



API Guide (Printable)

Added by K-2, last edited by K-2 on Sep 21, 2009

Sun Role Manager 5.0 API Guide

About This Guide

This guide describes the Sun™ Role Manager Web Services application programming interface (API). Web Services support interactions between third-party systems and the user, role, business unit, and identity audit information inside Sun Role Manager.

Who Should Read This Guide

This guide is intended for system integrators who need to integrate Sun Role Manager with other IT systems.

User Service

Overview

Sun™ Role Manager Web Services (web services) have been designed to support interaction of third-party systems with the user information inside Sun Role Manager 5.0 software (Role Manager). A set of methods are provided to perform various operations inside Role Manager such as adding, deleting, or modifying users.

This chapter describes the web services used to get information from Role Manager regarding users in the Identity Warehouse.

Each web service has two parts: the request parameters and the response. For example, to obtain a list of roles allocated to a user, use the `username` request parameter. The `RoleVO` object is then returned with the information on the user's roles.

Note - The User Management Service has been deprecated in Sun Role Manager 5.0 and is superseded by the User Service. Implementations using methods from the User Management Service, however, can continue to use the older methods.

WSDL Access

WSDL stands for Web Services Description Language. It is an XML format for describing web services as a set of endpoints operating on messages, and contains information on how to access them. It specifies the location of each service and the operations (or methods) the service exposes.

Use the following to access User Service WSDL through a browser:

```
http://server-name:port-number/rbacx/ws/userService?wsdl
```

where *server-name* is the IP address or alias of the server on which Role Manager is running, and *port-number* is the port number of the application server that is listening to Role Manager calls.

Example:

```
http://localhost:8081/rbacx/ws/userService?wsdl
```

In the previous example, `localhost` is the server name, and `8081` is the port number of the application server.

[top](#)

API Reference

This section describes the web services used to get information from Role Manager regarding users.

Find All Users by Filter String

This web service finds all of the users filtered by a specific criteria. The request parameter is the `String filterString`, and an array of `UserVO[]` objects is returned with all of the users that satisfy the filter string.

Method Signature

```
public UserVO[] findUsers(String filterString) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
<code>filterString</code>	<code>String</code>	The string to filter to find all of the users.

Response

Data Type	Description
<code>UserVO[]</code>	An array of <code>UserVO</code> objects is returned.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Find Roles For A User

This web service finds the roles allocated to a user. The request parameter is the string `username` whose roles need to be found, and an array of `RoleVO[]` objects is returned with the information on the user's roles.

Method Signature

```
public RoleVO[] getRolesForUser(String userName) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
<code>userName</code>	<code>String</code>	The user name for which roles need to be checked.

Response

Data Type	Description
RoleVO[]	An array of RoleVO objects is returned.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Find Accounts for a User

This web service finds the accounts associated with a user. The request parameter is the String `username`, and an array of `AccountVO[]` objects is returned with a list of all the accounts that the user has.

Method Signature

```
public AccountVO[] getAccountsForUser(String userName) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
<code>userName</code>	<code>String</code>	The user name for which an account list needs to be generated.

Response

Data Type	Description
<code>AccountVO[]</code>	An array of <code>AccountVO</code> objects is returned.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Create a New User

This web service creates a new user. The request parameter is `UserVO user`, and a `Boolean` is returned indicating whether or not the user was created.

Method Signature

```
public boolean createUser(UserVO user) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
<code>user</code>	<code>UserVO</code>	The user object that needs to be created.

Response

Data Type	Description
Boolean	True if the user has been successfully created; false otherwise.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Create a New User if Rule-Based Role Assignment

This web service creates a new user if there is a rule-based role assignment. The request parameters are `UserVO user` and a Boolean, `isRuleBasedRoleAssignment`. A Boolean is returned indicating whether or not the user was created.

Method Signature

```
public boolean createUser(UserVO user, boolean isRuleBasedRoleAssignment) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
user	UserVO	The user object that needs to be created.
isRuleBasedRoleAssignment	Boolean	True if it is a rule-based role assignment; false otherwise .

Response

Data Type	Description
Boolean	True if the user has been successfully created; false otherwise.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Update User

This web service provides the capability to update an existing user. The request parameter is `UserVO user`, and a Boolean is returned indicating whether or not the user was successfully updated.

Method Signature

```
public boolean updateUser(UserVO user) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
user	UserVO	The user object that needs to be updated.

Response

Data Type	Description
Boolean	True if the user has been successfully updated; false otherwise.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Update User if Rule-Based Role Assignment

This web service updates a user if there is a rule-based role assignment. The request parameter is UserVO `user` and a Boolean, `isRuleBasedRoleAssignment`. A Boolean is returned indicating whether or not the user was updated.

