Laboratoire: Async et Tests Models

# Objectif(s)

* Convertir les méthodes Action des controllers et Repositories en asynchrones
* Ajouter des tests unitaires en TDD

# Créer le repo pour les laboratoires individuels

## Création de la branche de la séance

1. Dans Gitkraken, clonez votre repository de laboratoire
2. Créez la branche de la séance **S06\_Async\_TDD**
3. *Comment-commit-push*
4. **Répartir le travail entre les coéquipiers :**

# Méthodes et action Async

## Convertir les Repositories

1. Copiez l’Interface et la classe Repository générique afin que les méthodes soient async
2. Nommez les éléments en ajoutant **async** à la fin

**Exemple :**

public async Task AddAsync(T entity)

{

await dbSet.AddAsync(entity);

}

public async Task<T> GetAsync(int id)

{

return await dbSet.FindAsync(id);

}

1. Coipez les Interfaces et les classes des Repositories spécifiques afin que les méthodes soient async

## Convertir les controllers

1. Modifiez les procédures1Action des controllers
2. Modifiez les appels aux repositories afin qu’ils appellent les repositories **async**

**Exemple :**

public async Task<IActionResult> Index()

{

IEnumerable<Book> objList = await \_unitOfWork.Book.GetAllAsync(includeProperties:"Publisher,Subject");

return View(objList);

}

1. Modifiez les procédures1Action des controllers qui appelent des ViewModels

**Exemple :**

//GET CREATE

public async Task<IActionResult> Create()

{

IEnumerable<Subject> SubList = await \_unitOfWork.Subject.GetAllAsync();

IEnumerable<Publisher> PubList = await \_unitOfWork.Publisher.GetAllAsync();

BookVM bookVM = new BookVM()

{

Book = new Book(),

SubjectList = SubList.Select(i => new SelectListItem

{

Text = i.Name,

Value = i.Id.ToString()

….

1. Comment-Commit-Push

# Tests Unitaires sur les Models

## Créer le projet de tests pour les models

1. Ajoutez un projet de tests de type Nunit appelé **ZombieParty\_Models\_Tests**
2. Modifiez le namespace par défaut du projet test pour **ZombieParty\_Models**
3. Ajoutez une référence de projet du projet Models au Projet de tests

## Tests pour ZombieType

1. Faites la liste des scénarios de tests pour la création d'un ZombieType
   * Scénario avec un Name correct soit
2. Créez un test pour chaque scénario ou un test avec plusieurs TestCase selon les valeurs à tester