


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

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

```
public static List<int> ConnectedUsersId
{
    get
    {
        if (HttpContext.Cache["OnLineUsers"] == null)
            HttpContext.Cache["OnLineUsers"] = new List<int>();
        return (List<int>)HttpContext.Cache["OnLineUsers"];
    }
}
```

Note : Le dictionnaire `HttpContext.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'utilisateur à la liste `OnLineUsers` lorsqu'il se connecte et le retirer lorsqu'il se déconnecte. Il vous sera utile d'ajouter une méthode `OnlineUsers.IsOnline(userId)`.
3. Empêchez la double connexion d'un usager.
4. Ajoutez du code JavaScript dans `Views/Shared/_Layout.cshtml` qui invoque `Accounts/ExpiredSession` après x minutes d'inactivité avec un message informant l'utilisateur que sa session est expirée. Note : la vue `Accounts/login` ne doit jamais expirer. La vue `Accounts/ExpiredSession` doit rediriger l'utilisateur à la vue `Accounts/login` après y minutes. Vous pouvez reprendre les techniques de gestion de session présentées dans le cours 420-KB9.
5. Ajoutez l'icône de commande  (`fa-users`) dans le menu principal donnant accès usagers administrateurs à la vue `Accounts/UsersList`. Cet icône devra apparaître que si l'utilisateur connecté est administrateur.

6. Produire la vue [Accounts/UsersList](#) suivante. Cette vue donnera accès à la liste des usagers (sauf l'utilisateur connecté) en ordre alphabétique **rafraîchie périodiquement**. 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)


i. Hors ligne 

ii. En ligne 

iii. Bloqué 

d. Le type d'utilisateur : Un clic fait basculer l'ayant droit (user -> powerUser -> admin -> user -> ...)

