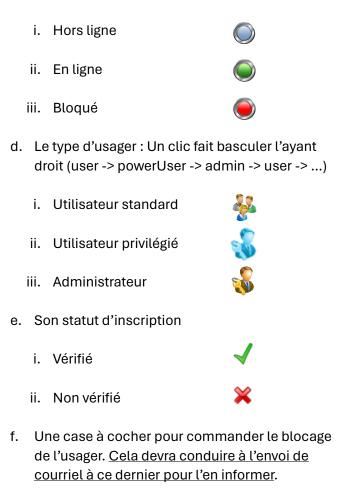
Projet de départ : https://github.com/Nicolas-Chourot/MoviesDBManager-Auth

1. Ajoutez à la classe OnlineUsers l'accesseur suivant :

Note : Le dictionnaire HttpRuntime.Cache est disponible tant que l'application Web n'est pas expirée. Cela permet de conserver des données accessibles à toutes les sessions.

- 2. Ajoutez le code nécessaire pour maintenir la liste des usagers « connectés ». Ajoutez l'usager à la liste i.e. OnLineUsers.ConnectedUsersId.Add(userId), lorsqu'il se connecte et le retirer lorsqu'il se déconnecte i.e. . OnLineUsers.ConnectedUsersId.Remove(userId). Il vous sera aussi utile d'ajouter une méthode OnlineUsers.IsOnline(userId).
- 3. Empêchez la double connexion d'un usager.
- 4. Ajoutez le code nécessaire pour la gestion d'expiration de session (voir le projet https://github.com/Nicolas-Chourot/PhotosManager-ASP.net).
- 5. Ajoutez l'icone de commande (fa-users) dans le menu principal donnant accès aux usagers administrateurs à la vue Accounts/UsersList. Cet icone devra apparaître que si l'usager connecté est administrateur.

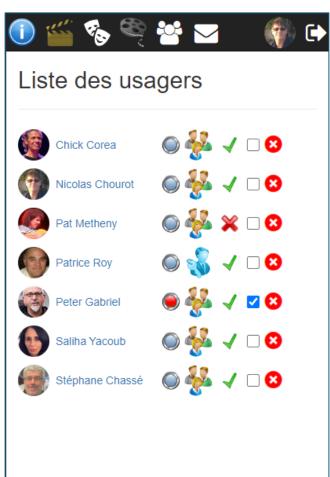
- 6. Produire la vue Accounts/UsersList ci-dessous. Cette vue donnera accès à la liste des usagers (sauf l'usager connecté) en ordre alphabétique <u>rafraichie périodiquement</u> (voir comment cela est implémenté dans les contrôleurs Actors, Movies et Distributors). Pour chacun des usagers il faudra afficher:
 - a. Son avatar
 - b. Son nom au complet (Nom Prénom)
 - c. Un indicateur de connexion (Voir dans content/UI-Images pour les fichiers d'images)



g. L'icone 🔀 pour commander le retrait de

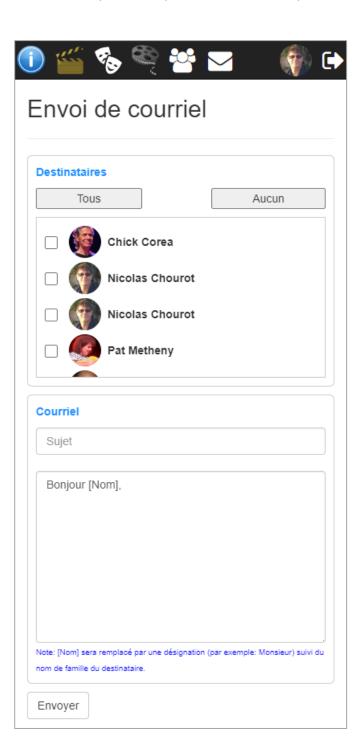
le CSS suivant : fa-time-circle, color:red

l'usager avec confirmation. <u>Il faudra envoyer un courriel à ce dernier pour l'en informer</u>. Utilisez



Note: Tous les icones devront présenter une infobulle indiquant sa fonction lorsque survolés. Le rafraichissement périodique aura besoin de savoir s'il y a eu un changement au niveau des usagers (changement de nom, d'avatar, statut d'inscription, connexion/déconnexion, etc.).

- 7. Ajoutez l'icone (fa-enveloppe) pour atteindre la vue Accounts/GroupEmails. Uniquement les usagers administrateurs pourront voir cet icone.
- 8. Produire la vue Accounts/GroupEmail suivante ci-dessous. Elle permet de sélectionner les usagers ciblés. De rédiger un courriel avec sujet et contenu. Si ce dernier contient le bloc de caractères [Nom], il sera substitué selon le genre de l'usager ciblé par :
 - « < Monsieur | Madame | > < Nom de famille | Nom de famille | Prénom Nom> »



Note importante:

Il faudra envoyer un courriel à chaque utilisateur ciblé plutôt qu'un envoi de groupe. En bref pas de CC ou BCC.

Code javascript utile:

```
<script defer>
   function validateSelection() {
        let oneSelected = false;
$("input[type='checkbox'][name='SelectedUsers']").each(function () {
            if ($(this).prop('checked')) oneSelected = true;
        if (!oneSelected)
            $("#Error_SelectedUsers").css("display", "inline");
        else
            $("#Error_SelectedUsers").css("display", "none");
        return oneSelected;
   $("input[type='checkbox'][name='SelectedUsers']").change(function () { validateSelection(); });
   $("form").submit(function () {
       return validateSelection();
   $("#SelectAll").click(function () {
        $("input[type='checkbox'][name='SelectedUsers']").each(function () { $(this).prop('checked', true) });
        validateSelection();
       return false; // prevent form submit
   $("#DeSelectAll").click(function () {
        $("input[type='checkbox'][name='SelectedUsers']").each(function () { $(this).prop('checked', false) });
        validateSelection();
        return false; // prevent form submit
</script>
```

Critères d'évaluation

- Respect des spécifications
- Fonctionnement sans faille
- Doit être mis en production sur Azure
- Mise en page adéquate et réactive des vues
- Sécurité
- · Remise du projet dans Colnet

Référence sur icone :

https://www.antidote.info/fr/blogue/enquetes/un-icone-un-icone-une-icone-une-icone