

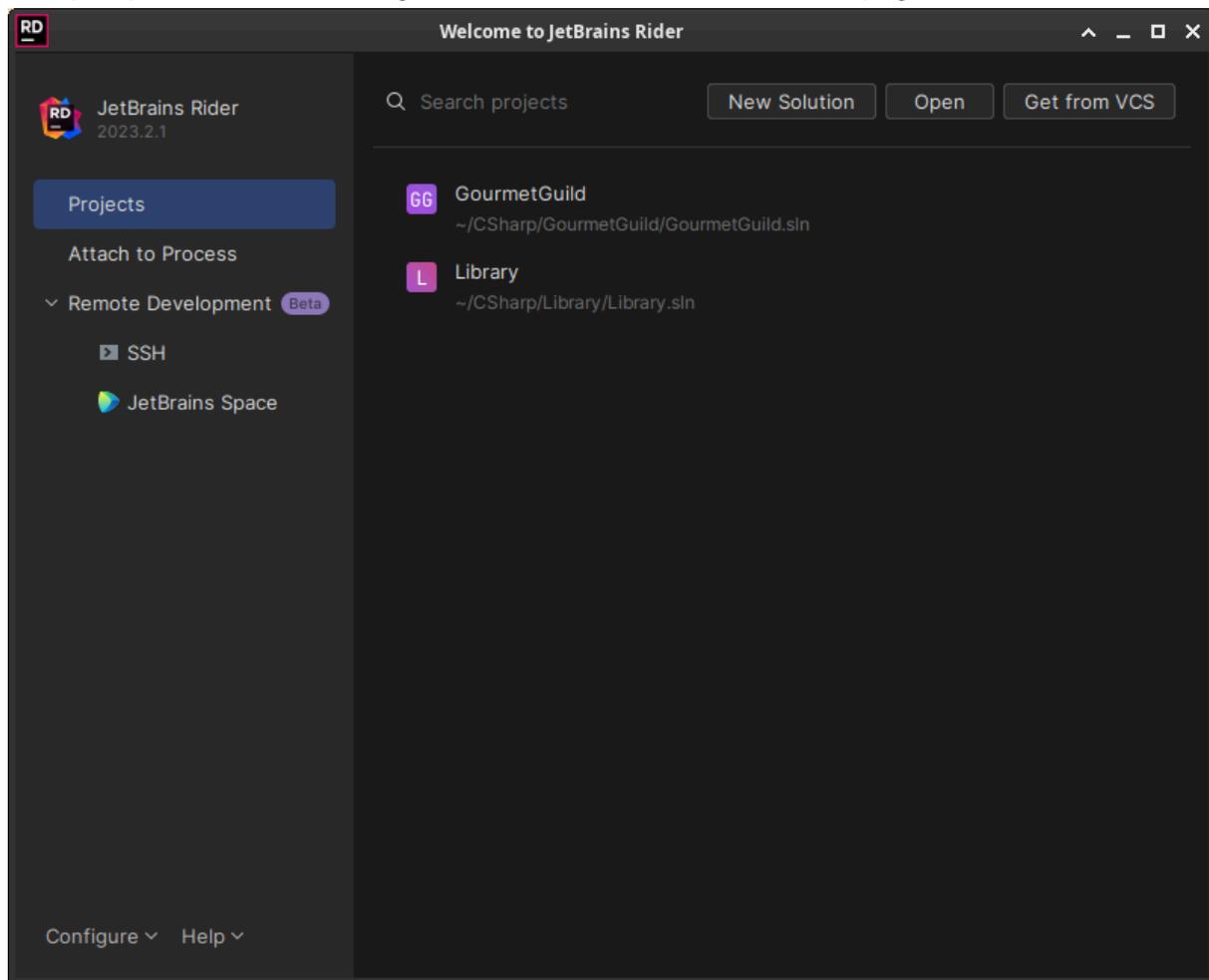
Mémento langage C#

Barillot
Esteban

Présentation :

