Examen 2 - Formatif

# 420-5W5

## Consignes

* Toute documentation est permise pour l’examen formatif et sommatif.

# Question 1 – Tests unitaires et Mock (12 points)

À partir des fichiers de départs, tester le TripsController.

Pour tester le contrôleur, vous devrez utiliser un ou plusieurs mock.

Vous devrez donc utiliser le package Nuget Moq.

Révision : pour créer des tests unitaires pour UneMethode d’UnService, simplement faire :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

(Ça génère même le projet !)

Le contrôleur ne contient que 2 actions, GetTrips et ShareTrip.

## GetTrips

Tout utilisateur peut avoir accès à GetTrips. GetTrips retourne 2 listes de voyage, les voyages appartenant à l’utilisateur connecté, ainsi que tous les voyages publics.

Dans le cas où l’utilisateur n’est pas connecté, la liste des voyages de l’utilisateur restera simplement vide.

## ShareTrip

Un utilisateur doit être connecté pour avoir accès à cette action.

L’action sert à partager un utilisateur à un voyage. On pourrait aussi dire que c’est d’ajouter un utilisateur à la liste d’utilisateurs du voyage.

Dans le cas où l’utilisateur connecté essaye d’ajouter un participant à un voyage dont il ne fait pas partie, l’action retourne un BadRequest.

Dans le cas où le id du voyage n’est pas trouvé, l’action retourne NotFound.

Dans le cas où l’adresse courriel de l’utilisateur que l’on souhaite ajouter au voyage ne correspond à aucun utilisateur, l’action retourne NotFound.

# Question 2 – SignalR (14 points)

À partir des fichiers de départs, compléter une application qui permet de :

* Choisir une couleur favorite
* Connaître le nombre d’utilisateurs qui ont sélectionnés chaque couleur
* Envoyer un message reçu par les utilisateurs qui ont choisi la même couleur

**Il n’y a pas de changements à faire sauf dans les deux fichiers qui sont mentionnés plus bas !**

La solution utilise 3 évènements pour communiquer entre le serveur et le client :

* InitFavorites qui fournit l’array **nbFavoritesPerColor**
* UpdateFavorites qui a 2 paramètres : la couleur et la quantité d’utilisateur qui la préfère
* ReceiveMsg qui reçoit le message à afficher

**Note** : Le système pourrait fonctionner avec un seul event pour mettre **nbFavoritesPerColor** à jour, mais on demande d’utiliser InitiFavorites à la connection et UpdateFavorites pour mettre à jour les couleurs de façon individuelle par la suite.

## Hub

Il faut compléter le code de **FavoriteColorHub.cs .** Il y a des **TODOs** dans le fichier qui indique ce qui doit être fait.

## Angular

Il faut compléter le code de **app.component.ts .** Il y a des **TODOs** dans le fichier qui indique ce qui doit être fait.

Conseil :

* Pour tester l’application vous pouvez simplement ouvrir plusieurs fenêtres. Même si c’est avec le même navigateur :

Une image contenant texte, capture d’écran, logiciel, conception

Description générée automatiquement

Dans cet exemple. Si l’un des 2 premiers navigateurs envoi un message, il va être reçu par les 2 premiers navigateurs (ils ont choisi rouge tout les deux). Par contre, si le 3e navigateur envoi un message, il va être le seul à le voir.

**Grille de correction**

|  |  |
| --- | --- |
| **Section** | **Points** |
| Question 1 | |
| Test de GetTrips avec utilisateur connecté | 2 |
| Test de GetTrips avec utilisateur pas connecté | 2 |
| Test de ShareTrip qui fonctionne correctement | 2 |
| Test de ShareTrip avec un voyage qui n’appartient pas à l’utilisateur connecté | 2 |
| Test de ShareTrip avec un mauvais id de voyage | 2 |
| Test de ShareTrip avec une mauvaise adresse courriel | 2 |
| Question 2 | |
| Connection du Hub | 2 |
| Déconnection du Hub | 2 |
| Choix de couleur sur le Hub | 3 |
| Envoi de message | 1 |
| Gestion des events Angular | 3 |
| Déconnexion Angular | 1 |
| Choix de la couleur Angular | 1 |
| Envoi de message Angular | 1 |
| **Total** | **26** |