
Spring Certified #15



A question lead guide to prepare Spring certification



Data Management

Which of the following transaction propagation behaviors would join the current transaction if one exists, or create a new transaction otherwise?
(select 2)

- **MANDATORY**
- **NESTED**
- **REQUIRED**
- **REQUIRES_NEW**

NESTED & REQUIRED

enum **Propagation** extends [Enum](#)<[Propagation](#)>

Enumeration that represents transaction propagation behaviors for use with the [Transactional](#) annotation, corresponding to the [TransactionDefinition](#) interface.

MANDATORY

Support a current transaction, and throw an exception if none exists.

NESTED

Execute within a nested transaction if a current transaction exists, behave like **REQUIRED** otherwise.

NEVER

Execute non-transactionally, and throw an exception if a transaction exists.

NOT_SUPPORTED

Execute non-transactionally, and suspend the current transaction if one exists.

REQUIRED

Support a current transaction, and create a new one if none exists.

REQUIRES_NEW

Create a new transaction, and suspend the current transaction if one exists.

SUPPORTS

Support a current transaction, and execute non-transactionally if none exists.

<https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/transaction/annotation/Propagation.html>



Spring Core

SpEL expressions begin with the `#` symbol and are wrapped in braces:
`{expression}`.

Properties can be referenced in a similar fashion, starting with which symbol and wrapped in braces: `X{property.name}` , X being the symbol?

- `@`
- `#`
- `$`
- `!`

\$

SpEL expressions begin with the `#` symbol and are wrapped in braces: `#{expression}`.

Properties can be referenced in a similar fashion, starting with a `$` symbol and wrapped in braces: `${property.name}`.

Property placeholders cannot contain SpEL expressions, but expressions can contain property references:

`#{${someProperty} + 2}`

In the example above, assume *someProperty* has value 2, so the resulting expression would be `2 + 2`, which would be evaluated to 4.

<https://www.baeldung.com/spring-expression-language#operators>

<https://docs.spring.io/spring-framework/docs/4.2.x/spring-framework-reference/html/expressions.html>

Testing

Which of the following is a meta-annotation of `@SpringBootTest`?

- `@ExtendWith(SpringExtension.class)`
- `@RunWith(SpringExtension.class)`
- `@Target(METHOD)`
- `@Retention(SOURCE)`

@ExtendWith(SpringExtension.class)

Package [org.springframework.boot.test.context](#)

Annotation Interface SpringBootTest

[@Target\(TYPE\)](#) [@Retention\(RUNTIME\)](#) [@Documented](#) [@Inherited](#) [@BootstrapWith\(SpringBootTestContextBootstrapper.class\)](#)
[@ExtendWith\(org.springframework.test.context.junit.jupiter.SpringExtension.class\)](#) public @interface **SpringBootTest**

Annotation that can be specified on a test class that runs Spring Boot based tests. Provides the following features over and above the regular *Spring TestContext Framework*:

- Uses [SpringBootTestContextLoader](#) as the default [ContextLoader](#) when no specific [@ContextConfiguration\(loader=...\)](#) is defined.
- Automatically searches for a [@SpringBootTestConfiguration](#) when nested [@Configuration](#) is not used, and no explicit [classes](#) are specified.
- Allows custom [Environment](#) properties to be defined using the [properties attribute](#).
- Allows application arguments to be defined using the [args attribute](#).
- Provides support for different [webEnvironment](#) modes, including the ability to start a fully running web server listening on a [defined](#) or [random](#) port.
- Registers a [TestRestTemplate](#) and/or [WebTestClient](#) bean for use in web tests that are using a fully running web server.

<https://docs.spring.io/spring-boot/docs/current/api/org/springframework/boot/test/context/SpringBootTest.html>



<https://bit.ly/2v7222>



<https://spring-book.mystrikingly.com>