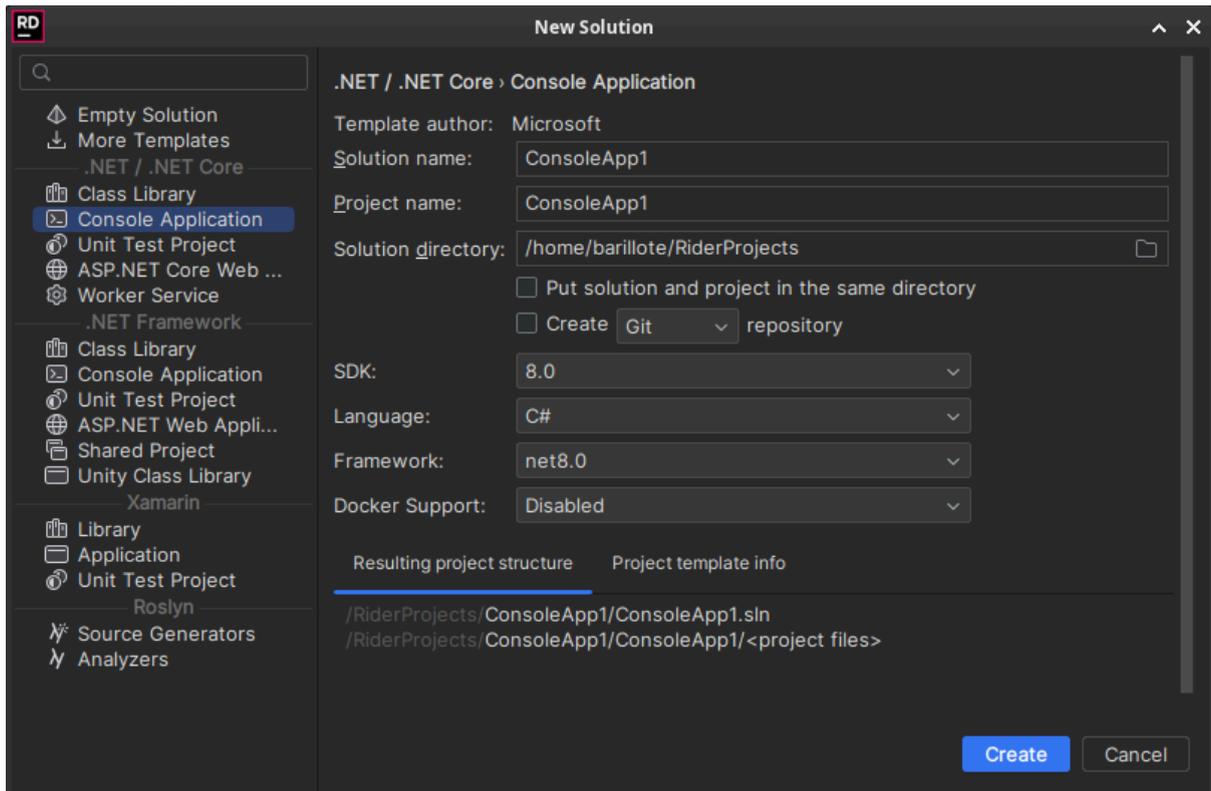
- I. Création d'un projet
 - A) Créer un nouveau projet sur Rider

Lorsque que vous ouvrez le logiciel Rider vous arriverez sur cette page ci-dessous



Cliquez sur le bouton "New Solution"

Puis cette page s'affichera



Il vous suffira de renseigner le nom de votre projet dans les cases “Solution name” et “Project name” et enfin cliquer sur le bouton bleu “Create”

B) Créer des migration sur Rider

Attention : bien avoir installer dotnet sur l'ordinateur en question où l'on souhaite faire le projet

Ajouter une migration :

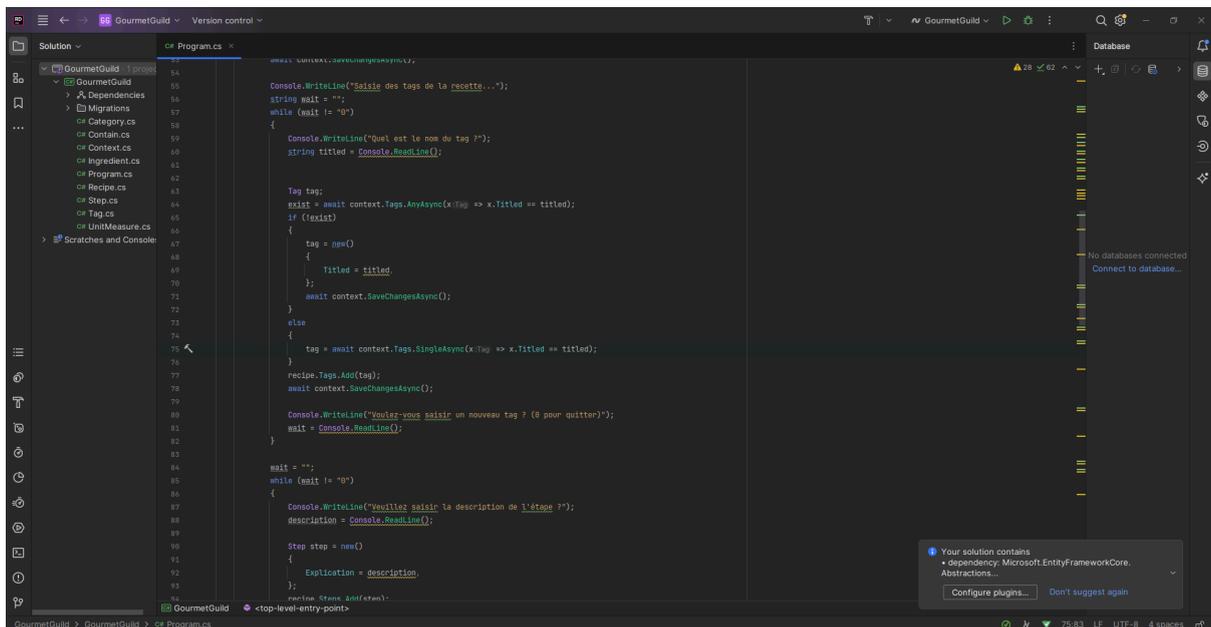
```
dotnet ef migrations add InitialDatabase --project NomProjet
```

