

Exercice : De Simples Recettes

1 Enoncé

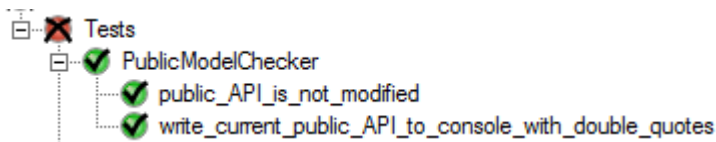
L'objectif de cet exercice est d'implémenter un petit modèle de recettes et d'ingrédients spécifié sous la forme de tests unitaires. Autrement dit : l'objectif de l'exercice est de faire passer en vert un jeu de tests.

Vous disposez pour cela d'une solution comprenant 2 projets :

- ITI.SimpleRecipes.Tests, contient les tests unitaires à valider
- ITI.SimpleRecipes, une implémentation minimale du modèle (le code écrit ne permet rien de plus que de pouvoir compiler sans erreur)

Pour faire tourner les tests unitaires il vous suffit de configurer le projet ITI. SimpleRecipes.Tests en tant que projet de démarrage puis d'exécuter la solution.

Comme vous pouvez le constater, pour le moment, tous les tests sont rouges, à l'exception de 2 d'entre eux :



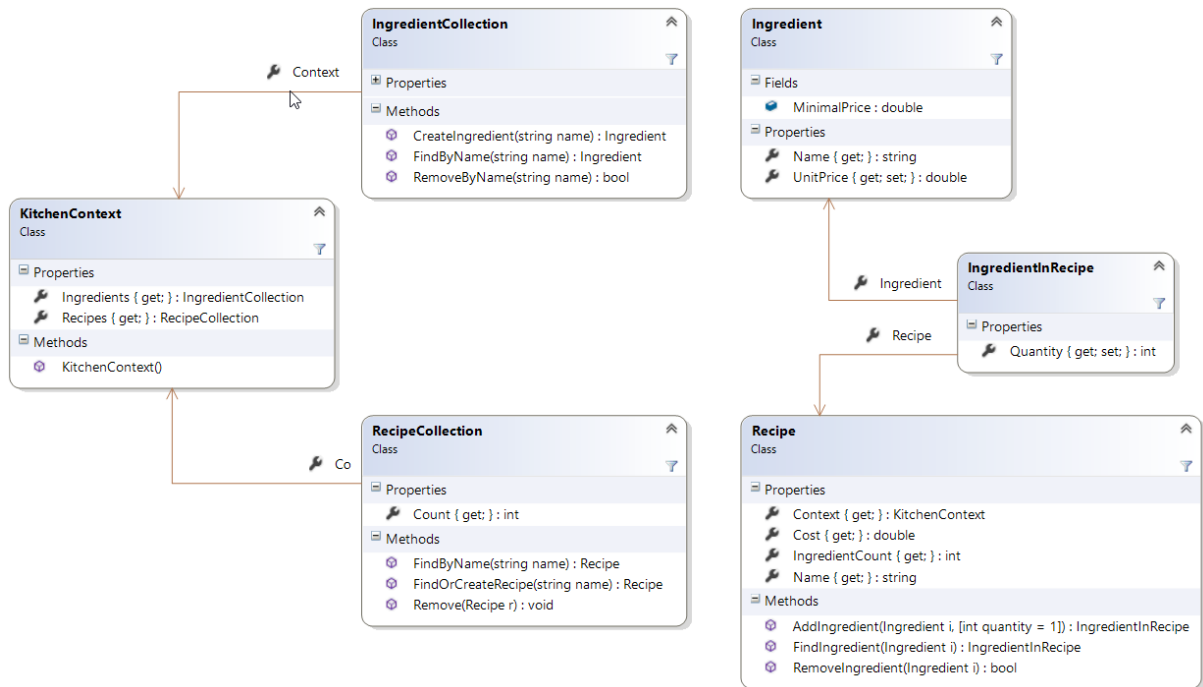
Ces deux tests analysent les objets implémentés dans ITI.SimpleRecipes.dll et en extraient l'API public sous forme XML. Le deuxième se contente d'afficher l'XML dans la console, le premier, plus intéressant, compare l'API public avec celle d'origine qui est mémorisée : si d'aventure, vous altérez l'API public des objets (en ajoutant une méthode, une propriété ou un champ public), il passe au rouge : vous devez vous assurer qu'il reste au vert, votre implémentation ne doit pas modifier l'API publique !

