

RIL 2020 - MODULE ASP.NET CORE

MISE EN PRATIQUE : BUSINESS LAYER & TESTS UNITAIRES

Objectif : En reprenant l'exercice précédent sur les ORM, finaliser la couche métier des applications présentes dans la formation en omettant pas les tests unitaires

Durée : 3h30

Exercice :

1. Récupération de la solution prévue pour l'exercice,
 - a. Vérifier que la solution passe la compilation
 - b. Vérifier l'ensemble des fichiers de configurations au niveau des chaines de connexion
 - c. Vérifier que les tests unitaires passent à l'exécution
2. Sur le modèle de ce qui a été réalisé dans Cesi.Web.Mvc vous devez mettre en place l'utilisation de la couche métier dans le projet Cesi.Web.Pages
3. Créer l'ensemble des classes et interfaces nécessaires pour gérer nos amis les animaux en terme « métier »
 - a. Les tests unitaires sont obligatoires sur le périmètre des méthodes de l'exercice précédent
 - b. Les tests unitaires peuvent être faits soit avec un mock, soit avec une base de données « en mémoire » (au choix).
 - c. La notion de CRUD est également obligatoire pour le service des animaux
 - d. La méthode GetAnimalByOwner a fait l'objet d'une analyse et doit être supprimée (tests inclus).
4. Mettre à jour le projet Cesi.Web.Api qui utilisera le service métier gérant les animaux
 - a. L'objet View model représentant les animaux ne devra comporter que les champs suivants :
 - i. Id
 - ii. Name
 - iii. Species
 - iv. Owner
 - b. La méthode GetAnimalByOwner ayant été supprimée de l'ensemble de l'application, elle est remplacée dans le projet Cesi.Web.Api par GetAnimalsBySpecies sur la route **species/<nom_espèce>**
5. **BONUS :** Toutes les applications Web à jour avec les services créés pour Person et Animal (Nécessite la création des CRUD manquants).

RAPPEL :

1. Apporter un soin particulier au code fourni (convention, mise en forme, etc...)
2. N'oubliez pas de tester le code introduit et/ou retiré