



PAP API Implementations

This appendix provides exercises that will teach you how to work with the Policy Administration Point (PAP) APIs to develop some of the more commonly used PAP functions.

Following PAP API methods are deprecated in this release:

- `importUsers()`
- `importRoles()`
- `exportBulkData()`
- `getPip()`
- `deletePip()`
- `pipInfo.createRule()`
- `createPolicyAttributesOnApplication()`

You can make use of the new PAP APIs for the above-mentioned functionalities. Please see the New API chapter for import/export feature.

Creating a User Type

To create a user type, follow these steps:

Step 1 Initialize the userType manager object, `IUserType`.

```
IUserType userTypeMgr = EntitlementManager.getInstance().getUserTypeManager();
```

Step 2 Initialize the userType object, `UserType`.

```
UserType userTypeObj = new usertype();
```

Step 3 Set the mandatory resource attributes (name, description, parent resource, application, type).

```
userType.setName("UserInfo");  
userType.setDescription("UserInfo");  
userType.setBelongsTo("Global");  
Attribute att = new Attribute("Location");  
att.setAttributeType("String");  
att.setAttributeValueType("Single");  
Attribute att1 = new Attribute("Address");  
att1.setAttributeType("String");  
att1.setAttributeValueType("Single");  
Attribute [] atts = new Attribute[2];  
atts[0]=att;
```

```
atts[1]=att1;
userType.setAttributes(atts);
```

Step 4 Create the usertype.

```
UserTypeMgr.createUserType(userTypeObj);
```

This creates the usertype named UserInfo with attributes such as Location and Address.

Step 5 Log in to the administration console and verify creation of the resource.

Creating a User

To create a user, follow these steps:

Step 1 Initialize the user manager object, ISubject.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Initialize the user object, User.

```
User userObj = new User();
```

Step 3 Set the user parameters.

```
userObj.setUsername("jdoe");
usrObj.setApplicationName("Prime group:Prime portal");
userObj.setUserType(EntitlementManager.getInstance().getUserTypeManager().getUserType("Default", "Global"));
/* define usertype attribute */
usrObj.setUserType(EntitlementManager.getInstance().getUserTypeManager().getUserType("Default", "Global"));
```

Step 4 Create the user.

```
subjectMgr.createUser(userObj);
```

This creates the user called jdoe.

Step 5 Log in to the administration console and verify creation of the user.

Creating a Role Type

To create a RoleType, follow these steps:

Step 1 Initialize the roleType manager object.

```
IRoletype roletypeMgr = EntitlementManager.getInstance().getRoleTypeManager();
```

Step 2 Initialize the roleType object, RoleType.

```
RoleType roleType = new RoleType();
```

Step 3 Set the roleType parameters (such as name, belongs to, attribute details).

```
roleType.setName("RoleInfo");
```

```

roleType.setDescription("This is a test");
roleType.setBelongsTo("Global");
Attribute att = new Attribute("Location");
att.setAttributeType("String");
att.setAttributeValueType("Single");
Attribute att1 = new Attribute("Address");
att1.setAttributeType("String");
att1.setAttributeValueType("Single");
Attribute [] atts = new Attribute[2];
atts[0]=att;
atts[1]=att1;
roleType.setAttributes (atts);

```

Step 4 Create the roleType.

```
roleMgr.create(roleType);
```

This creates the role type named RoleInfo with the specified attributes.

Step 5 Log in to the administration console and verify creation of the user.

Creating a Role

To create a role, follow these steps:

Step 1 Initialize the role manager object, IRole.

```
IRole roleMgr = EntitlementManager.getInstance().getRoleManager();
```

Step 2 Initialize the role object, Role.

```
Role roleObj = new Role();
```

Step 3 Set the role parameters (name, description, parent role, application, role type).

```

roleObj.setRoleName("Test Role");
roleObj.setRoleDesc("This is a test");
roleObj.setParentRoleName("Prime group");
roleObj.setApplicationName("Prime group");
roleObj.setRoleStatus("STATIC");
roleObj.setRoleType(EntitlementManager.getInstance().getRoleTypeManager().getRoleType("Default", "Global"));

```

Step 4 Create the role.

```
roleMgr.createRole(roleObj);
```

This creates the role called TestRole.

