

Exercice 04 – Web API

Objectif

Comprendre l'intérêt de l'utilisation du **mocking** dans le cadre de tests unitaires d'une classe réalisant des manipulations de données, comme par exemple une **Web API** servant à accéder à un **CRUD basique** portant sur plusieurs modèles et usant du **Repository Pattern**.

Sujet

En vous basant sur le projet de **Web API** fourni, vous devrez réaliser, sans modifier le code fourni, une série de tests unitaires en usant du framework **NUnit** visant à tester le fonctionnement d'un ou des contrôleurs de l'API Web. Pour ce faire, vous devrez également faire en sorte de réaliser ces tests **sans modifier la base de données**, en vous servant justement du **mocking** pour en mocker le fonctionnement.

Au niveau des tests, vous devrez vérifier les retours possibles du ou des contrôleurs en vous basant sous le code de statut de la réponse (pensez à utiliser l'une des méthodes de la classe **Is**) Par exemple, dans le cadre du contrôleur des adresses, vous devrez réaliser les tests unitaires suivant :

- Pour l'**Ajout**
 - Vérifier que lors de l'ajout en BdD, le type de retour est bien un **200**
 - Vérifier que lors de l'ajout d'un élément ne respectant pas les contraintes, on a un type de retour de valeur **400**
 - Vérifier que lors d'une impossibilité d'ajout en BdD le type de retour est un **400**
- Pour la **Récupération**
 - Vérifier que lors de la récupération d'un ou de plusieurs éléments présent en BdD, le type de retour est bien un code de statut **200**
 - Vérifier que lors de la récupération de valeurs manquantes en BdD, le type de retour est un **404**.
- Pour la **Modification**
 - Vérifier que lors de l'édition d'une valeur en BdD effective, on a bien un code de statut de valeur **200**
 - Vérifier que lors de l'édition, en cas d'impossibilité de faire cette modification en BdD, on a un code erreur **400**
- Pour la **Suppression**
 - Vérifier que lors d'une suppression correctement effectuée, on obtient un résultat de type code **200**
 - Vérifier que si la suppression en BdD est impossible, alors on obtient un code d'erreur de type **400**

```
[Route("api/[controller]")]
[ApiController]
10 références
public class AddressesController : ControllerBase
{
    private readonly ILogger<AddressesController> _logger;
    private readonly IRepository<Address> _addressRepository;

    1 référence
    public AddressesController(ILogger<AddressesController> logger, IRepository<Address> addressRepository)
    {
        _logger = logger;
        _addressRepository = addressRepository;
    }

    [HttpPost("/address")]
    3 références
    public IActionResult AddAddress(Address newAddress)...

    [HttpGet("/addresses")]
    2 références
    public IActionResult GetAllAddresses()...

    [HttpGet("/address/{id}")]
    2 références
    public IActionResult GetAddress(int id)...

    [HttpPatch("/address/{id}")]
    3 références
    public IActionResult UpdateAddress(int id, Address newValues)...

    [HttpDelete("/address/{id}")]
    3 références
    public IActionResult DeleteAddress(int id)...
```