)

{

user = HttpContext.User.Identity.Name;

user = user.Split("@")[0].Replace(".", "");

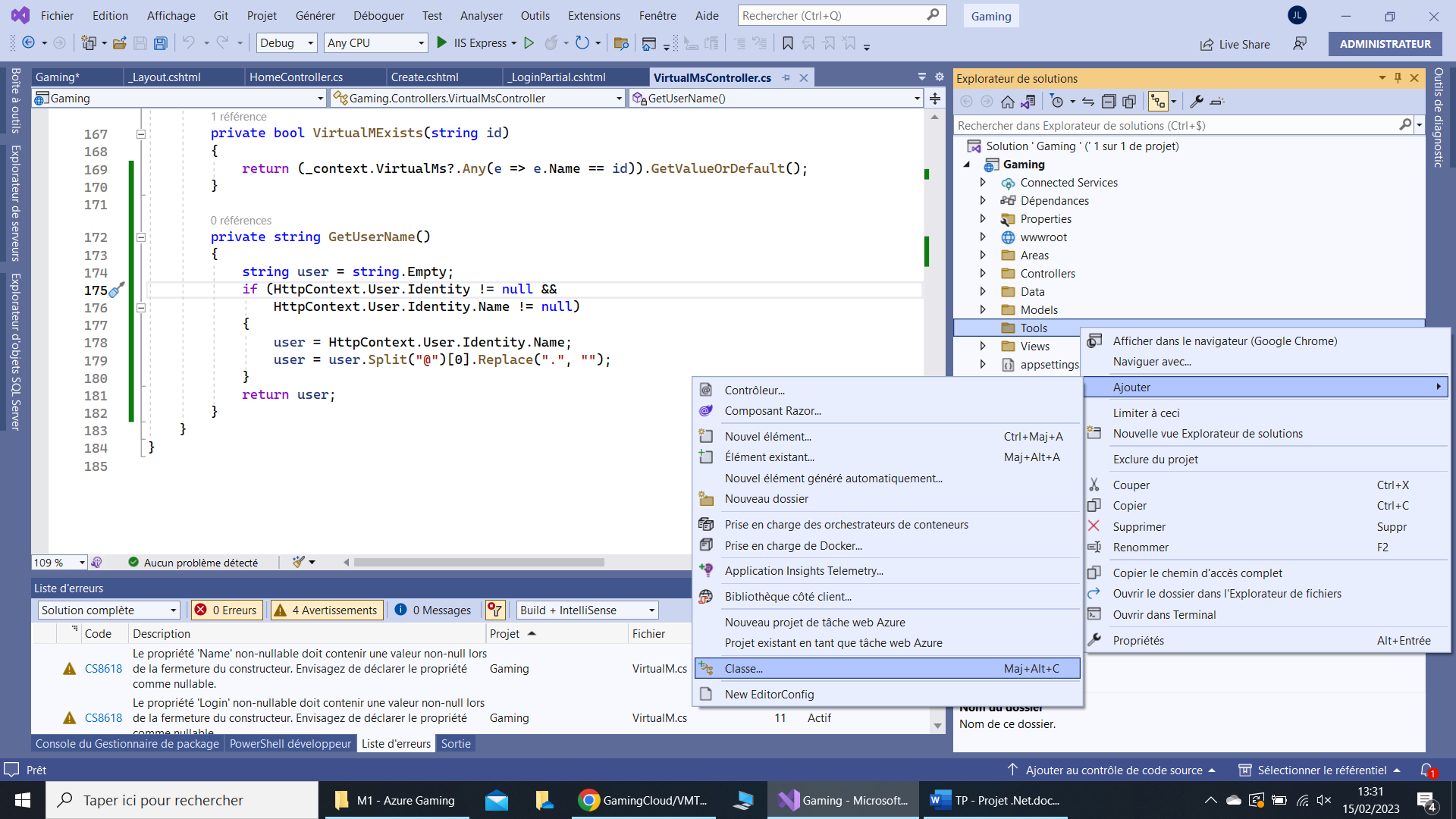
}

return user;

}

# Appels Azure

Création d’un dossier « Tools » dans lequel on créer une classe « AzureTools » :



Ajout des packages NuGet :

« Azure.Identity »

« Azure.ResourceManager »,

« Azure.ResourceManager.Network »

« Azure.ResourceManager.Compute »