Method Signature

```
public boolean updateUser(UserVO user, boolean isRuleBasedRoleAssignment) throws RbacxServiceException;{}
```

Request Parameter

Name	Data Type	Description
user	UserVO	The user object that needs to be created.
isRuleBasedRoleAssignment	Boolean	True if it is a rule-based role assignment; false otherwise .

Response

Data Type	Description
Boolean	True if the user has been successfully updated; false otherwise.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Disable Users

This web service disables existing users. The request parameter is a list of `usernames`, and a Boolean is returned indicating whether or not all of the users were disabled.

Method Signature

```
public boolean disableUsers(String[] usernames) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
usernames	String[]	A list of usernames that have to be disabled.

Response

Data Type	Description
Boolean	True if all of the users have been disabled; false otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Enable Users

This web service provides the capability to enable existing users. The request parameter is a list of usernames, and a Boolean is returned indicating whether or not all of the users were enabled.

Method Signature

```
public boolean enableUsers(String[] usernames) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
usernames	String[]	A list of usernames that need to be enabled.

Response

Data Type	Description
Boolean	True if all of the users are enabled; false otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Find Business Units for a User

This web service finds the business units assigned to a user. The request parameter is the string username whose business units need to be found. An array of `BusinessUnitVO[]` objects is returned with the information on the user's business units.

Method Signature

```
public BusinessUnitVO[] getBusinessUnitsForUser(String userName) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
userName	String	The user name for which business units need to be checked.

Response

Data Type	Description
BusinessUnitVO[]	An array of BusinessUnitVO objects is returned.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Remove All Roles From User

This web service removes all the roles from a user. The request parameter is the String `username`, and a Boolean flag is returned to indicate the result.

Method Signature

```
public boolean removeAllRolesfromUser(String userName) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
userName	String	The name of the user whose roles need to be removed.

Response

Data Type	Description
Boolean	True if all the roles have been successfully removed from the user; false otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Role Service

Overview

This chapter describes the web services that are used to get roles and policies information from Sun Role Manager.

Each web service has two parts: the request parameters and the response. Request parameters such as `businessUnitName` and `roleId` are used to request a role by business unit name or role name, and the `policyId` request parameter is used to request a policy by policy ID. The typical responses received are the `RoleVO` and `AccessPolicyVO` objects. The Role Value Object provides the resultant role object on the request criteria of `roleId`.

Note - The Role Management Service has been deprecated in Sun Role Manager 5.0 and is superseded by the Role Service. Implementations using methods from the Role Management Service, however, can continue to use the older methods.

WSDL Access

WSDL stands for Web Services Description Language. It is an XML format that describes web services as a set of endpoints operating on messages, and that contains information on how to access them. It specifies the location of each service and the operations (or methods) the service exposes.

Use the following to access the Role Service WSDL through a browser:

```
http://server-name:port-number/rbacx/ws/roleService?wsdl
```

where *server-name* is the IP address or alias of the server on which Role Manager is running, and *port-number* is the port number of the application server that is listening to Role Manager calls.

Example:

```
http://localhost:8081/rbacx/ws/roleService?wsdl
```

In the previous example, `localhost` is the server name, and `8081` is the port number of the application server.

API Reference

This section describes the web services used to get information from Role Manager regarding roles and policies.

Find Roles in Business Units

This web service provides the capability to search for roles in a particular business unit. The request parameters are the business unit name (`businessUnitName`), `traverse`, and `stopWhenFound`, and the response is a `RoleVO[]` array associated with the business unit.

Method Signature

```
public RoleVO[] findRolesInBusinessUnit(String businessUnitName, boolean traverse, boolean stopWhenFound) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
<code>businessUnitName</code>	String	The name of the business unit to be queried for roles.
<code>traverse</code>	Boolean	If true, traverse the business unit tree to find roles.

stopWhenFound	Boolean	If <code>true</code> , stop traversing the business unit tree as soon as roles are found. If <code>false</code> , continue traversing the entire tree.
---------------	---------	--

Response

Data Type	Description
RoleVO[]	An array of RoleVO objects containing role information is returned from the requested business unit. An empty array is returned if no roles are found.

Exception

This method throws an [RbacServiceException](#) that has an error code and an error message.

[top](#)

Find Users in Role

This web service searches for all users who have access to a particular role. The request parameter is the String `roleName`, and the response is a `UserVO[]` array that contains all users with the role specified.

Method Signature