Faire une update du projet :

```
dotnet ef database update --project NomProjet
```

C) Les dépendances importantes à avoir

Une fois après avoir pris en main le logiciel Rider vous pouvez cliquer sur le menu en haut de l'écran

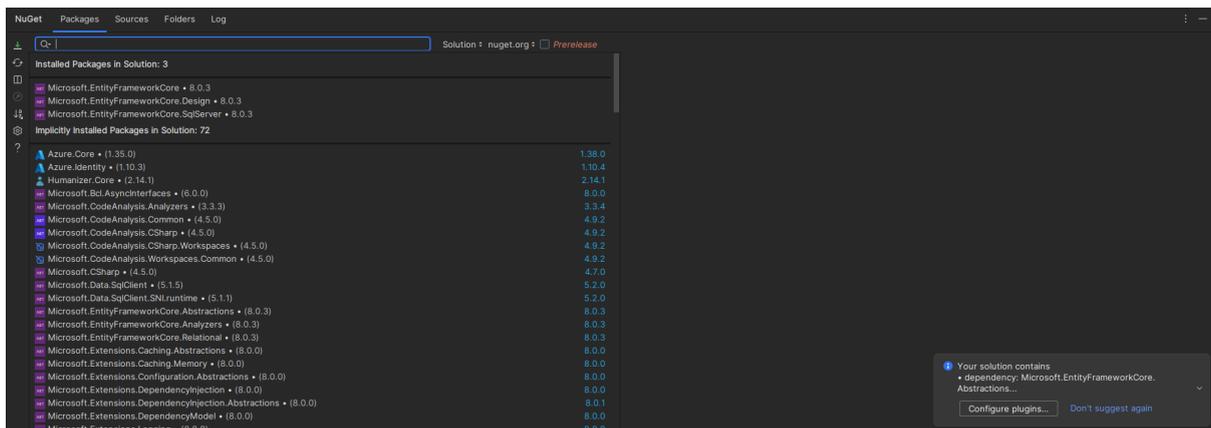


Puis cliquez sur "View" puis "Tool Windows" et cliquez sur "NuGet"



Une icône s'affiche alors dans la barre à gauche comme ceci, cliquez dessus

La page de NuGet s'affichera



Dans la barre de recherche vous pourrez alors faire vos recherches

Packages conseillés :
Microsoft.EntityFrameworkCore
Microsoft.EntityFrameworkCore.Design
Microsoft.EntityFrameworkCore.SqlServer

II. Les bases du Csharp

A) Ecrire dans la console

Ecrire un élément dans la console :

```
Console.WriteLine("Hello World");
```

Ecrire un élément dans la console avec des valeurs

```
Console.WriteLine("Hello World" + valeur);
```

Ecrire un élément dans la console avec des valeurs plus simplement

```
Console.WriteLine($"Hello World {valeur}");
```

B) Lire des éléments

Lire un élément de type 'string' dans la console :

```
string variable = Console.ReadLine();
```

Lire un autre type d'élément dans la console :

```
type variable = type.Parse(Console.ReadLine());
```

C) Ajouter des commentaires

Faire un commentaire sur une seule ligne :

```
// Commentaire
```

Faire un commentaire sur plusieurs lignes :

```
/*  
    Commentaire  
*/
```

III. Les types de variables

Préfixe	Type
int	entier
float	décimal à 4 bites
double	décimal à 8 bites
bool	booléen
char	caractère
string	chaîne de caractères

A) Les constantes

Une constante est un élément dont la valeur ne peut pas être modifiée une fois qu'elle a été définie. Les constantes sont déclarées à l'aide du mot-clé `const`. Voici quelques caractéristiques importantes des constantes en C# :

- **Valeur immuable:** Une fois qu'une constante est définie, sa valeur ne peut pas être modifiée pendant l'exécution du programme.
- **Détermination à la compilation:** La valeur d'une constante doit être déterminée au moment de la compilation et ne peut pas être dérivée à partir de l'exécution du programme.
- **Portée:** Les constantes ont une portée limitée au bloc dans lequel elles sont déclarées. Elles ne peuvent pas être locales à une méthode, elles doivent être déclarées au niveau de classe ou au niveau de structure.
- **Nommage conventionnel:** Par convention, les noms de constantes sont écrits en majuscules, avec des mots séparés par des underscores (`_`).

B) Les opérateurs arithmétiques (assignation)

Opérateur	Nom	Exemple
=		x = 5
+=		x += 3
--		x -= 3
*=		x *= 3
/=		x /= 3
%=		x %= 3
&=		x &= 3
=		x = 3
^=		x ^= 3
>>=		x >>= 3
<<=		x <<= 3

C) Les opérateurs de comparaison

Opérateur	Nom	Exemple
==	égal à	$x == y$
!=	différent de	$x != y$
>	supérieur à	$x > y$
<	inférieur à	$x < y$
>=	supérieur ou égal à	$x >= y$
<=	inférieur ou égal à	$x <= y$

IV. Les conditions

V. Les boucles

IV.

https://www.w3schools.com/cs/cs_operators_assignment.php