Step 5 Log in to the administration console and verify creation of the role.

Creating a Group Type

To create a group type, follow these steps:

Step 1 Initialize the group type manager object, `IGroupType`.

```
groupType=EntitlementManager.getInstance().getGroupTypeManager();
```

Step 2 Initialize the group type object, `grpType`.

```
GroupType grpType = new GroupType();
```

Step 3 Set the mandatory group type attributes (name, description, set belongs to)

```
grpType.setName("GroupInfo");
grpType.setDescription("GroupInfo");
grpType.setBelongsTo("Global");
Attribute att = new Attribute("Location");
att.setAttributeType("String");
att.setAttributeValueType("Single");
Attribute att1 = new Attribute("Address");
att1.setAttributeType("String");
att1.setAttributeValueType("Single");
Attribute [] atts = new Attribute[2];
atts[0]=att;
atts[1]=att1;
grpType.setAttributes(atts);
```

Step 4 Create the group type.

```
groupType.create(grpType);
```

This creates the group type named `GroupInfo` with the specified attributes.

Step 5 Log in to the administration console and verify creation of the group type.

Creating a Group

To create a Group, follow these steps:

Step 1 Initialize the subject manager object, `ISubject`.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Initialize the group object, `usrGrp`.

```
Group usrGrp =new Group();
```

Step 3 Set the mandatory group attributes (name, description, application name, attributes).

```
usrGrp.setGroupName("TestGroup");
usrGrp.setParentGroupName("Prime group:Prime portal");
usrGrp.setApplicationName("Prime group:Prime portal");
usrGrp.setGroupDesc("TestGroup");
usrGrp.setGroupValue("static");
usrGrp.setGrpType(EntitlementManager.getInstance().getGroupTypeManager().getGroupType("Default", "Global"));
```

Step 4 Create the group.

```
user.createUserGroup(usrGrp);
```

This creates the group named TestGroup with the specified attributes.

- Step 5** Log in to the administration console and verify creation of the group.
-

Creating a Resource Type (Action)

In CEPM, you can create a resource type either as an Action or any other type (including Untype).

To create an action, follow these steps:

- Step 1** Initialize the resource type manager object, `IResourceType`.

```
IResourceType resTypeMgr=EntitlementManager.getInstance().getResourceTypeManager();
```

- Step 2** Initialize the resource type object, `resType`.

```
ResourceType resType = new ResourceType();
```

- Step 3** Set the mandatory resource type attributes (name and description).

```
resType.setName("Jdoe Type");
resType.setDescription("ResourceTypeInfo");
resType.setBelongsTo("Global");
Action att = new Action();
String[] tempActions={"View", "Delete", "Modify", "Edit"};
Action [] actions =new Action[tempActions.length];
for(int i=0;i<tempActions.length;i++)
actions[i]=new Action(tempActions[i]);
resType.setActions(actions);
att.setActionName("Maximum");
```

- Step 4** Create the resource type (Action).

```
resTypeMgr.create(resType);
```

- Step 5** Log in to the administration console and verify creation of the resource type.
-

Creating a Resource Type (Attributes)

To create a resource type Attributes, follow these steps:

- Step 1** Initialize the resource type manager object, `IResourceType`.

```
IResourceType resTypeMgr=EntitlementManager.getInstance().getResourceTypeManager();
```

- Step 2** Initialize the resource type object, `resType`.

```
ResourceType resType = new ResourceType();
```

- Step 3** Enter the resource type attribute details:

```
resType.setName("ResourceInfo");
resType.setDescription("ResourceInfo");
resType.setBelongsTo("Global");
```

```

Attribute att = new Attribute("Location");
att.setAttributeType("String");
att.setAttributeValueType("Single");
Attribute att1 = new Attribute("Address");
att1.setAttributeType("String");
att1.setAttributeValueType("Single");
Attribute [] atts = new Attribute[2];
atts[0]=att;
atts[1]=att1;
resType.setAttributes(atts);

```

Step 4 Create the Cisco resource type.

```
resTypeMgr.create(resType);
```

This creates the resource type named resourceInfo with attributes such as Location and Address.

