

Tests End-to-End (E2E) – MySurvey

1. Introduction

Les tests end-to-end (E2E) valident le comportement global de l'application en simulant des appels HTTP réels vers les endpoints exposés par l'API REST. Ils permettent de vérifier que l'ensemble des couches (contrôleurs, services, mapping DTO/entités et persistance) interagit correctement dans un contexte complet. Ces tests sont exécutés avec Spring Boot dans un environnement de test avec une base de données et utilisent MockMvc pour simuler des requêtes HTTP.

2. Environnement de Test

- **Framework** : Spring Boot Test avec JUnit 5.
- **Web Environment** : Lancement de l'application sur un port aléatoire (`RANDOM_PORT`).
- **Configuration** : Les tests sont annotés avec `@Transactional` afin que chaque scénario soit isolé et annulé à la fin de son exécution.
- **Outils** :
 - **MockMvc** pour simuler des requêtes HTTP.
 - **ObjectMapper** pour la sérialisation/désérialisation JSON.

3. Scénarios de Tests

3.1. Test des Commentaires

- **Objectif** : Vérifier la gestion complète des commentaires sur un sondage.
- **Scénarios** :
 - **Création d'un Commentaire** :
 - Un participant et un sondage sont d'abord créés.
 - Un nouveau commentaire est envoyé via un POST sur l'endpoint `/api/sondage/{sondageId}/commentaires`.
 - Le test vérifie que le commentaire est créé avec le contenu attendu (par exemple, « C'est un excellent produit! »).
 - **Mise à jour d'un Commentaire** :
 - Après création, le commentaire est mis à jour via un PUT sur l'endpoint `/api/commentaire/{commentaireId}`.
 - Le test vérifie que le contenu du commentaire a été modifié correctement (par exemple, « Commentaire mis à jour! »).

- **Suppression d'un Commentaire :**
 - Un commentaire est créé, puis supprimé via l'endpoint DELETE `/api/commentaire/{commentaireId}`.
 - Le test confirme que la suppression se fait sans erreurs.

3.2. Test des Participants (ParticipantE2eTest)

- **Objectif :** Valider l'ensemble des opérations sur l'entité Participant via l'API.
- **Scénarios :**
 - **Création :**
 - Un nouveau participant (par exemple, "Charlie Brown") est créé via l'endpoint `/api/participant/`.
 - Le test vérifie que le participant créé retourne les attributs corrects (nom et prénom).
 - **Consultation :**
 - Le test récupère un participant par son ID et vérifie que les données retournées correspondent aux valeurs attendues.
 - **Mise à jour :**
 - Un participant existant est mis à jour via un PUT sur `/api/participant/{id}`.
 - Le test vérifie que les modifications (par exemple, changement de nom ou prénom) sont correctement persistées.
 - **Suppression :**
 - La suppression d'un participant est réalisée via l'endpoint DELETE et le test vérifie que l'opération se termine correctement.

3.3. Test des Sondages

- **Objectif :** Vérifier l'intégralité du cycle de vie d'un sondage et la gestion de ses relations.
- **Scénarios :**
 - **Création :**
 - Un sondage est créé via l'endpoint `/api/sondage/` en utilisant un participant existant comme créateur.
 - Le test vérifie que le sondage créé possède les bonnes informations (nom, description, etc.).
 - **Consultation :**
 - L'API est interrogée pour récupérer la liste des sondages. Le test s'assure que le résultat est un tableau et que les sondages retournés possèdent les attributs attendus.
 - **Mise à jour :**
 - Un sondage existant est mis à jour via un PUT sur `/api/sondage/{sondageId}`.

- Le test vérifie que les nouvelles valeurs (nom et description modifiés) sont correctement enregistrées.
- **Suppression :**
 - La suppression d'un sondage est effectuée via l'endpoint DELETE et le test confirme que l'opération se termine avec succès.

4. Conclusion

Les tests end-to-end de MySurvey ont permis de valider les scénarios critiques suivants :

- La création, la mise à jour et la suppression des commentaires, des participants et des sondages.
- La cohérence des données échangées via l'API REST grâce à un mapping fiable entre DTOs et entités.
- La robustesse des endpoints exposés, garantissant que l'ensemble des couches de l'application interagissent correctement dans un contexte global.

Ces tests E2E renforcent la confiance dans la qualité et la fiabilité de l'application en simulant des interactions utilisateur réelles dans un environnement proche de la production.