Spring Professional Exam Tutorial v5.0

Question 10

Question 10 - How are you going to define a transaction in Spring? What does @Transactional do? What is the PlatformTransactionManager?

To use transactions in Spring Framework, you need to:

- Enable transaction management by using `@EnableTransactionManagement` annotation on top of your Configuration class
- Create bean method in configuration class that will return bean implementing interface `PlatformTransactionManager`, examples of transactions managers:
  - `DataSourceTransactionManager`
  - `JtaTransactionManager`
  - `JpaTransactionManager`
  - …
- Use `@Transactional` annotation on top of classes or methods that should involve transaction management

Question 10 - How are you going to define a transaction in Spring? What does @Transactional do? What is the PlatformTransactionManager?

`@Transactional` annotation can be used on top of classes or methods to enable transaction management for entire class or specified methods. When method with `@Transactional` annotation is called, invocation is proxied by `TransactionInterceptor` and `TransactionAspectSupport` which are using `PlatformTransactionManager` to manage transaction.

Transaction is being started at the beginning of the method (if none transaction exists), and it is being committed at the end of successful execution. Transaction can be rolled back upon exception being thrown. This behavior is dependent on transaction propagation type.

Question 10 - How are you going to define a transaction in Spring? What does @Transactional do? What is the PlatformTransactionManager?

`@Transactional` annotation allows you to configure following attributes:

- Transaction Manager
- Propagation Type
- Isolation Level
- Timeout for Transaction
- Read Only Flag
- Define which exception types will cause transaction rollback
- Define which exception types will not cause transaction rollback

Question 10 - How are you going to define a transaction in Spring? What does @Transactional do? What is the PlatformTransactionManager?

`PlatformTransactionManager` is an interface that is used by declarative Spring's AOP Transaction Management to create, commit and rollback transactions.

`PlatformTransactionManager` contains following methods:

- getTransaction – returns currently active transaction or creates new one
- commit – commits transaction, or rolls back transaction if it was marked for rollback
- rollback – performs rollback of transaction