Transactions

Transactions in Spring Boot

Spring Transactional Context

Transactional Context: When a method annotated with @Transactional is called, Spring creates a transactional context. This context involves database connection management, which ensures the following:

- Beginning a transaction: Before the method starts executing, Spring opens a database connection and begins a transaction.
- Committing: If the method completes successfully without throwing any exception, Spring will commit the transaction, persisting all changes to the database.
- Rolling Back: If the method throws an unchecked exception (RuntimeException or Error), the transaction is rolled back

How does it work?

Proxy-based AOP: Spring uses Aspect-Oriented Programming (AOP) with proxies to manage transactions. When you annotate a method with @Transactional, Spring wraps the method call with proxy logic:

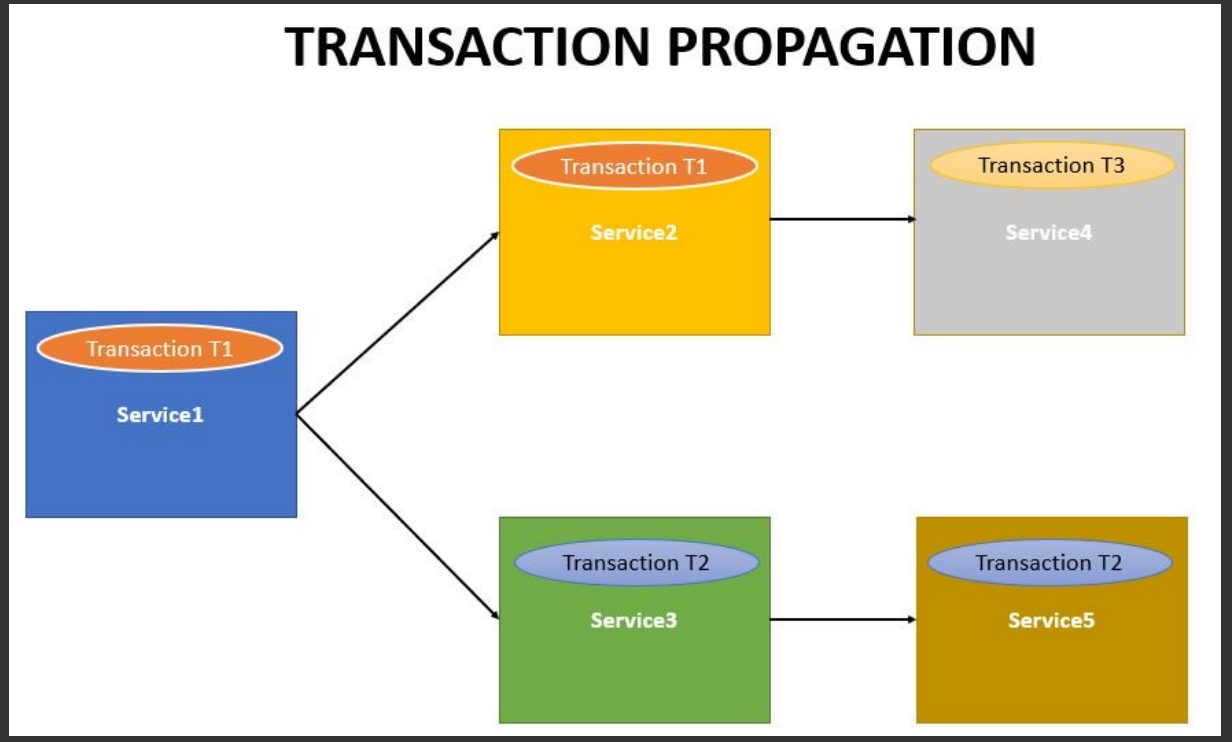
- Before entering the method, it starts a transaction.
- After the method completes, it commits or rolls back the transaction depending on whether an exception was thrown.

Transactional Propagation

There is also another important parameter of @Transactional: propagation.

Propagation can be set to REQUIRED, REQUIRES_NEW, MANDATORY, SUPPORTS, NOT_SUPPORTED, NESTED or NEVER.

```
@Transactional
(propagation=Propagation.REQUIRED)
```



Transactional Propagation

Propagation	Calling method (outer)	Called method (inner)
REQUIRED	No	T2
REQUIRED	T1	T1
REQUIRES_NEW	No	T2
REQUIRES_NEW	T1	T2
MANDATORY	No	Exception
MANDATORY	T1	T1
NOT_SUPPORTED	No	No
NOT_SUPPORTED	T1	No
SUPPORTS	No	No
SUPPORTS	T1	T1
NEVER	No	No
NEVER	T1	Exception
NESTED	No	T2
NESTED	T1	T2

Transactional Isolation in Spring

In Spring it is possible to set one of the 5 isolation level values: DEFAULT, READ_UNCOMMITTED, READ_COMMITED, REPETABLE_READ and SERIALIZABLE.

```
@Transactional (isolation=Isolation.READ_COMMITED)
```

Table 13.1. Transaction Isolation Levels

Isolation Level	Dirty Read	Nonrepeatable Read	Phantom Read	Serialization Anomaly
Read uncommitted	Allowed, but not in PG	Possible	Possible	Possible
Read committed	Not possible	Possible	Possible	Possible
Repeatable read	Not possible	Not possible	Allowed, but not in PG	Possible
Serializable	Not possible	Not possible	Not possible	Not possible