

Exercice : Les trains

# Enoncé

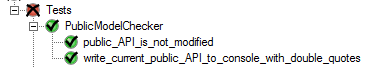
L'objectif de cet exercice est d'implémenter un petit modèle de gestion d’une ville et de ses trains spécifié sous la forme de tests unitaires. Autrement dit : l'objectif de l'exercice est de faire passer en vert un jeu de tests.

Vous disposez pour cela d'une solution comprenant 3 projets :

* ITI.Train.Tests, contient les tests unitaires
* ITI.Train.Model, un modèle à implémenter, constitué exclusivement d’abstractions
* ITI.Train.Impl, 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.Train.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 :



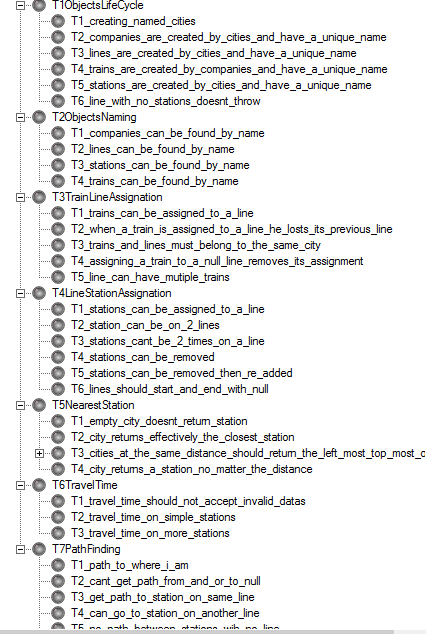
Ces deux tests analysent les objets implémentés dans ITI.Train.dll et en extraient l’API public sous forme XML. Le deuxième se contente d’afficher l’XML dans la console, le premier, plus intéressant, compare l’API public avec celle d’origine qui est mémorisée : si d’aventure, vous altérez l’API public 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.Train (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.Train.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 une ville. Les villes sont indépendantes les unes des autres.
* On peut créer des sociétés dans une ville.
  + Les sociétés ont un nom qui ne change jamais.
  + On doit pouvoir retrouver une société par son nom.
* On peut créer des trains dans une société.
  + Les trains ont un nom qui ne change jamais.
  + On doit pouvoir retrouver un train par son nom.
  + On doit pouvoir assigner un train à une ligne et l‘en changer.
* On doit pouvoir créer des stations.
  + Chaque station à un nom et une position (X, Y) unique.
  + On doit retrouver une station par son nom.
* On doit pouvoir créer des lignes (Lines).
  + Chaque ligne à un nom unique.
  + On doit retrouver une ligne par son nom.
  + On doit pouvoir assigner une station à une ou plusieurs lignes (mais pas deux fois à la même).
* Une ville permet de savoir qu’elle est la station la plus proche d’un point donné (X, Y).
* Une ville permet de trouver son chemin d’un point A à un point B s’il existe (plus court et plus rapide)
* Une ville permet de savoir le temps de trajet entre deux stations d’une même ligne

Il y a 36 tests à passer au vert :



Au boulot !