Pour les autres tests, à vous de faire en sorte qu'ils passent en vert. Pour cela, vous avez le droit de faire ce que bon vous semble dans le projet ITI. SimpleRecipes (modifier l'implémentation des méthodes, des propriétés ou des constructeurs, ajouter des membres *internal* ou *private*, créer de nouveaux types, etc.) En revanche, vous ne devez pas modifier le projet ITI.SimpleRecipes.Tests.

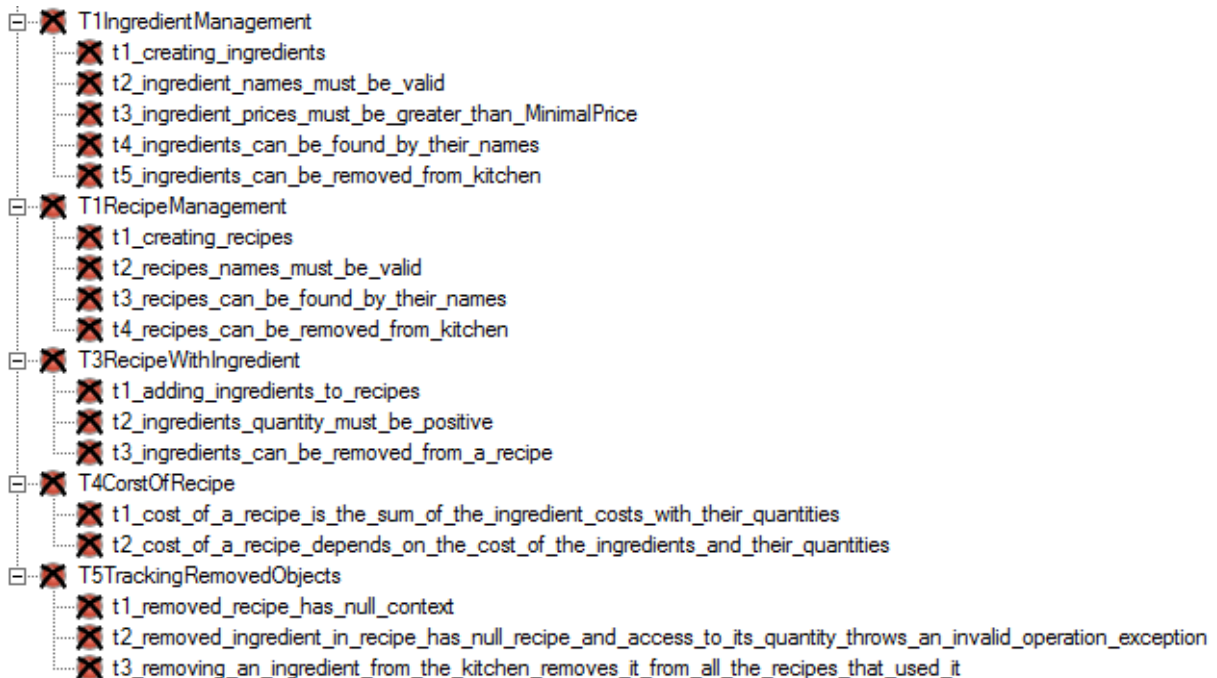
Les tests unitaires sont là pour spécifier de façon détaillée les fonctionnalités attendues. Cependant voici une courte description de ce qui est attendu :

- On peut créer un KitchenContext. Les KitchenContext sont indépendantes les unes des autres.
- On peut créer et supprimer des ingrédients (Ingredient) dans un KitchenContext.
 - Les ingrédients ont un nom (Name), qui ne change jamais, et un prix unitaire (UnitPrice) qui lui peut être modifié. Le prix unitaire ne peut être en-dessous de sa valeur initiale qui est fixée dans le champ static readonly Ingredient.MinimalPrice.
 - On peut retrouver un ingrédient par son nom.
- On peut créer/supprimer des recettes (Recipe).
 - Chaque recette à un nom unique.
 - On peut retrouver une recette par son nom.

- On peut ajouter des ingrédients à une recette et en changer la quantité (par défaut la quantité est à un).
- Chaque recette a un coût, calculé en fonction du coût de ses ingrédients et de leur quantité dans la recette.



Il y a 18 tests à valider :



Au boulot !