| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RelationTypeSupport.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/relation/Role.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/relation/RelationTypeSupport.html)    [**NO FRAMES**](http://docs.google.com/RelationTypeSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **javax.management.relation**

Class RelationTypeSupport

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.management.relation.RelationTypeSupport**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [RelationType](http://docs.google.com/javax/management/relation/RelationType.html)

public class **RelationTypeSupport**extends [Object](http://docs.google.com/java/lang/Object.html)implements [RelationType](http://docs.google.com/javax/management/relation/RelationType.html)

A RelationTypeSupport object implements the RelationType interface.

It represents a relation type, providing role information for each role expected to be supported in every relation of that type.

A relation type includes a relation type name and a list of role infos (represented by RoleInfo objects).

A relation type has to be declared in the Relation Service:

- either using the createRelationType() method, where a RelationTypeSupport object will be created and kept in the Relation Service

- either using the addRelationType() method where the user has to create an object implementing the RelationType interface, and this object will be used as representing a relation type in the Relation Service.

The **serialVersionUID** of this class is 4611072955724144607L.

**Since:** 1.5 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.management.relation.RelationTypeSupport)

| **Constructor Summary** | |
| --- | --- |
| protected | [**RelationTypeSupport**](http://docs.google.com/javax/management/relation/RelationTypeSupport.html#RelationTypeSupport(java.lang.String))([String](http://docs.google.com/java/lang/String.html) relationTypeName)            Constructor to be used for subclasses. |
|  | [**RelationTypeSupport**](http://docs.google.com/javax/management/relation/RelationTypeSupport.html#RelationTypeSupport(java.lang.String,%20javax.management.relation.RoleInfo%5B%5D))([String](http://docs.google.com/java/lang/String.html) relationTypeName, [RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html)[] roleInfoArray)            Constructor where all role definitions are dynamically created and passed as parameter. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**addRoleInfo**](http://docs.google.com/javax/management/relation/RelationTypeSupport.html#addRoleInfo(javax.management.relation.RoleInfo))([RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html) roleInfo)            Add a role info. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRelationTypeName**](http://docs.google.com/javax/management/relation/RelationTypeSupport.html#getRelationTypeName())()            Returns the relation type name. |
| [RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html) | [**getRoleInfo**](http://docs.google.com/javax/management/relation/RelationTypeSupport.html#getRoleInfo(java.lang.String))([String](http://docs.google.com/java/lang/String.html) roleInfoName)            Returns the role info (RoleInfo object) for the given role info name (null if not found). |
| [List](http://docs.google.com/java/util/List.html)<[RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html)> | [**getRoleInfos**](http://docs.google.com/javax/management/relation/RelationTypeSupport.html#getRoleInfos())()            Returns the list of role definitions (ArrayList of RoleInfo objects). |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### RelationTypeSupport

public **RelationTypeSupport**([String](http://docs.google.com/java/lang/String.html) relationTypeName,  
 [RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html)[] roleInfoArray)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [InvalidRelationTypeException](http://docs.google.com/javax/management/relation/InvalidRelationTypeException.html)

Constructor where all role definitions are dynamically created and passed as parameter.

**Parameters:**relationTypeName - Name of relation typeroleInfoArray - List of role definitions (RoleInfo objects) **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null parameter [InvalidRelationTypeException](http://docs.google.com/javax/management/relation/InvalidRelationTypeException.html) - if:

- the same name has been used for two different roles

- no role info provided

- one null role info provided

### RelationTypeSupport

protected **RelationTypeSupport**([String](http://docs.google.com/java/lang/String.html) relationTypeName)

Constructor to be used for subclasses.

**Parameters:**relationTypeName - Name of relation type. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null parameter.

| **Method Detail** |
| --- |

### getRelationTypeName

public [String](http://docs.google.com/java/lang/String.html) **getRelationTypeName**()

Returns the relation type name.

**Specified by:**[getRelationTypeName](http://docs.google.com/javax/management/relation/RelationType.html#getRelationTypeName()) in interface [RelationType](http://docs.google.com/javax/management/relation/RelationType.html) **Returns:**the relation type name.

### getRoleInfos

public [List](http://docs.google.com/java/util/List.html)<[RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html)> **getRoleInfos**()

Returns the list of role definitions (ArrayList of RoleInfo objects).

**Specified by:**[getRoleInfos](http://docs.google.com/javax/management/relation/RelationType.html#getRoleInfos()) in interface [RelationType](http://docs.google.com/javax/management/relation/RelationType.html) **Returns:**an [ArrayList](http://docs.google.com/java/util/ArrayList.html) of [RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html).

### getRoleInfo

public [RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html) **getRoleInfo**([String](http://docs.google.com/java/lang/String.html) roleInfoName)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [RoleInfoNotFoundException](http://docs.google.com/javax/management/relation/RoleInfoNotFoundException.html)

Returns the role info (RoleInfo object) for the given role info name (null if not found).

**Specified by:**[getRoleInfo](http://docs.google.com/javax/management/relation/RelationType.html#getRoleInfo(java.lang.String)) in interface [RelationType](http://docs.google.com/javax/management/relation/RelationType.html) **Parameters:**roleInfoName - role info name **Returns:**RoleInfo object providing role definition does not exist **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null parameter [RoleInfoNotFoundException](http://docs.google.com/javax/management/relation/RoleInfoNotFoundException.html) - if no role info with that name in relation type.

### addRoleInfo

protected void **addRoleInfo**([RoleInfo](http://docs.google.com/javax/management/relation/RoleInfo.html) roleInfo)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [InvalidRelationTypeException](http://docs.google.com/javax/management/relation/InvalidRelationTypeException.html)

Add a role info. This method of course should not be used after the creation of the relation type, because updating it would invalidate that the relations created associated to that type still conform to it. Can throw a RuntimeException if trying to update a relation type declared in the Relation Service.

**Parameters:**roleInfo - role info to be added. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null parameter. [InvalidRelationTypeException](http://docs.google.com/javax/management/relation/InvalidRelationTypeException.html) - if there is already a role info in current relation type with the same name.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RelationTypeSupport.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/relation/Role.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/relation/RelationTypeSupport.html)    [**NO FRAMES**](http://docs.google.com/RelationTypeSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).