| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Relation.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/MBeanServerNotificationFilter.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/relation/RelationException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/relation/Relation.html)    [**NO FRAMES**](http://docs.google.com/Relation.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.management.relation**

Interface Relation

**All Known Subinterfaces:** [RelationSupportMBean](http://docs.google.com/javax/management/relation/RelationSupportMBean.html) **All Known Implementing Classes:** [RelationSupport](http://docs.google.com/javax/management/relation/RelationSupport.html)

public interface **Relation**

This interface has to be implemented by any MBean class expected to represent a relation managed using the Relation Service.

Simple relations, i.e. having only roles, no properties or methods, can be created directly by the Relation Service (represented as RelationSupport objects, internally handled by the Relation Service).

If the user wants to represent more complex relations, involving properties and/or methods, he has to provide his own class implementing the Relation interface. This can be achieved either by inheriting from RelationSupport class, or by implementing the interface (fully or delegation to a RelationSupport object member).

Specifying such user relation class is to introduce properties and/or methods. Those have to be exposed for remote management. So this means that any user relation class must be a MBean class.

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| [RoleResult](http://docs.google.com/javax/management/relation/RoleResult.html) | [**getAllRoles**](http://docs.google.com/javax/management/relation/Relation.html#getAllRoles())()            Returns all roles present in the relation. |
| [Map](http://docs.google.com/java/util/Map.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html),[List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)>> | [**getReferencedMBeans**](http://docs.google.com/javax/management/relation/Relation.html#getReferencedMBeans())()            Retrieves MBeans referenced in the various roles of the relation. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRelationId**](http://docs.google.com/javax/management/relation/Relation.html#getRelationId())()            Returns relation identifier (used to uniquely identify the relation inside the Relation Service). |
| [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**getRelationServiceName**](http://docs.google.com/javax/management/relation/Relation.html#getRelationServiceName())()            Returns ObjectName of the Relation Service handling the relation. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRelationTypeName**](http://docs.google.com/javax/management/relation/Relation.html#getRelationTypeName())()            Returns name of associated relation type. |
| [List](http://docs.google.com/java/util/List.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html)> | [**getRole**](http://docs.google.com/javax/management/relation/Relation.html#getRole(java.lang.String))([String](http://docs.google.com/java/lang/String.html) roleName)            Retrieves role value for given role name. |
| [Integer](http://docs.google.com/java/lang/Integer.html) | [**getRoleCardinality**](http://docs.google.com/javax/management/relation/Relation.html#getRoleCardinality(java.lang.String))([String](http://docs.google.com/java/lang/String.html) roleName)            Returns the number of MBeans currently referenced in the given role. |
| [RoleResult](http://docs.google.com/javax/management/relation/RoleResult.html) | [**getRoles**](http://docs.google.com/javax/management/relation/Relation.html#getRoles(java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] roleNameArray)            Retrieves values of roles with given names. |
| void | [**handleMBeanUnregistration**](http://docs.google.com/javax/management/relation/Relation.html#handleMBeanUnregistration(javax.management.ObjectName,%20java.lang.String))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName, [String](http://docs.google.com/java/lang/String.html) roleName)            Callback used by the Relation Service when a MBean referenced in a role is unregistered. |
| [RoleList](http://docs.google.com/javax/management/relation/RoleList.html) | [**retrieveAllRoles**](http://docs.google.com/javax/management/relation/Relation.html#retrieveAllRoles())()            Returns all roles in the relation without checking read mode. |
| void | [**setRole**](http://docs.google.com/javax/management/relation/Relation.html#setRole(javax.management.relation.Role))([Role](http://docs.google.com/javax/management/relation/Role.html) role)            Sets the given role. |
| [RoleResult](http://docs.google.com/javax/management/relation/RoleResult.html) | [**setRoles**](http://docs.google.com/javax/management/relation/Relation.html#setRoles(javax.management.relation.RoleList))([RoleList](http://docs.google.com/javax/management/relation/RoleList.html) roleList)            Sets the given roles. |

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

### getRole

[List](http://docs.google.com/java/util/List.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html)> **getRole**([String](http://docs.google.com/java/lang/String.html) roleName)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html),  
 [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html)

Retrieves role value for given role name.

