Assicurarsi che la transazione sia finita

Come eseguire un'operazione avendo la certezza che una transazione sia terminata?

In questo articolo cercheremo di capire che cosa si intende per transazione e come intercettare il completamento di una transazione all'interno di un metodo dei services

Cenni sul concetto di transazione

Quando parliamo di "transazione" in ambito informatico intendiamo una serie di operazioni, tipicamente che coinvolgono scritture su database, eseguite in sequenza. Eseguire queste operazioni in una transazione significa che, se anche una sola di queste fallisce, allora vengono annullate anche tutte le altre precedentemente eseguite e la transazione si interrompe: in altre parole viene eseguito un rollback della transazione

Una caratteristica importante di una transazione è la coerenza dei dati al suo interno: anche se la transazione non è completata e i dati quindi non ancora persistiti su database, eventuali query eseguite all'interno della transazione su questi dati ritornano risultati coerenti con le operazioni compiute.

Ad esempio immaginiamo un'operazione di trasferimento di denaro dal conto A al conto B tramite un sistema online: possiamo identificare 2 macro operazioni che vanno a modificare i dati:

1. Prelievo dal conto A

2. Aggiunta sul conto B

Immaginiamo che si verifichi un errore nell'esecuzione della fase (2): la transazionalità di questo flusso ci assicura che l'intera operazione verrà annullata e il conto A rimarrà immutato. Se invece le operazioni non fossero state eseguite in modo transazionale la cifra prelevata da A sarebbe andata persa.
Inoltre la transazionalità ci assicura che eseguendo una verifica del saldo del conto A tra (1) e (2), la cifra ritornata sarà coerente con il prelievo effettuato, nonostante ancora l'operazione non sia stata effettivamente eseguita sul database.

All'interno di Liferay la transazionalità è garantita per tutte le operazioni eseguite all'interno dei service che coinvolgono entità su database: il rollback scatta quando vengono lanciate eccezioni di tipo SystemExcetion e PortalException.

Possibili problemi di coerenza di dati tra diverse transazioni

Ci possono essere casi in cui dobbiamo compiere delle operazioni in una transazione differente da quella che sta modificando i dati.

Ad esempio se implementiamo un Model Listener e nel metodo afterCreate cerchiamo di fare una findEntity dell'entità appena creata, riceveremo null invece dell'entità. Questo perché il model listener viene eseguito in una transazione separata da quella che ha generato l'entità, che potrebbe non essere ancora completata.

Oppure se usiamo il message bus per informare plugin esterni della creazione/modifica di un'entità (vedi http://marcorosce.cluster020.hosting.ovh.net/comunicazio

ne-tra-portlet-con-messagebus/) può succedere che il listener scatti prima che la transazione originaria sia completata, creando disallineamenti e problemi.

Intercettare il completamento di una transazione

Per intercettare il completamento di una transazione possiamo usare il metodo registerCallback della classe TransactionCommitCallbackRegistryUtil.

Questo metodo ci permette di registrare un task (che implementi l'interfaccia Callable) che verrà eseguito solamente al completamento della transazione.

Approfondimenti e riferimenti

- https://en.wikipedia.org/wiki/Database_transaction
- http://www.liferayaddicts.net/blogs/-/blogs/transaction-management-with-liferay-service
- http://proliferay.com/liferay-service-builder-transaction/