i. Utilisateur 

ii. Administrateur 


iii. PowerUser 

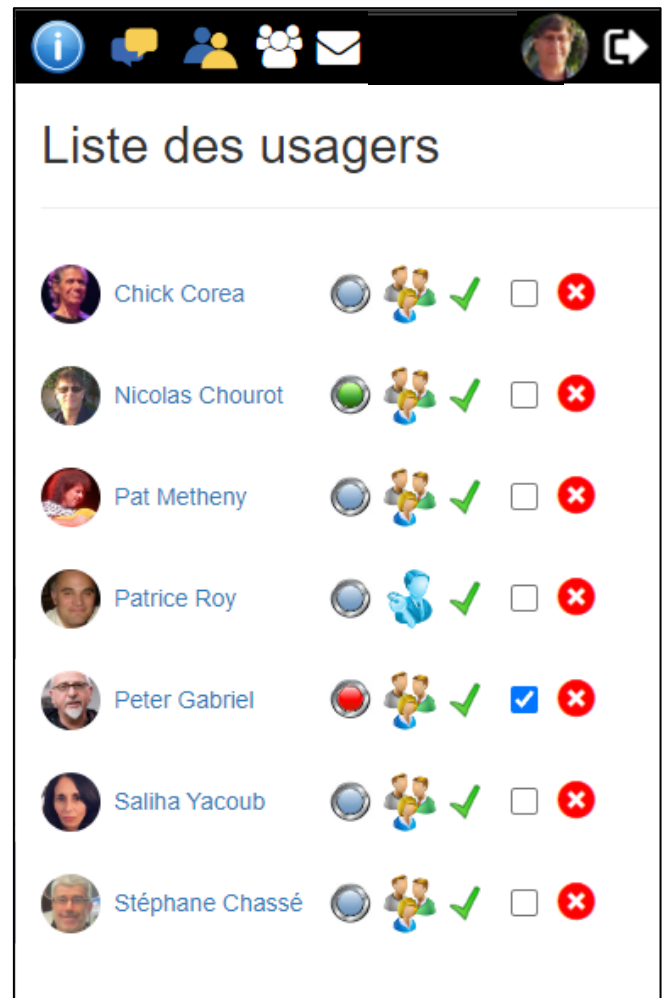
e. Son statut d'inscription

i. Vérifié 


ii. Non vérifié 

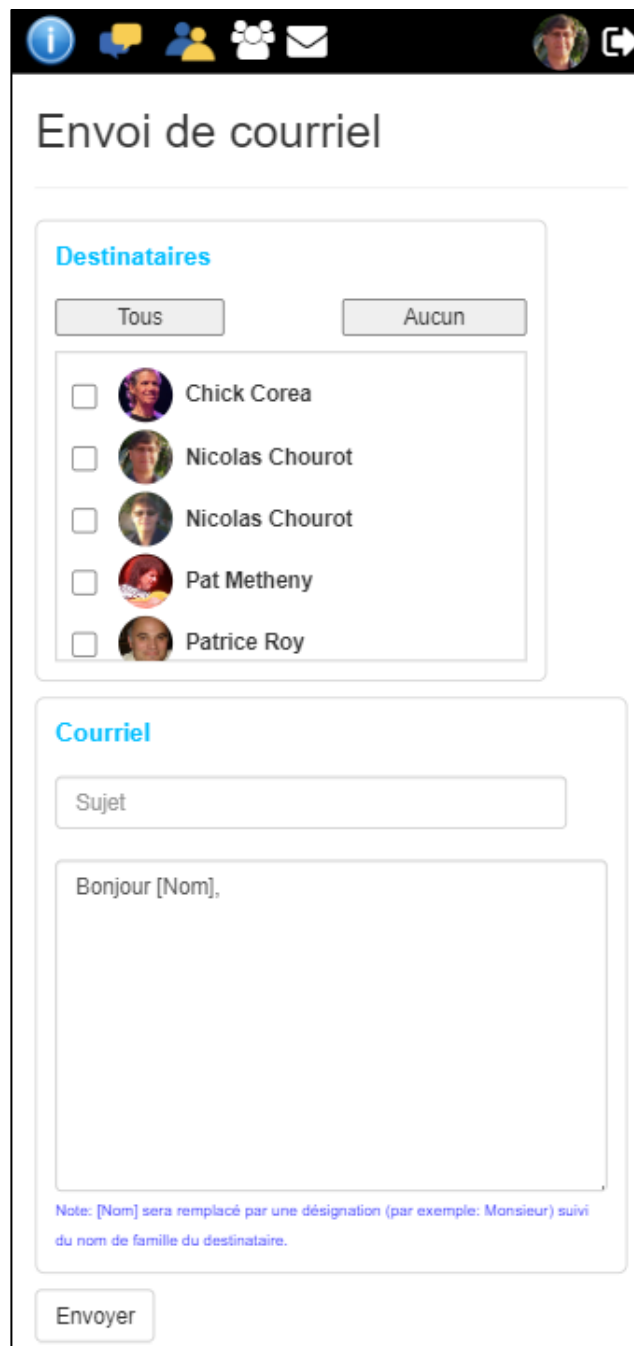
f. Une case à cocher pour commander le blocage de l'utilisateur. Cela devra conduire à l'envoi de courriel à ce dernier pour l'en informer.

g. L'icône  pour commander le retrait de l'utilisateur avec confirmation. Il faudra envoyer un courriel à ce dernier pour l'en informer. Utilisez le CSS suivant : `fa-time-circle, color:red`



Note : Tous les icônes devront présenter un **tooltip** indiquant sa fonction lorsque survolés.






7. Ajoutez l'icône  (fa-enveloppe) pour atteindre la vue [Accounts/GroupEmails](#). Uniquement les usagers administrateurs pourront voir cet icône.
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'utilisateur ciblé par :
« <[Monsieur | Madame](#) | - > <[Nom de famille | Nom de famille | Prénom Nom](#)> »



Envoi de courriel

Destinataires

Tous Aucun

- ☐  Chick Corea
- ☐  Nicolas Chourot
- ☐  Nicolas Chourot
- ☐  Pat Metheny
- ☐  Patrice Roy

Courriel

Sujet

Bonjour [Nom],

Note: [Nom] sera remplacé par une désignation (par exemple: Monsieur) suivi du nom de famille du destinataire.

Envoyer

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