Step 5 Log in to the administration console and verify the creation of the resource type.

Creating a Resource

To create a resource, follow these steps:

Step 1 Initialize the resource manager object, IResource.

```
IResource resourceMgr = EntitlementManager.getInstance().getResourceManager();
```

Step 2 Initialize the resource object, Resource.

```
Resource resObj = new Resource();
```

Step 3 Set the mandatory resource attributes (name, description, parent resource, application, type).

```

resObj.setResourceName("TestResource");
resObj.setResourceDesc("This is a test");
resObj.setResourceParentName("Prime group:Prime portal");
resObj.setApplicationName("Prime group:Prime portal");
resObj.setResourceType(EntitlementManager.getInstance().getResourceTypeManager().getResourceType("UNTYPE", "Global"));

```

Step 4 Create the resource.

```
resourceMgr.createResource(resObj);
```

This creates the resource named TestResource with attributes such as Location and Address.

Step 5 Log in to the administration console and verify creation of the resource.

Creating an Adhoc Resource Group

Resource groups are nothing but regular resources. So the process of creating a resource group is same as creating a regular resource.

To create a resource group, follow these steps:

Step 1 Initialize the resource manager object, IResource.

```
IResource resourceMgr = EntitlementManager.getInstance().getResourceManager();
```

Step 2 Initialize the resource group object, AdhocRG.

```
Resource AdhocRG = new Resource();
```

Step 3 Set the mandatory resource group attributes (name, description, parent resource, application, type, resource type).

```
AdhocRG.setResourceName("TestResourceGroup");
AdhocRG.setResourceDescription("description");
AdhocRG.setParentFQN("Prime group:Prime portal");
AdhocRG.setStatus("Active");
AdhocRG.setType(ResourceConstants.GROUP);
AdhocRG.setGroupMembershipPolicy(ResourceConstants.ADHOC_MEMBERSHIP_POLICY);
AdhocRG.setResourceType(resourceTypeObject);
AdhocRG.setNvPair(nvPair);
```

Step 4 Create the resource.

```
resourceMgr.createResource(AdhocRG, false, false);
```

This creates the resource named TestResourceGroup with the specified resource type.

Step 5 Log in to the administration console and verify creation of the resource group.

Add members to the Resource Group

To add member resources to a resource group, follow these steps:

Step 1 Get the resources to be added to the resource group:

```
Resource resource1=resourceMgr.getResource("Prime group:Prime portal:Resource1", 1) ;
Resource resource2=resourceMgr.getResource("Prime group:Prime portal:Resource2", 1) ;
Resource resource3=resourceMgr.getResource("Prime group:Prime portal:Resource3", 1) ;
```

Step 2 Create an array object:

```
Resource[] resources=new Resource[2];
resources[0]=resource1;
resources[1]=resource2;
resources[2]=resource3;
```

Step 3 Add the resources

```
int actCount=resourceMgr.addMembersToResourceGroup("Prime group:Prime
portal:TestResourceGroup", resources);
```

This adds the resources to a resource group.

Step 4 Log in to the administration console and verify creation of the resource group.

