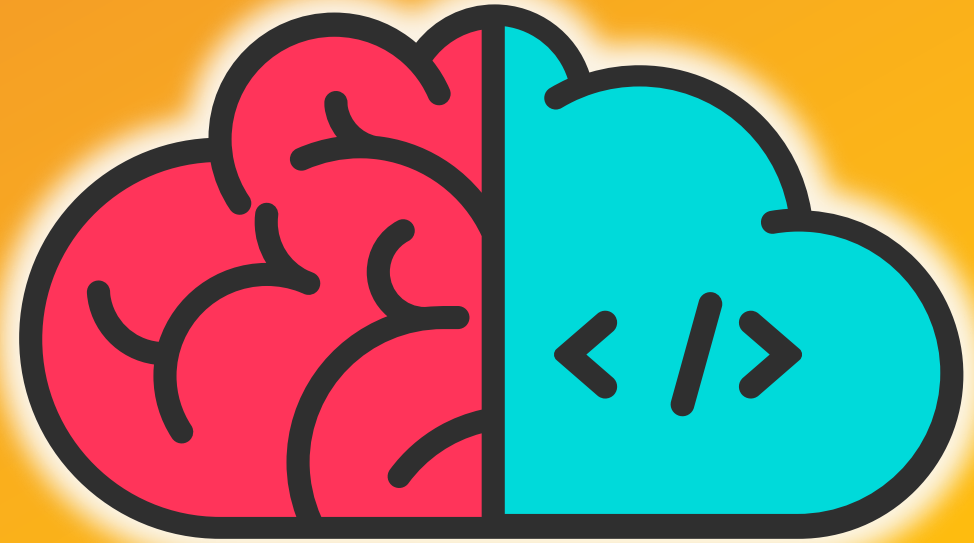


COURS C# / ASP.NET



Christophe MOMMER



<https://www.hts-learning.com>

PHOTOGRAPHIE ECOSYSTEME .NET

C# a été élu langage de l'année 2023 par Tiobe (<https://www.tiobe.com/tiobe-index>)

.NET est open-source et multiplateforme depuis 2016

Soutenu et propulsé par Microsoft (contrairement à .NET Framework qui est 100% interne)

ASP.NET arrive souvent très haut dans les benchmarks de TechEmpower(par exemple, 7 millions de requêtes par seconde sur plaintext)

C# et .NET permettent de réaliser tout type d'application : du web (site classique + SPA), du desktop (Windows bien sûr, mais aussi macOS et Linux avec des packages communautaires), des jeux vidéos (avec Unity ou Godot), des applications mobiles (Android & iOS), du machine learning avec ML.NET, etc.

C# VS JAVA

Syntaxe très proche de Java, conventions différentes (PascalCasing pour les méthodes)

Notion de propriétés : pas de getter/setter comme en Java, plutôt une propriété

```
public sealed class BlogPost
{
    public int Id { get; set; }
```

L'écriture des lambdas se fait avec le signe égal (=> au lieu de ->)

Il est possible d'utiliser le mot-clé « var » pour raccourcir la déclaration d'une variable (tout en restant fortement typé)

L'asynchronisme se fait avec les mots-clés « async » (niveau méthode) et « await » (niveau appel). Le type représentant une promesse est Task/Task<T>

PROJET

Pour apprendre le langage C#, nous allons développer une application console qui fera office de gestionnaire de mot de passe

L'application proposera un menu « user-friendly » et utilisera une base de données Sqlite de façon sous-jacente pour la persistance, avec l'utilisation de la librairie EF Core

Fonctionnellement, il faut pouvoir :

- Ajouter un mot de passe (qui a un nom et une valeur)
- Afficher un mot de passe (en donnant son nom)

Il est possible d'avoir 2 (ou +) mots de passe avec le même nom, dans ce cas, la date de mise à jour doit être affichée pour définir lequel afficher

Démonstration

ÉTAPE 1 : LE MENU

Le but est de proposer à l'utilisateur le menu d'accueil de 3 choix :

- 1) Ajouter un mot de passe
- 2) Consulter un mot de passe
- 3) Quitter

Pour le moment, on va se contenter d'afficher sur la console le choix fait par l'utilisateur
Le choix d'un élément de menu doit se faire avec les flèches directionnelles

