420-4W6 - TP3

Projet

SuperGalerieInfinie

Date de remise : Semaine 11

Pondération : 15%

# Consignes

Le projet sera fait de façon individuelle.

Le projet client sera fait avec le Framework Angular.

Le projet serveur sera fait avec le Framework .Net Core Web API.

L’usage de Bootstrap est interdit. (Sinon la partie 2 du TP ne marchera pas bien)

# Mise en contexte

Le but est de faire une application avec des galeries d’images. On peut s’inscrire et se connecter à l’application, créer des galeries (qui nous appartiennent), supprimer des galeries, voir les galeries des autres, etc.

### Le TP3 est la « Partie 1 »

Ce TP représente la partie 1. Le TP4 sera la partie 2 d’un même projet. Pour la partie 1, nous ne ferons pas encore la gestion d’images. (Donc les utilisateurs peuvent créer et supprimer des « galeries », mais pas mettre de photos à l’intérieur pour le moment !)

# Gestion des utilisateurs

Vous devrez permettre l’enregistrement d’un nouvel utilisateur, la connexion d’un utilisateur existant et la déconnexion d’un utilisateur.

# Seed de base de données

Vous devrez garnir votre base de données de données de test lors de son initialisation. Il faut au moins deux utilisateurs avec chacune une galerie. Une galerie doit être publique et l’autre doit être privée.

# Gestion des galeries

Un utilisateur pourra créer de nouvelles galeries.

Lors de la création, une image par défaut sera utilisée comme photo de couverture pour la galerie en attendant le TP4. Cette image peut être hardcodée côté client (Angular) pour l’instant.

Il sera possible de mettre rendre une galerie privée ou publique (et d’alterner pour une galerie existante), les galeries publiques peuvent être vues par tous les utilisateurs, mais ils ne peuvent bien entendu pas supprimer les galeries des autres. Un utilisateur peut supprimer ses galeries.

Il sera possible de « partager » une galerie avec un autre utilisateur. L’utilisateur sera ajouté à la galerie et deviendra propriétaire au même titre que l’utilisateur initial. (Il pourra donc désormais supprimer la galerie s’il le souhaite, la rendre privée / publique et voir la galerie même si elle est privée, comme l’utilisateur initial) On peut déduire que la relation entre Galerie et Utilisateur est Many-To-Many.

# Correction

|  |  |  |
| --- | --- | --- |
| 10 pts | Galeries | |
|  | 2 pts | Ajouter |
| 2 pts | Supprimer |
| 1 pt | Lier les galeries à un utilisateur dans la BD |
| 1 pt | *Partager une galerie (galerie du créateur, cette galerie à plusieurs users avec les mêmes droit)* |
| 1 pt | Changer la visibilité d’une galerie (privée / publique) (put) |
| 2 pts | Afficher les galeries (publiques et privées) d'un utilisateur |
| 1 pt | Afficher les galeries publiques des autres utilisateurs (linq dans le get) |
| 4 pts | Gestion des utilisateurs | |
|  | 1 pts | Inscription |
| 2 pts | Connexion |
| 1 pt | Déconnexion |
| 6 pts | Divers | |
|  | 1 pt | Sécurité (On ne peut pas supprimer les galeries des autres... on ne peut pas partager une galerie sans en être le propriétaire... on ne peut pas accéder aux infos des utilisateurs en obtenant la liste des galeries publiques...) |
| 1 pt | Services côté serveur (Couche entre les contrôleurs et le DbContext) (générique) |
| 1 pt | Utiliser un interceptor (Angular) |
| 2 pts | Données d’initialisation dans la BD (Seed) |
| 1 pt | Architecture Angular bien organisée (Usage de services, de modèles, de composants) |
| 20 pts | Total | |