Creating Rule Based Resource Group

To create a rule based resource group, follow these steps:

Step 1 Create a rule for the resource group in the following manner:

```

Rule rule = new Rule();
rule.setRuleElementType(RuleElement.RULE);
rule.setName("Rule1_12333"+randomstring());
AnyAttribute lhs = new AnyAttribute();
lhs.setAttrType(Operand.SUBJECT);
AnyAttribute rhs = new AnyAttribute();
rhs.setAttrType(Operand.ACTION);
Condition con = new Condition();
con.setLhsOperand(lhs);
Operator op = new Operator();
op.setOperatorType(Operator.EQ);
con.setOperator(op);
con.setRhsOperand(rhs);
Operator rule_op = new Operator();
rule_op.setOperatorType(Operator.AND);
rule.addRuleElement(con);
rule.addRuleElement(rule_op);
rule.addRuleElement(con);
Rule cXrule = new Rule();
cXrule.setRuleElementType(RuleElement.COMPLE_RULE);
cXrule.setName("CTX1_12324"+randomstring());
cXrule.addRuleElement(rule);

```

Step 2 Create the Resource Group in the following manner:

```

Resource RuleRG = new Resource();
RuleRG.setResourceName(resourceNameString+randomstring());
RuleRG.setResourceDescription(resourceDescriptionString);
RuleRG.setParentFQN(parentFQNString);
RuleRG.setStatus("Active");
RuleRG.setType(ResourceConstants.GROUP);
RuleRG.setGroupMembershipPolicy(ResourceConstants.RULEBASED_MEMBERSHIP_POLICY);
RuleRG.setRule(getSimpleRuleWithTwoConditionSubjectAction());
RuleRG.setResourceType(resTypeObject);
resourceMgr.createResource(RuleRG, false, false);

```

This create the rule based resource type with the specified rule.

Creating an Application Group Type

To create an application group type in the CEPM, follow these steps:

Step 1 Initialize the application group type manager, IApplicationGroupType.

```

IApplicationGroupType
appGrpTypeMgr=EntitlementManager.getInstance().getApplicationGroupTypeManager();

```

Step 2 Initialize the application group type object, appGrpType.

```

ApplicationGroupType appGrpType = new ApplicationGroupType();

```

Step 3 Set the mandatory application group type attributes (name, description, belongs to).

```

appGrpType.setName("AppGroupInfo");
appGrpType.setDescription("This is a test Application Group Type");
appGrpType.setBelongsTo("Global");
Attribute att = new Attribute("Location");

```



```

att.setAttributeType("String");
att.setAttributeValueType("Single");
Attribute att1 = new Attribute("Address");
att1.setAttributeType("String");
att1.setAttributeValueType("Single");
Attribute [] atts = new Attribute[2];
atts[0]=att;
atts[1]=att1;
appGrpType.setAttributes(atts);

```

Step 4 Create the application group type.

```
appGrpTypeMgr.create(appGrpType);
```

This creates the application group type named AppGroupInfo with attributes such as Location and Address.

Step 5 Log in to the administration console and verify creation of the application group type.

Creating an Application Group

To create an application group in the CEPM, follow these steps:

Step 1 Initialize the application group manager object, IApplicationGroup.

```
IApplicationGroup appgrpMgr=EntitlementManager.getInstance().getApplicationGroupManager();
```

Step 2 Initialize the application group object, appGrp.

```
ApplicationGroup appGrp=new ApplicationGroup();
```

Step 3 Set the mandatory application group attributes (name, description, owner).

```

appGrp.setApplicationGroupName("TestAppGroup");
appGrp.setRepositoryName("Default Domain");
appGrp.setApplicationGroupDesc("This is a test application group");
appGrp.setApplicationGroupType(appGrpType)
appGrp.setApplicationGroupOwner("superuser");
appGrp.setSubject("superuser");

```

Step 4 Create the application group.

```
appgrpMgr.createApplicationGroup(appGrp);
```

This creates the application group named TestAppGroup.

Step 5 Log in to the administration console and verify creation of the application group.

