

Cours n°9

Programme du cours

Activité	Durée
Vue d'ensemble sur l'architecture 3 Tiers	15m
Création des projets et installation des prérequis	20m
Création de la base de données + scaffold	15m
Création de l'UserService	30m
Création de l'interface de connexion, inscription et modification	1h
Hashing des mots de passe	30m
Message d'erreur	15m

Architecture 3 Tiers

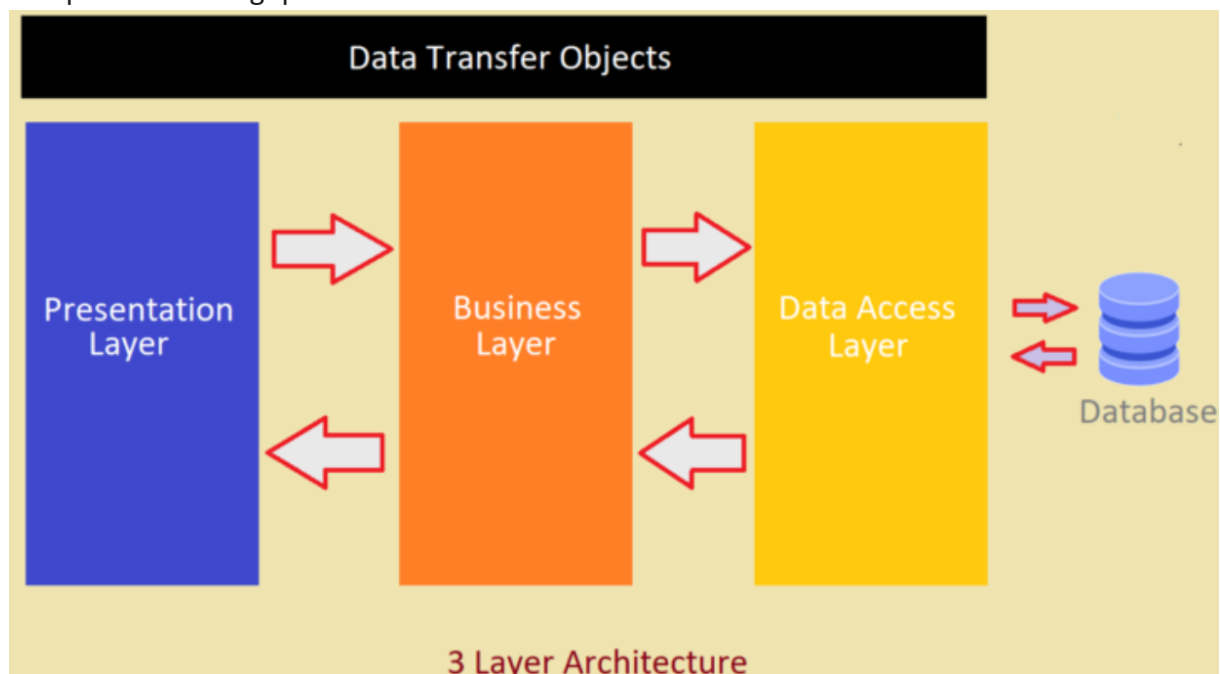
L'architecture logique du système est divisée en trois niveaux ou couches :

- La **présentation** des données, correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- Le **traitement** métier des données, correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- L'**accès aux données** persistantes : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Dans cette approche, les couches communiquent entre elles au travers d'un « modèle d'échange », et chacune d'entre elles propose un ensemble de services rendus. Les services d'une couche sont mis à disposition de la couche supérieure.

On s'interdit par conséquent qu'une couche invoque les services d'une couche plus basse que la couche immédiatement inférieure ou plus haute que la couche immédiatement supérieure (chaque couche ne communique qu'avec ses voisins immédiats).

Dans cette architecture logique du système, une couche Transverse appelée DTO (Data Transfer Object) est ajoutée pour faciliter la transition des entités d'une couche à l'autre. Cette couche est responsable de la conversion des données entre les couches et permet une communication transparente entre les différentes couches du système. Les DTO sont des objets qui transportent des données entre les différentes couches du système, sans comporter de la logique métier.



Installation

- Créer un projet nommé *Présentation* en app console
- Créer un projet nommé *BusinessLayer* en tant que librairie (bibliothèque de classes)
- Créer un projet nommé *DataLayer* en tant que app console
- Créer un projet nommé *DTO* en tant que librairie (bibliothèque de classes)
- Ajouter des références de projets de sorte à ce que la couche présentation référence la couche business et que la couche business référence la couche data
- La couche Business et la couche Présentation doivent référencer la couche DTO
- Installer le paquet NuGet Microsoft.EntityFrameworkCore.Design dans le projet DataLayer
- Installer Microsoft.EntityFrameworkCore.SqlServer dans le projet DataLayer
- Installer Microsoft.EntityFrameworkCore.Tools dans le projet DataLayer

Création de la base de données

Ajouter une base de données via l'outil de base de données intégré à Visual Studio.

Dans le projet *DataLayer*, ajouter une base de données nommé **Production**

Dans la base de données, ajouter une table User avec les champs **ID**, **Login** et **Password**

ID doit être une clé primaire avec l'auto-incrémente. **Login** et **Password** des champs *NVARCHAR(MAX)*. Aucune colonne n'est nullable.

Scaffold

Le scaffolding consiste à transformer une source de données en entités C#.

Pour lancer le scaffolding, il faut mettre la commande suivante dans la console de package Visual Studio (Outils/Gestionnaire de paquet). Veillez à bien sélectionner le projet *DataLayer* et le mettre en *startup project*

```
Scaffold-DbContext "Data  
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=<CheminVersLeMDF>;Integrated  
Security=True" Microsoft.EntityFrameworkCore.SqlServer -outputdir Models -contextdir Data
```

Refactorer ensuite le nom du **DbContext** en **ProductionDbContext**

UserService

Dans la *BusinessLayer* créer un fichier nommé **UserService**.

Il faut des méthodes pour gérer un utilisateur (*Ajouter, Modifier, Supprimer, Connecter*)

Interface console

Il faut ajouter la possibilité à l'utilisateur de pouvoir s'inscrire ou de se connecter

Une fois connecté l'utilisateur peut soit mettre à jour ces données, les supprimer.

Une fois l'utilisateur connecté, afficher *Bonjour [login]*

Attention il faut créer une classe user dans le projet DTO et passer cette classe à l'interface sans le password

```
C:\Users\Eric\Desktop\IDEC_Repo\Session 9\Exercices\Presentation\bin\Debug\net7.0\Presentation.exe
1 : Inscription
2 : Connexion
```

```
C:\Users\Eric\Desktop\IDEC_Repo\Session 9\Exercices\Presentation\bin\Debug\net7.0\Presentation.exe
Bonjour eric
1 : Mettre a jour mes données
2 : Supprimer mon compte
```

Hashing

Le hashing est une méthode qui permet de sécuriser le stockage des mots de passe dans la base de données. Il faut implémenter une méthode de hashing pour le champ **Password**

Message d'erreur

Empêcher l'ajout d'une personne si le login est déjà existant avec un message d'erreur