Transaction Propagation

1. PROPAGATION_MANDATORY

해당 메서드 호출 전에 반드시 진행 중인 트랜잭션이 존재해야 한다. 진행중인 트랜잭션이 존재하지 않을 경우에는 예외가 발생한다.

2. PROPAGATION_NESTED

이미 진행중인 트랜잭션이 존재하면 해당 메서드는 중첩된 트랜잭션(nested transaction)에서 실행되야 함을 나타낸다. 중첩된 트랜잭션은 본 트랜잭션(enclosing transaction)과 독립적으로 커밋되거나 롤백될 수 있다. 만약 본 트랜잭션이 없는 상황이라면 이는 PROPAGATION_REQUIRED와 동일하게 작동한다. 그러나 이 전파방식은 벤더 의존적이며 지원이 안 되는 경우도 많다. 관련 문서 등을 통해 중첩된 트랜잭션이 지원되는지 반드시 확인필요.

3. PROPAGATION_NEVER

해당 메서드는 트랜잭션 진행상황에서 실행될 수 없다. 만약 이미 진행 중인 트랜잭션이 존재하면 예외가 발생한다.

4. PROPAGATION_NOT_SUPPORTED

해당 메서드는 트랜잭션이 없는 상황에서 실행된다. 만약 진행 중인 트랜잭션이 있다면 해당 메서드가 반환되기 전까지 잠시 중단된다. 만약 JTATransactionManager를 사용하는 경우에는javax.transaction.TransactionManager에 대한 접근이 요구된다.

5. PROPAGATION_REQUIRED

해당 메서드는 트랜잭션 상황에서 실행되야 한다. 진행 중인 트랜잭션이 있으면 이 트랜잭션 상황에서 실행된다. 없는 경우에는 트랜잭션이 새로 시작된다.

6. PROPAGATION_REQUIRES_NEW

해당 메서드는 자신만의 트랜잭션 상황에서 실행돼야 한다. 이미 진행중인 트랜잭션이 있으면 그 트랜잭션은 해당 메서드가 반환되기 전까지 잠시 중단된다. 만약 JTATransactionManager를 사용하는 경우에는 javax.transaction.TransactionManager에 대한 접근이 요구된다.

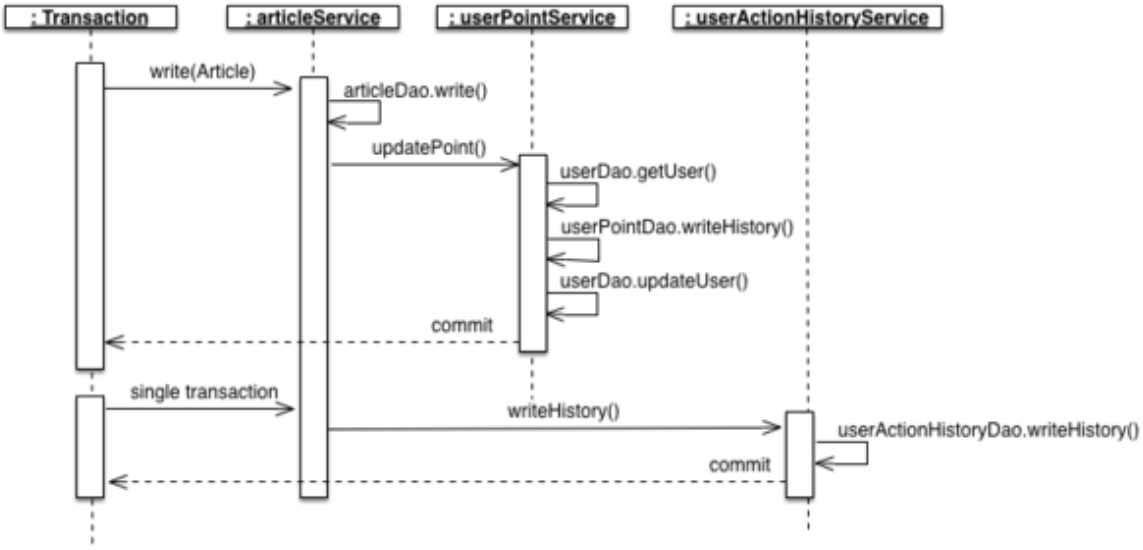
7. PROPAGATION_SUPPORTS

해당 메서드는 진행중인 트랜잭션이 없더라도 실행 가능하고, 트랜잭션이 있는 경우에는 이 트랜잭션 상황에서 실행된다

- PROPAGATION_NESTED versus PROPAGATION_REQUIRES_NEW (Juergen Hoeller)

  http://forum.spring.io/forum/spring-projects/data/7372-propagation-nested-versus-propagation-requires-new

Write Article



Write Article