Creating an Application Type

To create an application type, follow these steps:

Step 1 Initialize the application type manager object, IApplicationType.

```
IApplicationType appTypeMgr=EntitlementManager.getInstance().getApplicationTypeManager();
```

Step 2 Initialize the application type object, `appType`.

```
ApplicationType appType = new ApplicationType();
```

Step 3 Set the mandatory application type attributes.

```
appType.setName("AppInfo");
appType.setDescription("This is a test ApplicationType");
appType.setBelongsTo("Global");
Attribute att = new Attribute("Location");
att.setAttributeType("String");
att.setAttributeValueType("Single");
Attribute att1 = new Attribute("Address");
att1.setAttributeType("String");
att1.setAttributeValueType("Single");
Attribute [] atts = new Attribute[2];
atts[0]=att;
atts[1]=att1;
appType.setAttributes(atts);
```

Step 4 Create the application type.

```
appTypeMgr.create(appType);
```

This creates the application type named `AppInfo` with attributes such as `Location` and `Address`.

Step 5 Log in to the administration console and verify creation of the application type.

Creating an Application

Before creating an application, you must create a PDP to associate it with the new application.

To create an application in the CEPM, assuming that PDP(s) and application types have already been created, follow these steps:

Step 1 Initialize the application manager object, `IApplication`.

```
IApplication applicationMgr=EntitlementManager.getInstance().getApplicationManager();
```

Step 2 Initialize the application object, `application`.

```
Application application = new Application();
```

Step 3 Set the PDPs (assuming that the PDPs are already created). You can associate multiple PDPs with an application.

```
String pdp[] = new String[1];
pdp[0]="pdpserver";
```

Step 4 Set the mandatory application attributes (name, description, application group name).

```
application.setApplicationName("TestApp");
application.setApplicationDesc("This is a Test Portal");
application.setApplicationGroupName("Prime group");
application.setApplicationOwner("superuser");
application.setContext(context);
application.setDelegatedStatus("False");
application.setIsPepConfigured("No");
application.setRepositoryName("Default Domain");
application.setApplicationServer("Tomcat");
```

```

application.setEnableXacmlLogs("True");
application.setEnableCopyEntitlement("1");
application.setApplicationAction("any");
application.setPartialFQN("True");
application.setAllPolicies("1");
application.setPdpDestination(pdp);
application.setStatus("Active");
application.setApplicationType(apgrp);

```

Step 5 Create the application.

```
applicationMgr.createApplication(application);
```

This creates the application named TestApp.

Step 6 Log in to the administration console and verify creation of the application type.

Mapping a User to a Role

To map a user to a role, follow these steps:

Step 1 Initialize the user manager object, ISubject.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Map the user to the role (with user ID, role name, role bundle, and context).

```
subjectMgr.addUserToRole("Prime group:Prime portal:jdoe", "Prime group:Prime portal:Internal Dev", "Global:Default", "Global Context");
```

This maps the user jdoe to the role Internal Dev under the Default RoleBundle and Global context.

Step 3 Log in to the administration console and verify the user-to-role mapping.

Mapping Multiple Users to a Role

To map multiple users to a role, follow these steps:

Step 1 Initialize the user manager object, ISubject.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Specify the roles.

```
String users[]=null;
users[0]="Prime group:Prime portal:mary";
users[1]="Prime group:Prime portal:jdoe";
```

Step 3 Map the user to the role (with user ID, role name, role bundle, and context).

```
subjectMgr.addUserstoRole(users, "Prime group:Prime portal:Internal
Dev", "Global:Default", "Global Context");
```

This maps the users named mary and jdoe to the role Internal Dev under the Default RoleBundle and Global context.

Step 4 Log in to the administration console and verify the user-to-role mapping.

Mapping a User to a Group

To map a user to a role, follow these steps:

Step 1 Initialize the user manager object, ISubject.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Map the user to the role (user ID, group name, and context).

```
subjectMgr.addUserstoGroup("Prime group:Prime portal:jdoe", "TestGroup", "Prime group:Prime
portal", "Prime group:Prime portal", "Global Context");
```

This maps the user jdoe to the user group TestGroup under the Global context.

Step 3 Log in to the administration console and verify the user-to-group mapping.

