420-4W6 - TP4 - Post Hub (20%)

Date de remise : 13 ou 14 mai, selon le groupe

# Consignes

* Le projet sera fait en équipe de deux imposée. (Voir annexe)
* Le projet client sera fait avec le Framework Angular. (Un projet de départ est fourni !)
* Le projet serveur sera fait avec le Framework .Net Core.

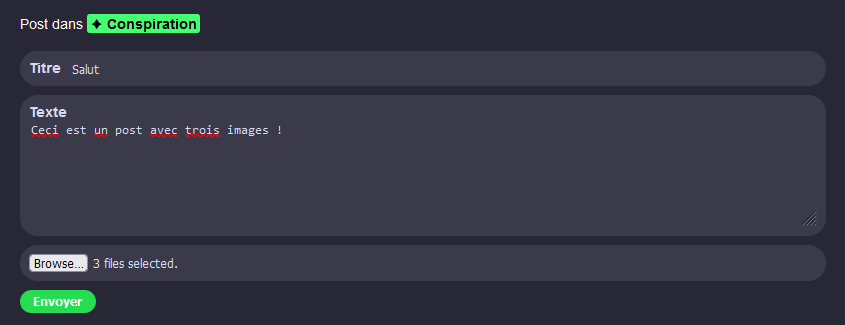
# Fonctionnalités Yang

Ce document donne des indices, mais bien entendu, il n’est ni exhaustif, ni parfait.

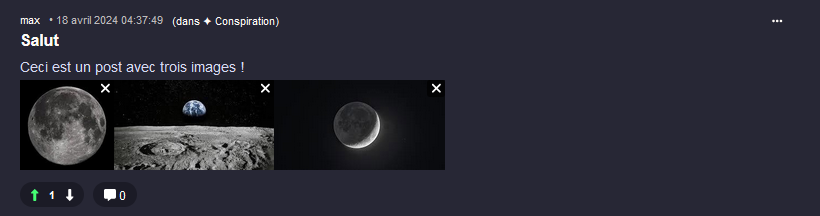
Étape C

« Lorsqu’on crée un post, on doit être capable d’y joindre une ou plusieurs images. »

* Cela se passe surtout dans le composant editPost. (Qui permet de créer un post, désolé il est mal nommé…) Pour créer un post, choisissez un hub spécifique d’abord.

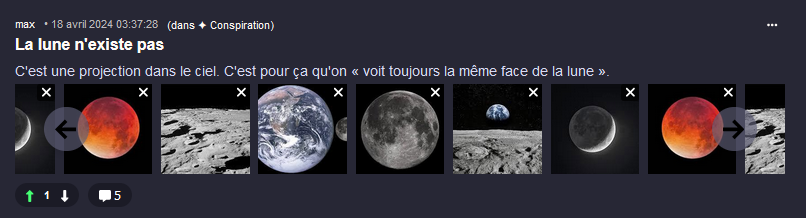


* Dans les notes de cours, on a seulement vu comment joindre une image à un input pour envoyer une seule image au serveur. Dans ce cas, ce sera un peu différent. Ce probème se règle entre autre avec une boucle while.
* Le serveur reçoit un PostDTO dans le corps de la requête actuellement pour créer un nouveau post… ce ne sera plus possible car on va maintenant envoyer un formData.
* N’oubliez pas d’afficher les images dans le composant **fullPost**.

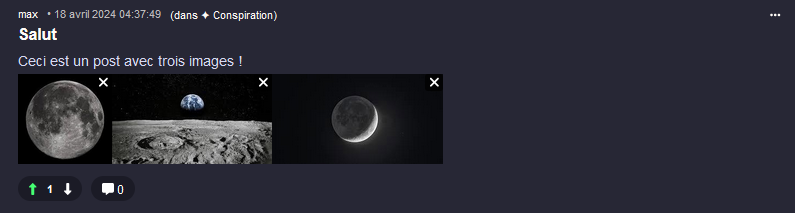


Étape D

« Si un post (commentaire principal d’un post) contient plus de 4 images, les images sont affichées avec un carrousel glidejs plutôt qu’en simple rangée. Les commentaires d’un post n’ont jamais de carrousel. »



* Pour le carrousel, vous êtes invités à créer un nouveau composant qui sera affiché dans le composant **fullPost**.
* Le seul endroit où il est obligatoire d’afficher un carrousel est dans **fullPost**, pour le commentaire principal d’un post avec plus de 4 images. Avec 4 images ou moins, on affiche simplement les images normalement :

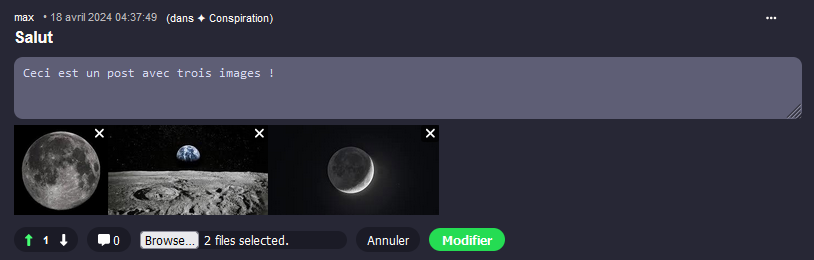


Étape E

« Lorsqu’on modifie un commentaire ou un post, on doit pouvoir ajouter une ou plusieurs images supplémentaires en même temps de modifier le texte. (Les anciennes images du commentaire, s’il y en avait, sont préservées) »

* Vous allez vous inspirer largement des actions PostPost / PostComment pour cette étape. N’oubliez pas qu’il y a deux endroits où on peut modifier un commentaire : dans **fullPost** et dans **comment**.
* Encore une fois, le formData va remplacer tout DTO qui était précédemment utilisé.

On voit, ci-dessous, que je modifie mon message et que j’ai ajouté 2 images dans l’input.



Étape F

« Lorsqu’on supprime un commentaire ou un post, toutes ses images doivent être supprimées. »

* Vous remarquerez que l’action DeleteComment fait peur. C’est parce qu’elle hard-supprime également tout sous-commentaires ou commentaires parents qui étaient déjà soft-supprimés. Ne vous inquiétez pas trop avec cela : votre objectif est seulement de supprimer les images du commentaire cliqué s’il en avait.
* Les images du commentaire supprimées doivent être cachées immédiatement dans la page Web.

Étape G

« Les utilisateurs doivent pouvoir se connecter en utilisant leur nom d’utilisateur OU leur adresse courriel. (Plutôt que seulement leur nom d’utilisateur) Les utilisateurs doivent pouvoir changer leur mot de passe. »

* Il faudra beaucoup chercher ailleurs que dans les notes de cours pour cette étape, mais ce n’est pas très compliqué. Tout se passe dans le composant **profile** et dans **UsersController**.

Étape H

« Un rôle administrateur existe. Les administrateurs peuvent ajouter le rôle modérateur à des utilisateurs. Ils peuvent supprimer les commentaires de leur choix via cette liste. Un utilisateur avec le rôle administrateur est ajouté dans le seed. »



* Un bouton dans le composant **profile** doit permettre d’afficher ce mini formulaire qui permettra de nommer des modérateurs grâce à leur nom d’utilisateur.