Checks if the role exists and is readable according to the relation type.

**Parameters:**roleName - name of role **Returns:**the ArrayList of ObjectName objects being the role value **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null role name [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html) - if:

- there is no role with given name

- the role is not readable.

[RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html) - if the Relation Service is not registered in the MBean Server**See Also:**[setRole(javax.management.relation.Role)](http://docs.google.com/javax/management/relation/Relation.html#setRole(javax.management.relation.Role))

### getRoles

[RoleResult](http://docs.google.com/javax/management/relation/RoleResult.html) **getRoles**([String](http://docs.google.com/java/lang/String.html)[] roleNameArray)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html)

Retrieves values of roles with given names.

Checks for each role if it exists and is readable according to the relation type.

**Parameters:**roleNameArray - array of names of roles to be retrieved **Returns:**a RoleResult object, including a RoleList (for roles successfully retrieved) and a RoleUnresolvedList (for roles not retrieved). **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null role name [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html) - if the Relation Service is not registered in the MBean Server**See Also:**[setRoles(javax.management.relation.RoleList)](http://docs.google.com/javax/management/relation/Relation.html#setRoles(javax.management.relation.RoleList))

### getRoleCardinality

[Integer](http://docs.google.com/java/lang/Integer.html) **getRoleCardinality**([String](http://docs.google.com/java/lang/String.html) roleName)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html)

Returns the number of MBeans currently referenced in the given role.

**Parameters:**roleName - name of role **Returns:**the number of currently referenced MBeans in that role **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null role name [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html) - if there is no role with given name

### getAllRoles

[RoleResult](http://docs.google.com/javax/management/relation/RoleResult.html) **getAllRoles**()  
 throws [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html)

Returns all roles present in the relation.

**Returns:**a RoleResult object, including a RoleList (for roles successfully retrieved) and a RoleUnresolvedList (for roles not readable). **Throws:** [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html) - if the Relation Service is not registered in the MBean Server

### retrieveAllRoles

[RoleList](http://docs.google.com/javax/management/relation/RoleList.html) **retrieveAllRoles**()

Returns all roles in the relation without checking read mode.

**Returns:**a RoleList.

### setRole

void **setRole**([Role](http://docs.google.com/javax/management/relation/Role.html) role)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html),  
 [RelationTypeNotFoundException](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html),  
 [InvalidRoleValueException](http://docs.google.com/javax/management/relation/InvalidRoleValueException.html),  
 [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html),  
 [RelationNotFoundException](http://docs.google.com/javax/management/relation/RelationNotFoundException.html)

Sets the given role.

Will check the role according to its corresponding role definition provided in relation's relation type

Will send a notification (RelationNotification with type RELATION\_BASIC\_UPDATE or RELATION\_MBEAN\_UPDATE, depending if the relation is a MBean or not).

**Parameters:**role - role to be set (name and new value) **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null role [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html) - if there is no role with the supplied role's name or if the role is not writable (no test on the write access mode performed when initializing the role) [InvalidRoleValueException](http://docs.google.com/javax/management/relation/InvalidRoleValueException.html) - if value provided for role is not valid, i.e.:

- the number of referenced MBeans in given value is less than expected minimum degree

- the number of referenced MBeans in provided value exceeds expected maximum degree

- one referenced MBean in the value is not an Object of the MBean class expected for that role

- a MBean provided for that role does not exist.

[RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html) - if the Relation Service is not registered in the MBean Server [RelationTypeNotFoundException](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html) - if the relation type has not been declared in the Relation Service. [RelationNotFoundException](http://docs.google.com/javax/management/relation/RelationNotFoundException.html) - if the relation has not been added in the Relation Service.**See Also:**[getRole(java.lang.String)](http://docs.google.com/javax/management/relation/Relation.html#getRole(java.lang.String))

### setRoles

[RoleResult](http://docs.google.com/javax/management/relation/RoleResult.html) **setRoles**([RoleList](http://docs.google.com/javax/management/relation/RoleList.html) roleList)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html),  
 [RelationTypeNotFoundException](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html),  
 [RelationNotFoundException](http://docs.google.com/javax/management/relation/RelationNotFoundException.html)

Sets the given roles.

