Spring Framework

OVERVIEW PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | ENUM CONSTANTS | FIELD | METHOD DETAIL: ENUM CONSTANTS | FIELD | METHOD

org.springframework.transaction.annotation

Enum Propagation

java.lang.Object
  java.lang.Enum<Propagation>
    org.springframework.transaction.annotation.Propagation

All Implemented Interfaces:

`Serializable, Comparable<Propagation>`

```
public enum Propagation
extends Enum<Propagation>
```

Enumeration that represents transaction propagation behaviors for use with the `Transactional` annotation, corresponding to the `TransactionDefinition` interface.

Since:

`1.2`

Author:

`Colin Sampaleanu, Juergen Hoeller`

Enum Constant Summary

Enum Constants

Enum Constant and Description
`MANDATORY` Support a current transaction, throw an exception if none exists.
`NESTED` Execute within a nested transaction if a current transaction exists, behave like PROPAGATION_REQUIRED else.
`NEVER` Execute non-transactionally, throw an exception if a transaction exists.
`NOT_SUPPORTED` Execute non-transactionally, suspend the current transaction if one exists.
`REQUIRED` Support a current transaction, 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, execute non-transactionally if none exists.

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`int`	`value()`
`static Propagation`	`valueOf(String name)` Returns the enum constant of this type with the specified name.
`static Propagation[]`	`values()` Returns an array containing the constants of this enum type, in the order they are declared.

Methods inherited from class java.lang.Enum

`clone, compareTo, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf`

Methods inherited from class java.lang.Object

`getClass, notify, notifyAll, wait, wait, wait`

Enum Constant Detail

REQUIRED

```
public static final Propagation REQUIRED
```

Support a current transaction, create a new one if none exists. Analogous to EJB transaction attribute of the same name.

This is the default setting of a transaction annotation.

SUPPORTS

```
public static final Propagation SUPPORTS
```

Support a current transaction, execute non-transactionally if none exists. Analogous to EJB transaction attribute of the same name.

Note: For transaction managers with transaction synchronization, PROPAGATION_SUPPORTS is slightly different from no transaction at all, as it defines a transaction scope that synchronization will apply for. As a consequence, the same resources (JDBC Connection, Hibernate Session, etc) will be shared for the entire specified scope. Note that this depends on the actual synchronization configuration of the transaction manager.

See Also:

`AbstractPlatformTransactionManager.setTransactionSynchronization(int)`

MANDATORY

```
public static final Propagation MANDATORY
```

Support a current transaction, throw an exception if none exists. Analogous to EJB transaction attribute of the same name.

REQUIRES_NEW

```
public static final Propagation REQUIRES_NEW
```

Create a new transaction, and suspend the current transaction if one exists. Analogous to the EJB transaction attribute of the same name.

NOTE: Actual transaction suspension will not work out-of-the-box on all transaction managers. This in particular applies to `JtaTransactionManager`, which requires the `javax.transaction.TransactionManager` to be made available it to it (which is server-specific in standard Java EE).

See Also:

`JtaTransactionManager.setTransactionManager(javax.transaction.TransactionManager)`

NOT_SUPPORTED

```
public static final Propagation NOT_SUPPORTED
```

Execute non-transactionally, suspend the current transaction if one exists. Analogous to EJB transaction attribute of the same name.

NOTE: Actual transaction suspension will not work out-of-the-box on all transaction managers. This in particular applies to `JtaTransactionManager`, which requires the `javax.transaction.TransactionManager` to be made available it to it (which is server-specific in standard Java EE).

See Also:

`JtaTransactionManager.setTransactionManager(javax.transaction.TransactionManager)`

NEVER

```
public static final Propagation NEVER
```

Execute non-transactionally, throw an exception if a transaction exists. Analogous to EJB transaction attribute of the same name.

NESTED

```
public static final Propagation NESTED
```

Execute within a nested transaction if a current transaction exists, behave like PROPAGATION_REQUIRED else. There is no analogous feature in EJB.

Note: Actual creation of a nested transaction will only work on specific transaction managers. Out of the box, this only applies to the JDBC DataSourceTransactionManager when working on a JDBC 3.0 driver. Some JTA providers might support nested transactions as well.

See Also:

`DataSourceTransactionManager`

Method Detail

values

```
public static Propagation[] values()
```

Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows:

```
for (Propagation c : Propagation.values())
    System.out.println(c);
```

Returns:

`an array containing the constants of this enum type, in the order they are declared`

valueOf

```
public static Propagation valueOf(String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

`name - the name of the enum constant to be returned.`

Returns:

`the enum constant with the specified name`

Throws:

`IllegalArgumentException - if this enum type has no constant with the specified name`

`NullPointerException - if the argument is null`

value

```
public int value()
```

PREV CLASS **NEXT CLASS** FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | ENUM CONSTANTS | FIELD | METHOD DETAIL: ENUM CONSTANTS | FIELD | METHOD