

PREV CLASSNEXT CLASSFRAMESNO FRAMESALL CLASSES

SUMMARY: FIELD | REQUIRED | OPTIONALDETAIL: FIELD | ELEMENT

org.springframework.transaction.annotation

Annotation Type Transactional

```
@Target(value={METHOD,TYPE})
@Retention(value=RUNTIME)
@Inherited
@Documented
public @interface Transactional
```

Describes transaction attributes on a method or class.

This annotation type is generally directly comparable to Spring's `RuleBasedTransactionAttribute` class, and in fact `AnnotationTransactionAttributeSource` will directly convert the data to the latter class, so that Spring's transaction support code does not have to know about annotations. If no rules are relevant to the exception, it will be treated like `DefaultTransactionAttribute` (rolling back on `RuntimeException` and `Error` but not on checked exceptions).

For specific information about the semantics of this annotation's attributes, consult the `TransactionDefinition` and `TransactionAttribute` javadocs.

Since:
1.2

Author:
Colin Sampaleanu, Juergen Hoeller, Sam Brannen

See Also:
`TransactionAttribute`, `DefaultTransactionAttribute`, `RuleBasedTransactionAttribute`

Optional Element Summary

Optional Elements	
Modifier and Type	Optional Element and Description
<code>Isolation</code>	<code>isolation</code> The transaction isolation level.
<code>Class<? extends Throwable>[]</code>	<code>noRollbackFor</code> Defines zero (0) or more exception Classes , which must be subclasses of <code>Throwable</code> , indicating which exception types must not cause a transaction rollback.
<code>String[]</code>	<code>noRollbackForClassName</code> Defines zero (0) or more exception names (for exceptions which must be a subclass of <code>Throwable</code>) indicating which exception types must not cause a transaction rollback.

Propagation	propagation The transaction propagation type.
boolean	readOnly true if the transaction is read-only.
Class <? extends Throwable >[]	rollbackFor Defines zero (0) or more exception classes , which must be subclasses of Throwable , indicating which exception types must cause a transaction rollback.
String []	rollbackForClassName Defines zero (0) or more exception names (for exceptions which must be a subclass of Throwable), indicating which exception types must cause a transaction rollback.
int	timeout The timeout for this transaction.
String	transactionManager A <i>qualifier</i> value for the specified transaction.
String	value Alias for transactionManager() .

Element Detail

value

```
@AliasFor(value="transactionManager")
public abstract String value
```

Alias for `transactionManager()`.

See Also:
`transactionManager()`

Default:
""

transactionManager

```
@AliasFor(value="value")
public abstract String transactionManager
```

A *qualifier* value for the specified transaction.

May be used to determine the target transaction manager, matching the qualifier value (or the bean name) of a specific `PlatformTransactionManager` bean definition.

Since:
4.2

See Also:`value()`**Default:**`""`**propagation**

```
public abstract Propagation propagation
```

The transaction propagation type.

Defaults to `Propagation.REQUIRED`.

See Also:`TransactionDefinition.getPropagationBehavior()`**Default:**`org.springframework.transaction.annotation.Propagation.REQUIRED`**isolation**

```
public abstract Isolation isolation
```

The transaction isolation level.

Defaults to `Isolation.DEFAULT`.

See Also:`TransactionDefinition.getIsolationLevel()`**Default:**`org.springframework.transaction.annotation.Isolation.DEFAULT`**timeout**

```
public abstract int timeout
```

The timeout for this transaction.

Defaults to the default timeout of the underlying transaction system.

See Also:`TransactionDefinition.getTimeout()`**Default:**`-1`

readOnly

```
public abstract boolean readOnly
```

true if the transaction is read-only.

Defaults to false.

This just serves as a hint for the actual transaction subsystem; it will *not necessarily* cause failure of write access attempts. A transaction manager which cannot interpret the read-only hint will *not* throw an exception when asked for a read-only transaction but rather silently ignore the hint.

See Also:

`TransactionDefinition.isReadOnly()`

Default:

false

rollbackFor

```
public abstract Class<? extends Throwable>[] rollbackFor
```

Defines zero (0) or more exception `classes`, which must be subclasses of `Throwable`, indicating which exception types must cause a transaction rollback.

By default, a transaction will be rolling back on `RuntimeException` and `Error` but not on checked exceptions (business exceptions). See `DefaultTransactionAttribute.rollbackOn(Throwable)` for a detailed explanation.

This is the preferred way to construct a rollback rule (in contrast to `rollbackForClassName()`), matching the exception class and its subclasses.

Similar to `RollbackRuleAttribute.RollbackRuleAttribute(Class clazz)`.

See Also:

`rollbackForClassName()`, `DefaultTransactionAttribute.rollbackOn(Throwable)`

Default:

```
{}
```

rollbackForClassName

```
public abstract String[] rollbackForClassName
```

Defines zero (0) or more exception names (for exceptions which must be a subclass of `Throwable`), indicating which exception types must cause a transaction rollback.

This can be a substring of a fully qualified class name, with no wildcard support at present. For example, a value of "ServletException" would match `javax.servlet.ServletException` and its

subclasses.

NB: Consider carefully how specific the pattern is and whether to include package information (which isn't mandatory). For example, "Exception" will match nearly anything and will probably hide other rules. "java.lang.Exception" would be correct if "Exception" were meant to define a rule for all checked exceptions. With more unusual `Exception` names such as "BaseBusinessException" there is no need to use a FQN.

Similar to `RollbackRuleAttribute.RollbackRuleAttribute(String exceptionName)`.

See Also:

`rollbackFor()`, `DefaultTransactionAttribute.rollbackOn(Throwable)`

Default:

`{}`

noRollbackFor

```
public abstract Class<? extends Throwable>[] noRollbackFor
```

Defines zero (0) or more exception `Classes`, which must be subclasses of `Throwable`, indicating which exception types must **not** cause a transaction rollback.

This is the preferred way to construct a rollback rule (in contrast to `noRollbackForClassName()`), matching the exception class and its subclasses.

Similar to `NoRollbackRuleAttribute.NoRollbackRuleAttribute(Class clazz)`.

See Also:

`noRollbackForClassName()`, `DefaultTransactionAttribute.rollbackOn(Throwable)`

Default:

`{}`

noRollbackForClassName

```
public abstract String[] noRollbackForClassName
```

Defines zero (0) or more exception names (for exceptions which must be a subclass of `Throwable`) indicating which exception types must **not** cause a transaction rollback.

See the description of `rollbackForClassName()` for further information on how the specified names are treated.

Similar to `NoRollbackRuleAttribute.NoRollbackRuleAttribute(String exceptionName)`.

See Also:

`noRollbackFor()`, `DefaultTransactionAttribute.rollbackOn(Throwable)`

Default:

`{}`

OVERVIEW PACKAGE CLASS TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: FIELD | REQUIRED | OPTIONAL DETAIL: FIELD | ELEMENT