

Note d'analyse et de Correction

- **Problème initial :** La fonction d'ajout d'un avis ne fonctionne pas.

Analyse :

1. Le problème initial était que la méthode `updateEvent` dans le contrôleur ne faisait rien, ce qui empêchait l'ajout d'avis de fonctionner.
2. Pour résoudre ce problème, j'ai ajouté une implémentation à la méthode `updateEvent` dans le contrôleur pour appeler le service correspondant.

Correction initiale :

1. J'ai modifié la méthode `updateEvent` dans le contrôleur pour appeler `eventService.updateStars(id, event)` afin de mettre en œuvre la mise à jour des étoiles (nbStars) et du commentaire (comment) de l'événement.
2. Cette modification a permis d'activer la fonction d'ajout d'un avis.

Problème ultérieur : Même après l'ajout de l'implémentation dans la méthode `updateEvent`, les avis ajoutés n'étaient pas enregistrés de manière permanente et étaient perdus après l'actualisation de la page.

Analyse ultérieure :

1. Le problème ultérieur était lié à la persistance des avis dans la base de données.
2. Pour résoudre ce problème, j'ai ajouté l'annotation `@Transactional` à la méthode `updateStars` dans le service pour garantir la persistance des modifications apportées à la base de données.

Correction ultérieure :

1. J'ai ajouté `@Transactional` à la méthode `updateStars` du service pour garantir que les modifications apportées à la base de données (ajout d'avis) sont commitées et enregistrées de manière permanente lorsque la méthode se termine avec succès.
2. Cette modification a résolu le problème de non-persistance des avis après l'ajout.

Résultat final : Après avoir ajouté `@Transactional` et mis en œuvre l'appel à la méthode `eventService.updateStars` dans le contrôleur, les avis sont désormais correctement enregistrés dans la base de données et ne sont plus perdus après l'actualisation de la page.

- **Problème :** Lors de l'utilisation de la méthode `delete` dans le service `EventService`, les éléments supprimés reviennent lorsque la page est actualisée.

Problème : Lors de l'utilisation de la méthode `delete` dans le service `EventService`, les éléments supprimés reviennent lorsque la page est actualisée.

Analyse :

1. Le problème semble être lié à la gestion des transactions lors de la suppression d'événements dans la base de données.
2. La méthode `delete` supprime l'événement, mais il semble que les modifications ne soient pas persistées correctement dans la base de données.
3. Il est important de s'assurer que la suppression est effectivement commitée dans la base de données pour éviter que les données ne reviennent.

Correction :

1. J'ai ajouté l'annotation `@Transactional` à la méthode `delete` dans la classe `EventService`. Cela indique à Spring de gérer la transaction automatiquement autour de cette méthode.
2. L'annotation `@Transactional` garantit que la suppression est effectuée dans le contexte d'une transaction.
3. J'ai également ajouté des instructions de journalisation (`logger.info`) pour documenter les actions effectuées et les états des transactions.
4. Si la suppression réussit, la transaction est validée (`commit`) et les modifications sont persistées dans la base de données.
5. En cas d'erreur, la transaction est annulée (`rollback`), garantissant que la base de données reste cohérente même en cas d'exception.

Correction :

1. J'ai ajouté l'annotation `@Transactional` à la méthode `delete` dans la classe `EventService`. Cela indique à Spring de gérer la transaction automatiquement autour de cette méthode.
2. L'annotation `@Transactional` garantit que la suppression est effectuée dans le contexte d'une transaction.
3. J'ai également ajouté des instructions de journalisation (`logger.info`) pour documenter les actions effectuées et les états des transactions.
4. Si la suppression réussit, la transaction est validée (commit) et les modifications sont persistées dans la base de données.
5. En cas d'erreur, la transaction est annulée (rollback), garantissant que la base de données reste cohérente même en cas d'exception.

Journalisation :

- `logger.info("Début de la suppression de l'événement avec l'ID : " + id);` - Documente le début de la suppression de l'événement.
- `logger.info("Suppression de l'événement avec succès.");` - Documente la réussite de la suppression.
- `logger.error("Erreur lors de la suppression de l'événement avec l'ID : " + id, e);` - Documente une erreur en cas de problème lors de la suppression.