

Exercice Récapitulatif

Objectifs

1. Revoir la création d'un modèle à partir d'une base de données
2. Revoir les patterns Repositories et UnitOfWork
3. Voir la programmation Multitask
4. Revoir la création d'une Web API avec Controller
5. Revoir Swagger

Exercice 1 – Northwind_API

Le but de cet exercice est de développer toutes les couches nécessaires afin d'exposer la base de données Northwind. Il s'agit d'un bon exercice récapitulatif des concepts vus jusqu'ici. On va également utiliser la programmation multithread via les `async/await`.

1. Créer un projet ASP.NET Core Web API (avec Controllers) que vous nommerez Northwind_API
2. Ajouter dans ce projet un répertoire Entities et générer les entités à partir de la base de données Northwind comme vu précédemment
3. Implémenter le pattern Repository et UnitOfWork
 - a. Limiter votre implémentation à la table Employees et Orders
 - b. Les repositories seront adaptés pour utiliser la programmation multitâche. Voici l'interface IRepository adaptée à cette programmation

```
public interface IRepository<T>
{
    Task InsertAsync(T entity);
    Task DeleteAsync(T entity);
    Task<IList<T>> SearchForAsync(Expression<Func<T, bool>> predicate);
    // save entity, test via predicate if entity exists
    Task<bool?> SaveAsync(T entity, Expression<Func<T, bool>> predicate);
    Task<IList<T>> GetAllAsync();
    Task<T?> GetByIdAsync(int id);
}
```

4. Implémenter deux controllers (un pour Employee, un autre pour Order) qui permettront d'effectuer les opérations CRUD
 - a. N'oubliez pas que vous devez créer des modèles DTO et ne pas utiliser directement les Entities comme modèle de transfert.
 - b. Commencez par le controller Employee
 - c. Faites le controller Order s'il vous reste du temps