```
public UserVO[] findUsersInRole(String roleName) throws RbacServiceException
```

Request Parameters

Name	Data Type	Description
roleName	String	The name of the role to be queried for users.

Response

Data Type	Description
UserVO[]	An array of UserVO containing user information. An empty array is returned if no users are found.

Exception

This method throws an [RbacServiceException](#) that has an error code and an error message.

[top](#)

Assign User to Role

This web service assigns a role to a user. The request parameters are `roleName`, `username`, `startDateVO`, and `endDateVO`, and a Boolean flag is returned to indicate the result of the assignment.

Method Signature

```
public boolean assignUserToRole(String roleName, String userName, DateVO startDateVO, DateVO
```

endDateVO) throws RbacxServiceException

Request Parameters

Name	Data Type	Description
roleName	String	The name of the role that is being assigned.
username	String	The name of the user to whom the role needs to be assigned.
startDateVO	DateVO	The date that the role association should start.
endDateVO	DateVO	The date that the role association should end.

Response

Data Type	Description
Boolean	True if the role is successfully assigned to the user; false otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Remove User From Role

This web service removes a user from a role. The request parameters are `roleName` and `username`, and a Boolean flag is returned to indicate the result of the removal.

Method Signature

```
public boolean removeUserFromRole(String roleName, String userName) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
roleName	String	The name of the role that needs to be removed from the user.
username	String	The name of the user from whom the role should be removed.

Response

Data Type	Description
Boolean	True if the user is successfully removed from the role; false otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Preview User to Role Assignment

This web service previews the results of a rule-based role assignment. The request parameters are `userVO[]` and `modifiedAccounts`, and the response is `RoleVO[]`.

Method Signature

```
public RoleVO[] userRoleAssignmentPreview(UserVO user, AccountVO[] modifiedAccounts) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
<code>user</code>	<code>UserVO</code>	An <code>UserVO</code> for whom the role assignment needs to be previewed.
<code>modifiedAccounts</code>	<code>AccountVO[]</code>	An array of <code>AccountVO</code> that is allocated to the user.

Response

Data Type	Description
<code>RoleVO[]</code>	An array of <code>RoleVO</code> containing role value objects is returned from the requested list of users. An empty array is returned if no roles are found.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Business Unit Service

Overview

This chapter describes the web services used to get business unit information from the Sun Role Manager Identity Warehouse.

Each web service has two parts: request parameters and the response. Request parameters such as `businessUnitname` are used to request a business unit.

Note - The BusinessUnit Management Service has been deprecated in Sun Role Manager 5.0 and is superseded by the Business Unit Service. Implementations using methods from the BusinessUnit Management Service, however, can continue to use the older methods.

WSDL Access

WSDL stands for Web Services Description Language. It is an XML format that describes web services as a set of endpoints operating on messages, and that contains information on how to access them. It specifies the location of each service and the operations (or methods) the service exposes.

Use the following to access the Business Unit Service WSDL through a browser:

```
http://server-name:port-number/rbacx/ws/businessUnitService?wsdl
```

where *server-name* is the IP address or alias of the server on which Role Manager is running, and *port-number* is the port number of the application server that is listening to Role Manager calls.

Example:

```
http://localhost:8081/rbacx/ws/businessUnitService?wsdl
```

In the previous example, `localhost` is the server name, and `8081` is the port number of the application server.

[top](#)

API Reference

This section describes the web services used to get business unit information from the Role Manager Identity Warehouse.

Add Roles to a Business Unit

This web service adds roles to a business unit. The request parameters are the `businessUnitName` and the `roleNames` that should be added to the business unit. A Boolean flag is returned indicating if the roles were successfully added.

Method Signature

```
public boolean addRolesToBusinessUnit(String businessUnitName, String[] roleNames) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
<code>businessUnitName</code>	<code>String</code>	The name of the business unit to which one or more roles should be added.
<code>roleNames</code>	<code>String[]</code>	The list of roles that need to be added to the business unit.

Response

Data Type	Description
<code>Boolean</code>	<code>True</code> if the roles are successfully added to the business unit; <code>false</code> otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Remove Roles From a Business Unit

This web service removes roles from a business unit. The request parameters are `businessUnitName` and `roleNames`. A Boolean flag is returned indicating if the roles were successfully removed.

