

**Exercice : Instagram Events**

# Enoncé

L'objectif de cet exercice est d'implémenter un modèle de gestion d’une plateforme similaire à Instagram 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 :

* InstagramEvents.Tests, contient les tests unitaires
* InstagramEvents, 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 InstagramEvents.Tests en tant que projet de démarrage puis d'exécuter la solution.

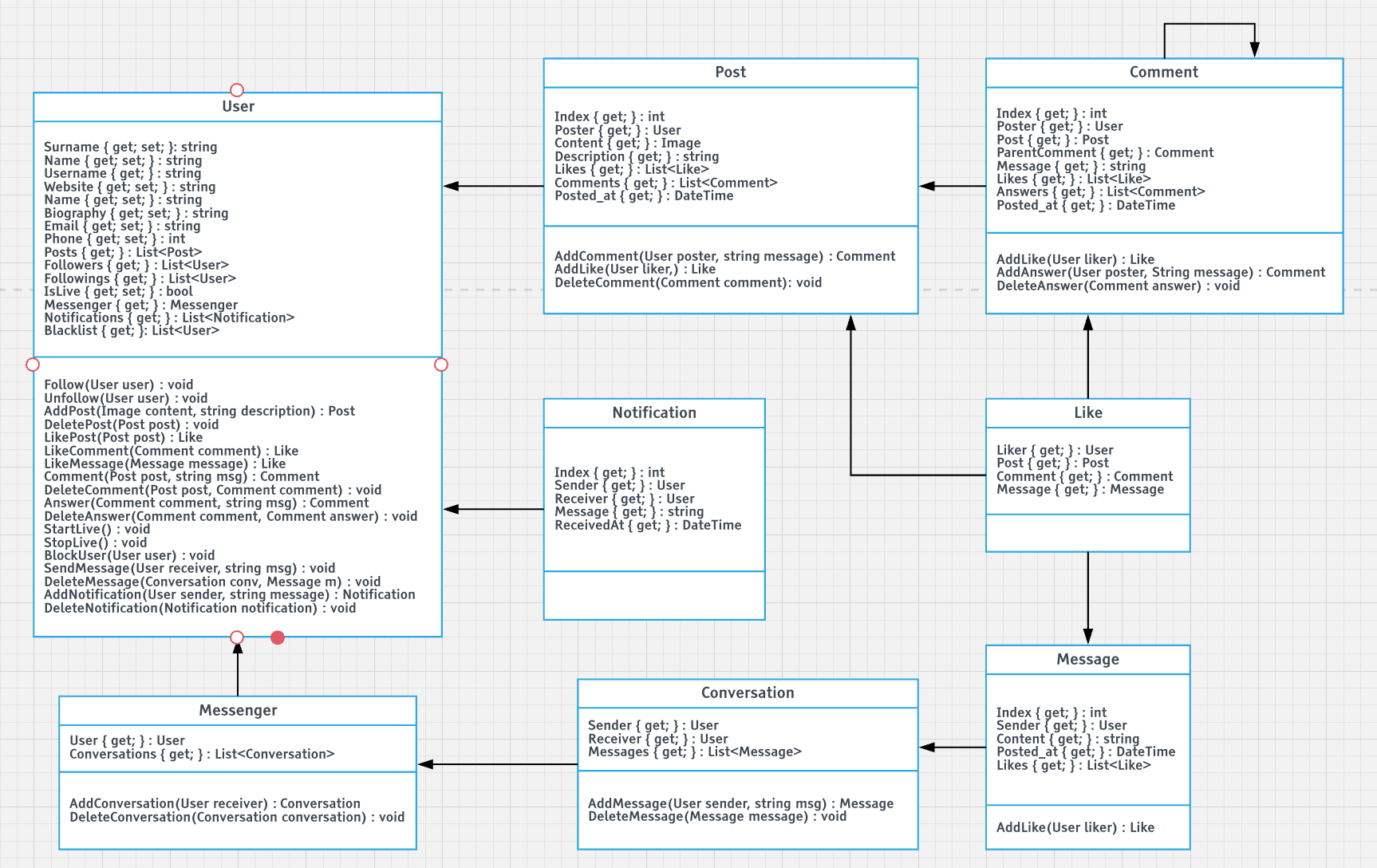
Comme vous pouvez le constater, pour le moment, tous les tests sont rouges.

Votre implémentation ne doit pas modifier l’API publique !

A vous de faire en sorte que les tests passent en vert. Pour cela, vous avez le droit de faire ce que bon vous semble dans le projet InstagramEvents (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 InstagramEvent.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 utilisateur.
  + Les utilisateurs ont un username qui ne change jamais.
  + Les utilisateurs sont indépendants les uns des autres.
* On doit pouvoir follow/unfollow/bloquer d’autres utilisateurs.
  + Se follow/bloquer soi-même doit être impossible.
* On doit pouvoir créer/supprimer des posts à partir d’un utilisateur.
  + Supprimer des posts qui ne nous appartiennent pas doit être impossible.
* On doit pouvoir créer/supprimer des commentaires à un post à partir d’un utilisateur.
  + Supprimer des commentaires qui ne nous appartiennent pas doit être impossible.
* On doit pouvoir créer/supprimer des réponses à un commentaire à partir d’un utilisateur.
  + Supprimer des réponses qui ne nous appartiennent pas doit être impossible.
* On doit pouvoir créer des likes à un post, à un commentaire ou à un message à partir d’un utilisateur.
* Le blocage d’un utilisateur doit avoir des conséquences fonctionnelles.
* Chaque utilisateur doit avoir un Messenger, qui correspond à un fil de conversations.
  + On doit pouvoir envoyer des messages à un autre utilisateur.
  + On doit pouvoir créer/supprimer des conversations à partir du Messenger.
  + On doit pouvoir créer/supprimer des messages à partir d’une conversation.
* Les actions follow, like, comment, startLive, sendMessage doivent déclencher des évènements.
  + L’event handling doit déboucher sur la génération de notifications rattachées à un utilisateur.
  + On ne peut pas se notifier soi-même.



Il y a 30 tests à passer au vert :

