

Exercice : Gestion des notifications utilisateur avec Spring Cloud AWS et SQS

Contexte :

Une entreprise veut mettre en place un système de gestion des notifications pour informer ses utilisateurs des événements importants (inscription, changement de mot de passe, etc.). Ces notifications doivent être envoyées via une file d'attente SQS pour assurer la résilience et la scalabilité.

Objectifs :

1. Créer et gérer une file d'attente SQS.
2. Envoyer des notifications sous forme de messages JSON à la file.
3. Consommer ces messages pour les afficher ou les traiter (par exemple, les envoyer par e-mail ou SMS).

Étapes :

1. Créer un modèle de données pour les notifications

- Définir un format JSON pour représenter une notification utilisateur.

Exemple : `{ "userId": "123", "type": "PASSWORD_RESET", "message": "Your password has been reset successfully." }`

2. Configurer une file d'attente SQS :

- Identifier le nom de la file dans la configuration.
- Utiliser LocalStack pour simuler l'environnement AWS si vous ne souhaitez pas déployer directement sur AWS.

3. Créer un service d'envoi des notifications :

- Implémentez une méthode pour envoyer des notifications à la file SQS.
- Simulez des envois pour plusieurs types d'événements utilisateur (exemple : inscription, modification de profil).

4. Créer un service de réception des notifications :

- Implémentez un listener qui consomme les messages depuis SQS.
- Simulez le traitement de ces messages en les affichant dans les logs ou en les stockant dans une base de données.

5. Tests fonctionnels :

- Simulez un scénario où plusieurs notifications sont envoyées simultanément.
- Vérifiez que tous les messages sont consommés et traités correctement.
- Testez les cas où une notification est invalide (exemple : un champ obligatoire manquant).

6. Options avancées (facultatives) :

- Implémentez un mécanisme de gestion des erreurs (Dead Letter Queue).
- Ajoutez des métriques pour surveiller le nombre de messages envoyés et consommés.
- Proposez un moyen de prioriser certains types de messages (exemple : notifications critiques).

Livrables attendus :

1. Une application fonctionnelle capable d'envoyer et de recevoir des notifications via SQS.
2. Un fichier README.md expliquant :
 - Les étapes pour configurer et exécuter l'application.
 - Les résultats des tests (incluant des captures d'écran ou des logs montrant les notifications traitées).
3. (Optionnel) Des extensions ou améliorations proposées dans l'option avancée.

Critères d'évaluation :

- Respect des bonnes pratiques de Spring Boot et AWS.
- Capacité à simuler un environnement AWS localement (LocalStack).
- Robustesse de la gestion des erreurs.
- Documentation claire et complète.