La récupération d'une touche sur la console se fait avec `Console.ReadKey`
`Console.BackgroundColor` permet de définir la couleur de fond
`Console.ForegroundColor` permet de définir la couleur de la police

Attention à ne pas dépasser de la liste des choix possibles dans le menu

ÉTAPE 2 : EF CORE

L'application doit vérifier si la base de données existe et la créer si ce n'est pas le cas

Nous allons utiliser EF Core ainsi que le système de migrations à cet effet

Installer le package avec le connecteur Sqlite sur le projet :

➔ `dotnet add package Microsoft.EntityFrameworkCore.Sqlite`

- Créer une classe de modèle pour représenter un mdp : Id (Guid), Name (string) et Value (string)
- Créer une classe pour l'accès BDD qui hérite de DbContext
- Ajouter une propriété de type pour les mdp de type DbSet du modèle défini
- Overrider la méthode OnConfiguring pour appeler la méthode d'extension UseSqlite avec une chaine de connexion (Filename=pwds.db)

Installer l'outil EF :

➔ `dotnet tool install dotnet-ef`

ÉTAPE 3 : MIGRATIONS

Pour créer la BDD, il faut une migration initiale :

```
dotnet ef migrations add InitialMigration [-o RepertoireDeSortie]
```

Pour appliquer les migrations (EF crée la base si besoin) : appeler la méthode `MigrateAsync` sur la propriété `Database` de l'instance de votre classe de `DbContext`

ÉTAPE 4 : LE CHIFFREMENT

Pour chiffrer les mots de passe, il faut un mot de passe maître. Demander à l'utilisateur son mot de passe maître à l'ouverture de l'application pour l'avoir pour toutes les opérations

➔ Attention, la saisie de mots de passe est une opération sensible, il est préférable de remplacer le texte affiché par « ***** » (en faire une méthode pour la récupération du mdp maître mais aussi les autres). Astuce : explorer les signatures de méthode ReadKey, Attention au backspace pour supprimer le dernier caractère

Implémenter une classe d'aide pour chiffrer et déchiffrer du texte. Le mot de passe maître doit être encodé en SHA256 pour correspondre aux contraintes de la classe de chiffrement Aes

Chercher dans la documentation & sur le web une façon d'implémenter les deux méthodes

ÉTAPE 5 : INSERER/MAJ UN MDP

Implémenter l'élément de menu pour permettre à l'utilisateur d'insérer/mettre à jour un mdp en cas de choix dans le menu

- Récupérer la saisie de façon sécurisée grâce à la méthode faite précédemment + récupérer le nom du mdp à stocker
- Chiffrer la valeur grâce à la classe créée précédemment
- Utiliser l'instance de DbContext pour ajouter l'élément dans le DbSet. Attention : DbContext utilise le pattern Unif Of Work, il faut sauvegarder sa transaction
- Gérer le cas : un mdp avec le même nom existe déjà, demander à l'utilisateur s'il veut écraser la valeur (maj) ou en ajouter un autre

ÉTAPE 6 : GESTION DE LA DATE + LECTURE

Suite à l'implémentation de la slide précédente, il faut distinguer deux mdp qui ont le même nom :

- Faire évoluer le modèle pour prendre en compte la date de mise à jour et générer une nouvelle migration
- Mettre à jour le code pour définir la valeur de cette date

Tout est prêt pour la lecture, implémenter le code permettant à utilisateur de lire un mot de passe. On lui demandera le nom du mdp.

En cas de multiples occurrences, utiliser un menu similaire au menu d'accueil afin que l'utilisateur puisse choisir dans une liste le mdp à récupérer en fonction de la date

ÉTAPE 7 : SÉCURITÉ

Il faut implémenter une sécurité minimaliste pour empêcher l'accès à l'application si le mot de passe maître est incorrect.

Au premier lancement, sauvegarder le SHA256 du mot de passe maître dans un fichier.

A chaque lancement, comparer le SHA256 du mot de passe maître avec le contenu binaire du fichier

➔ Sentez-vous libre de rajouter d'autres sécurités si vous avez le temps, toute amélioration fonctionnelle est appréciable

Si vous souhaitez un cours « classique » :

https://www.youtube.com/channel/UCBu6vkOw_tMx_mWNKTpnv-w