Will check the role according to its corresponding role definition provided in relation's relation type

Will send one notification (RelationNotification with type RELATION\_BASIC\_UPDATE or RELATION\_MBEAN\_UPDATE, depending if the relation is a MBean or not) per updated role.

**Parameters:**roleList - list of roles to be set **Returns:**a RoleResult object, including a RoleList (for roles successfully set) and a RoleUnresolvedList (for roles not set). **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null role list [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html) - if the Relation Service is not registered in the MBean Server [RelationTypeNotFoundException](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html) - if the relation type has not been declared in the Relation Service. [RelationNotFoundException](http://docs.google.com/javax/management/relation/RelationNotFoundException.html) - if the relation MBean has not been added in the Relation Service.**See Also:**[getRoles(java.lang.String[])](http://docs.google.com/javax/management/relation/Relation.html#getRoles(java.lang.String%5B%5D))

### handleMBeanUnregistration

void **handleMBeanUnregistration**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName,  
 [String](http://docs.google.com/java/lang/String.html) roleName)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html),  
 [InvalidRoleValueException](http://docs.google.com/javax/management/relation/InvalidRoleValueException.html),  
 [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html),  
 [RelationTypeNotFoundException](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html),  
 [RelationNotFoundException](http://docs.google.com/javax/management/relation/RelationNotFoundException.html)

Callback used by the Relation Service when a MBean referenced in a role is unregistered.

The Relation Service will call this method to let the relation take action to reflect the impact of such unregistration.

BEWARE. the user is not expected to call this method.

Current implementation is to set the role with its current value (list of ObjectNames of referenced MBeans) without the unregistered one.

**Parameters:**objectName - ObjectName of unregistered MBeanroleName - name of role where the MBean is referenced **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if null parameter [RoleNotFoundException](http://docs.google.com/javax/management/relation/RoleNotFoundException.html) - if role does not exist in the relation or is not writable [InvalidRoleValueException](http://docs.google.com/javax/management/relation/InvalidRoleValueException.html) - if role value does not conform to the associated role info (this will never happen when called from the Relation Service) [RelationServiceNotRegisteredException](http://docs.google.com/javax/management/relation/RelationServiceNotRegisteredException.html) - if the Relation Service is not registered in the MBean Server [RelationTypeNotFoundException](http://docs.google.com/javax/management/relation/RelationTypeNotFoundException.html) - if the relation type has not been declared in the Relation Service. [RelationNotFoundException](http://docs.google.com/javax/management/relation/RelationNotFoundException.html) - if this method is called for a relation MBean not added in the Relation Service.

### getReferencedMBeans

[Map](http://docs.google.com/java/util/Map.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html),[List](http://docs.google.com/java/util/List.html)<[String](http://docs.google.com/java/lang/String.html)>> **getReferencedMBeans**()

Retrieves MBeans referenced in the various roles of the relation.

**Returns:**a HashMap mapping:

ObjectName -> ArrayList of String (role names)

### getRelationTypeName

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

Returns name of associated relation type.

**Returns:**the name of the relation type.

### getRelationServiceName

[ObjectName](http://docs.google.com/javax/management/ObjectName.html) **getRelationServiceName**()

Returns ObjectName of the Relation Service handling the relation.

**Returns:**the ObjectName of the Relation Service.

### getRelationId

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

Returns relation identifier (used to uniquely identify the relation inside the Relation Service).

**Returns:**the relation id.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Relation.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/MBeanServerNotificationFilter.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/relation/RelationException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/relation/Relation.html)    [**NO FRAMES**](http://docs.google.com/Relation.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).