

TD-1 Synchronisation dans l'Application de Gestion de Tâches

1. Explication de la Mise en Œuvre de la Synchronisation des Données

La synchronisation des données est cruciale pour permettre à plusieurs utilisateurs de collaborer en temps réel sur un projet. Dans votre application de gestion de tâches, voici les éléments essentiels concernant la synchronisation des données :

Technologies Utilisées

- **API REST** : Les API RESTful seront utilisées pour les opérations classiques comme la création, la modification et la suppression de projets, de tâches et de groupes de tâches. Ces opérations seront effectuées de manière asynchrone, garantissant une communication fluide entre le client (interface utilisateur) et le serveur (backend).
- **WebSockets** : Pour les interactions en temps réel, la technologie WebSocket sera utilisée. Cela permettra de pousser des mises à jour en temps réel vers les clients connectés sans avoir besoin d'envoyer des requêtes répétitives au serveur (polling).

Fonctionnement Global

1. **Accès initial aux données** : Lorsqu'un utilisateur se connecte à l'application ou accède à un projet spécifique, une requête **API REST** est effectuée pour récupérer toutes les données nécessaires (projets, groupes de tâches, tâches, etc.) à partir de la base de données.
2. **Synchronisation continue avec WebSockets** : Une fois connecté à un projet, un canal WebSocket est ouvert entre le client et le serveur pour écouter toutes les mises à jour effectuées par d'autres utilisateurs en temps réel. Par exemple, si un utilisateur modifie une tâche ou change son état, la modification est immédiatement envoyée à tous les autres utilisateurs connectés au projet via ce canal.
3. **Notification des modifications** : Dès qu'une tâche est modifiée, supprimée ou ajoutée, ou qu'un groupe de tâches est réorganisé, le serveur envoie un événement via le WebSocket. Le client côté utilisateur reçoit cet événement et met à jour l'interface utilisateur en conséquence, garantissant que toutes les données restent synchronisées.

Gestion des conflits de données

Pour éviter les conflits potentiels lorsque plusieurs utilisateurs modifient simultanément les mêmes données :

- **Horodatage des modifications** : Chaque modification est associée à un horodatage. En cas de conflit, la modification la plus récente est prise en compte.
- **Locking optimiste** : En utilisant un verrouillage optimiste (optimistic locking), si un utilisateur essaie de modifier une ressource qui a été modifiée par un autre utilisateur entre-temps, un message d'erreur est renvoyé, et l'utilisateur est invité à recharger les données.

Exemple Concret

- Lorsqu'un utilisateur modifie l'état d'une tâche de "À faire" à "En cours", cette modification est envoyée via une requête REST (si c'est une action unique) ou via WebSocket (si d'autres utilisateurs sont connectés).
- Tous les autres utilisateurs connectés à ce projet verront instantanément la tâche passer de "À faire" à "En cours" sur leur écran sans avoir besoin de recharger la page.

2. Prise en Charge de la Synchronisation Entre Utilisateurs

Pour permettre une **collaboration fluide** entre plusieurs utilisateurs travaillant sur le même projet, plusieurs mécanismes de synchronisation entre utilisateurs seront mis en place.

Accès Collaboratif en Temps Réel

Grâce à l'utilisation de **WebSockets**, chaque utilisateur connecté à un projet reçoit des mises à jour en temps réel concernant :

- **Modifications de projet** : Tout changement dans les informations du projet (nom, description, état) sera immédiatement synchronisé entre les utilisateurs.
- **Ajout/Suppression de Groupes de Tâches et de Tâches** : Lorsqu'un utilisateur ajoute un nouveau groupe de tâches ou une tâche, cette modification est envoyée instantanément à tous les autres utilisateurs connectés au projet.
- **Changement d'état des tâches** : Si un utilisateur modifie l'état d'une tâche (par exemple, de "En attente" à "Terminé"), ce changement est immédiatement visible par tous les autres utilisateurs.

Cela permet à chaque membre de l'équipe d'avoir une vue d'ensemble du projet sans retard ni désynchronisation.

Notifications de Modification

- Les utilisateurs reçoivent des **notifications visuelles** lorsque des changements sont effectués par d'autres membres de l'équipe, comme la modification d'une tâche ou l'ajout de nouveaux utilisateurs au projet.
- Ces notifications peuvent apparaître sous forme de **pop-ups** ou de **modifications animées** dans l'interface utilisateur pour rendre la collaboration intuitive.

Gestion des Permissions et Synchronisation

- Les utilisateurs ayant des **droits d'édition** peuvent apporter des modifications (ajout, suppression, mise à jour des tâches et projets) qui seront synchronisées avec tous les autres utilisateurs.
- Les utilisateurs ayant des droits en tant que **lecteurs** peuvent visualiser les modifications en temps réel sans avoir la possibilité d'apporter eux-mêmes des modifications. Cependant, toutes les actions qu'ils réalisent, comme le changement de vue ou la navigation dans l'interface, seront également synchronisées pour assurer une expérience collaborative cohérente.

Exemple de Collaboration

- **Utilisateur A** et **Utilisateur B** travaillent sur le même projet. Utilisateur A modifie la description d'un projet, changeant l'état d'un groupe de tâches de "En attente" à "En cours".
- **En temps réel**, Utilisateur B voit les modifications effectuées par Utilisateur A sans avoir à recharger l'application.
- Si Utilisateur B modifie ensuite une tâche, cette modification est également immédiatement visible pour Utilisateur A et tout autre utilisateur connecté au projet.

Synchronisation entre les Appareils

- Les données sont également synchronisées lorsque les utilisateurs utilisent plusieurs appareils. Par exemple, un utilisateur peut accéder au même projet depuis un ordinateur et un smartphone, et toute modification apportée sur un appareil sera immédiatement synchronisée sur l'autre.

Conclusion

La synchronisation dans l'application de gestion de tâches est mise en œuvre grâce à une combinaison d'API REST pour les opérations de base et de WebSockets pour les mises à jour en temps réel. Cela permet une collaboration fluide entre plusieurs utilisateurs, garantissant que toutes les données sont constamment à jour pour chaque membre de l'équipe. Des mécanismes de gestion des conflits assurent une intégrité des données, et les permissions des utilisateurs sont respectées pour une collaboration sécurisée et contrôlée.