

org.springframework.transaction.support

Class TransactionTemplate

java.lang.Object
 org.springframework.transaction.support.DefaultTransactionDefinition
 org.springframework.transaction.support.TransactionTemplate

All Implemented Interfaces:
Serializable, InitializingBean, TransactionOperations, TransactionDefinition

```
public class TransactionTemplate
extends DefaultTransactionDefinition
implements TransactionOperations, InitializingBean
```

Template class that simplifies programmatic transaction demarcation and transaction exception handling.

The central method is `execute(org.springframework.transaction.support.TransactionCallback<T>)`, supporting transactional code that implements the `TransactionCallback` interface. This template handles the transaction lifecycle and possible exceptions such that neither the `TransactionCallback` implementation nor the calling code needs to explicitly handle transactions.

Typical usage: Allows for writing low-level data access objects that use resources such as JDBC DataSources but are not transaction-aware themselves. Instead, they can implicitly participate in transactions handled by higher-level application services utilizing this class, making calls to the low-level services via an inner-class callback object.

Can be used within a service implementation via direct instantiation with a transaction manager reference, or get prepared in an application context and passed to services as bean reference. Note: The transaction manager should always be configured as bean in the application context: in the first case given to the service directly, in the second case given to the prepared template.

Supports setting the propagation behavior and the isolation level by name, for convenient configuration in context definitions.

Since:
17.03.2003

Author:
Juergen Hoeller

See Also:
`execute(org.springframework.transaction.support.TransactionCallback<T>)`,
`setTransactionManager(org.springframework.transaction.PlatformTransactionManager)`,
`PlatformTransactionManager`, Serialized Form

Field Summary

Fields	
Modifier and Type	Field and Description
protected Log	logger Logger available to subclasses

Fields inherited from
class `org.springframework.transaction.support.DefaultTransactionDefinition`

`PREFIX_ISOLATION`, `PREFIX_PROPAGATION`, `PREFIX_TIMEOUT`, `READ_ONLY_MARKER`

Fields inherited from interface `org.springframework.transaction.TransactionDefinition`

`ISOLATION_DEFAULT`, `ISOLATION_READ_COMMITTED`, `ISOLATION_READ_UNCOMMITTED`, `ISOLATION_REPEATABLE_READ`, `ISOLATION_SERIALIZABLE`, `PROPAGATION_MANDATORY`, `PROPAGATION_NESTED`, `PROPAGATION_NEVER`, `PROPAGATION_NOT_SUPPORTED`, `PROPAGATION_REQUIRED`, `PROPAGATION_REQUIRES_NEW`, `PROPAGATION_SUPPORTS`, `TIMEOUT_DEFAULT`

Constructor Summary

Constructors

Constructor and Description

`TransactionTemplate()`

Construct a new TransactionTemplate for bean usage.

`TransactionTemplate(PlatformTransactionManager transactionManager)`

Construct a new TransactionTemplate using the given transaction manager.

`TransactionTemplate(PlatformTransactionManager transactionManager, TransactionDefinition transactionDefinition)`

Construct a new TransactionTemplate using the given transaction manager, taking its default settings from the given transaction definition.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
void	<code>afterPropertiesSet()</code> Invoked by a BeanFactory after it has set all bean properties supplied (and satisfied BeanFactoryAware and ApplicationContextAware).
<T> T	<code>execute(TransactionCallback<T> action)</code> Execute the action specified by the given callback object within a transaction.
<code>PlatformTransactionManager</code>	<code>getTransactionManager()</code> Return the transaction management strategy to be used.
void	<code>setTransactionManager(PlatformTransactionManager transactionManager)</code> Set the transaction management strategy to be used.

Methods inherited from
class `org.springframework.transaction.support.DefaultTransactionDefinition`

`equals`, `getDefinitionDescription`, `getIsolationLevel`, `getName`, `getPropagationBehavior`, `getTimeout`, `hashCode`, `isReadOnly`, `setIsolationLevel`, `setIsolationLevelName`, `setName`, `setPropagationBehavior`, `setPropagationBehaviorName`, `setReadOnly`, `setTimeout`, `toString`

Methods inherited from class `java.lang.Object`

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Field Detail

`logger`

protected final `Log` logger

Logger available to subclasses

Constructor Detail

`TransactionTemplate`

```
public TransactionTemplate()
```

Construct a new `TransactionTemplate` for bean usage.

Note: The `PlatformTransactionManager` needs to be set before any execute calls.

See Also:

`setTransactionManager(org.springframework.transaction.PlatformTransactionManager)`

`TransactionTemplate`

```
public TransactionTemplate(PlatformTransactionManager transactionManager)
```

Construct a new `TransactionTemplate` using the given transaction manager.

Parameters:

`transactionManager` - the transaction management strategy to be used

`TransactionTemplate`

```
public TransactionTemplate(PlatformTransactionManager transactionManager,  
                           TransactionDefinition transactionDefinition)
```

Construct a new `TransactionTemplate` using the given transaction manager, taking its default settings from the given transaction definition.

Parameters:

`transactionManager` - the transaction management strategy to be used

`transactionDefinition` - the transaction definition to copy the default settings from.
Local properties can still be set to change values.

Method Detail

setTransactionManager

```
public void setTransactionManager(PlatformTransactionManager transactionManager)
```

Set the transaction management strategy to be used.

getTransactionManager

```
public PlatformTransactionManager getTransactionManager()
```

Return the transaction management strategy to be used.

afterPropertiesSet

```
public void afterPropertiesSet()
```

Description copied from interface: [InitializingBean](#)

Invoked by a BeanFactory after it has set all bean properties supplied (and satisfied BeanFactoryAware and ApplicationContextAware).

This method allows the bean instance to perform initialization only possible when all bean properties have been set and to throw an exception in the event of misconfiguration.

Specified by:

[afterPropertiesSet](#) in interface [InitializingBean](#)

execute

```
public <T> T execute(TransactionCallback<T> action)  
    throws TransactionException
```

Description copied from interface: [TransactionOperations](#)

Execute the action specified by the given callback object within a transaction.

Allows for returning a result object created within the transaction, that is, a domain object or a collection of domain objects. A RuntimeException thrown by the callback is treated as a fatal exception that enforces a rollback. Such an exception gets propagated to the caller of the template.

Specified by:

[execute](#) in interface [TransactionOperations](#)

Parameters:

action - the callback object that specifies the transactional action

Returns:

a result object returned by the callback, or null if none

Throws:

[TransactionException](#) - in case of initialization, rollback, or system errors