Method Signature

```
public boolean removeRolesFromBusinessUnit(String businessUnitName, String[] roleNames) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
businessUnitName	String	The name of the business unit from which one or more roles should be removed.
roleNames	String[]	The list of roles that need to be removed from a business unit.

Response

Data Type	Description
Boolean	True if the roles are successfully removed from the business unit; false otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Add Users to a Business Unit

This web service adds users to a business unit. The request parameters are the `businessUnitName` and `String[] userNames` that need to be associated. A Boolean flag is returned indicating the result of the businessunit-users association.

Method Signature

```
public boolean addUsersToBusinessUnit(String businessUnitName, String[] userNames) throws RbacxServiceException
```

Request Parameters

Name	Data Type	Description
businessUnitName	String	The name of the business unit to be queried for users.
userNames	String[]	The user names that need to be associated with the business unit.

Response

Data Type	Description
Boolean	True if a businessunit-role membership is successfully established; false otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Remove Users From a Business Unit

This web service removes users from a business unit. The request parameters are the `businessUnitName` and `userNames`. A Boolean flag is returned indicating the results of the businessunit-users disassociation.

Method Signature

```
public boolean removeUsersFromBusinessUnit(String businessUnitName, String[] userNames) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
<code>businessUnitName</code>	<code>String</code>	The name of the business unit to be queried for users.
<code>userNames</code>	<code>String[]</code>	The user names that need to be removed from the business unit.

Response

Data Type	Description
<code>Boolean</code>	<code>True</code> if the users are removed from the business unit; <code>false</code> otherwise.

Exception

This method throws an [RbacxServiceException](#) that has an error code and an error message.

[top](#)

Get Parent Business Structure for a Business Unit

This web service gets the parent for a particular business unit. The request parameter is the string `businessUnitName`, and an array of `BusinessUnitVO[]` is returned.

Method Signature

```
public BusinessUnitVO[] getParentBusinessUnits(String businessUnitName) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
<code>businessUnitName</code>	<code>String</code>	Name of the business unit for which the parent business unit name needs to be queried.

Response

Data Type	Description
<code>BusinessUnitVO[]</code>	An array of <code>BusinessUnit</code> value objects is returned.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Get Children of a Business Unit

This web service gets the children of a particular business unit. The request parameter is the string `businessUnitName`, and `BusinessUnitVO[]` is returned.

Method Signature

```
public BusinessUnitVO[] getChildrenBusinessUnits(String businessUnitName) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
<code>businessUnitName</code>	<code>String</code>	The name of the business unit for which the child business units need to be queried.

Response

Data Type	Description
<code>BusinessUnitVO[]</code>	The names of the child business units are returned in a value object.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Update Business Unit

This web service provides the capability to update an existing business unit. The request parameter is `businessUnitVO`, and a `Boolean` is returned indicating whether the business unit was successfully updated.

Method Signature

```
public boolean updateBusinessUnit(BusinessUnitVO businessUnitVO) throws RbacxServiceException
```

Request Parameter

Name	Data Type	Description
<code>businessUnitVO</code>	<code>BusinessUnitVO</code>	The name of the business unit that needs to be updated.

Response

Data Type	Description
<code>Boolean</code>	True if the update was successful; false otherwise.

Exception

This method throws an [RbacxService Exception](#) that has an error code and an error message.

[top](#)

Identity Audit Service

Web Service Exception Handling

Overview

Exceptions are notifications that something has gone wrong with the normal flow of execution. A corresponding error message indicates the root cause of the exception.

In Sun Role Manager 5.0, web services are designed to return error codes and error messages for the following handled exceptions.

Error Codes and Error Messages

Error Code	Error Message
RBX0001	Generic Exception Occurred
RBX0002	Search Filter String cannot be null or empty
RBX1001	User Value Object cannot be null
RBX1002	Username/FirstName/LastName cannot be null or empty
RBX1003	Username already exists
RBX1004	User already exists
RBX1005	User(s) Not found
RBX2001	User already has the role assigned
RBX2002	Role not found
RBX2003	User doesn't have the role
RBX2004	Role name cannot be null or empty
RBX3001	BusinessUnit Name cannot be null or empty
RBX3002	BusinessUnit doesn't exist
RBX3003	Role already exist in BusinessUnit
RBX3004	BusinessUnit doesn't have the role(s) to be removed
RBX3005	User(s) already exist in BusinessUnit

RBX3006	BusinessUnit doesn't have the user(s) to be removed
RBX3007	Parent BusinessUnit doesn't exist
RBX3008	Child BusinessUnit doesn't exist
RBX3009	BusinessUnit Value Object cannot be null
RBX3010	Duplicate BusinessUnit Name

The individuals who post here are part of the extended Oracle Corporation community and they might not be employed or in any way formally affiliated with Oracle Corporation. The opinions expressed here are their own, are not necessarily reviewed in advance by anyone but the individual authors, and neither Oracle Corporation nor any other party necessarily agrees with them.

[Oracle Social Media Participation Policy](#) | [Privacy Policy](#) | [Terms of Use](#) | [Trademarks](#) | [Site Map](#) | [Employment](#) | [Investor Relations](#) | [Contact](#) © 2010, Oracle Corporation and/or its affiliates
Powered by Atlassian Confluence