Mapping Multiple Users to a Group

To map bulk users to a group, follow these steps:

Step 1 Initialize the subject manager object, ISubject.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Select the users to be mapped with the group.

```
String users[]=null;
users[0]="Prime group:Prime portal:mary";
users[1]="Prime group:Prime portal:jdoe";
```

Step 3 Map the users to the group (user IDs, group name, and context).

```
subjectMgr.addUserstoGroup(users, "Prime group:Prime
portal:TestGroup", "Global:Default", "Global Context");
```

This maps the users named mary and jdoe to the user group TestGroup under the Global context.

Step 4 Log in to the administration console and verify the users-to-group mapping.

Mapping a Group to a Role

To map Cisco user groups to a role, follow these steps:

Step 1 Initialize the subject manager object, `ISubject`.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Map the Cisco group to the role (group name, role name, roleBundle, and context).

```
subjectMgr.addGroupToRole("Prime group:Prime portal:TestGroup", "Prime group:Prime portal:Internal Dev", "Prime group:Prime portal", "Global:Default", "Global Context");
```

This maps the user group named `TestGroup` to the role `Internal Dev` under the `Default RoleBundle` and `Global` context.

Step 3 Log in to the administration console and verify the group-to-role mapping.

Mapping Multiple Groups to a Role

To map multiple user groups to a role, follow these steps:

Step 1 Initialize the subject manager object, `ISubject`.

```
ISubject subjectMgr = EntitlementManager.getInstance().getSubjectManager();
```

Step 2 Set the groups to be mapped to the role.

```
String[] groups = {"Prime group:Prime portal:Grp10", "Prime group:Prime portal:Grp20"};
```

Step 3 Map the groups to the role.

```
subjectMgr.addGroupstoRole(groups, "Prime group:Prime portal:Internal Dev:Internal Dev Tokyo", "Prime group:Prime portal", "Global:Default", "Global Context");
```

This maps the user groups `Grp10` and `Grp20` to the role `Internal Dev` under the `Default RoleBundle` and `Global` context.

Step 4 Log in to the administration console and verify the groups-to-role mapping.

Creating a Policy on a Resource

To create a resource-based policy for the resource (otherwise called role-to-resource mapping), follow these steps:

Step 1 Initialize the role manager object, `IRole`.

```
IRole roleMgr = EntitlementManager.getInstance().getRoleManager();
```

Step 2 Set the fully qualified name of the resource.

```
String[] resourceFQN={"Prime group:Prime portal:Account"};
```

- Step 3** Map the role to the resource after setting the mandatory attributes (RoleFQN, policy name [allow or deny], context name, transaction value [Boolean], appended value [Boolean]).

```
roleMgr.mapRoleToResources("Prime group:Prime portal:Internal
Dev", resourceFQN, "Allow", "Global Context", false, false);
```

This creates an Allow policy for the role Internal Dev on the resource Account.

- Step 4** Log in to the administration console and verify creation of the policy.
-

Revoking a Policy on a Resource

To revoke a resource-based policy, follow these steps:

- Step 1** Initialize the role manager object, IRole.

```
IRole roleMgr = EntitlementManager.getInstance().getRoleManager();
```

- Step 2** Set the policy details (RoleFQN, ResourceFQN, Parent resource FQN, Application Name, context name).

```
roleMgr.revokePolicyFromResource("Prime group:Prime portal:Internal Dev", "Allow", "Prime
group:Prime portal:Send Trades", "Prime group:Prime portal", "Prime group:Prime
portal", "Global Context");
```

This deletes the Allow policy for the role Internal Dev from the resource Send Trades.

- Step 3** Log in to the administration console and verify creation of the policy.
-

Creating a User-based Policy

To create a user-based policy, follow these steps:

- Step 1** Initialize the resource manager object, IResource.

```
IResource resourceMgr = EntitlementManager.getInstance().getResourceManager();
```

- Step 2** Set the mandatory policy attributes (UserFQN, policy value [Allow or Deny], resource name, parent resource name, application name, context name).

```
resourceMgr.createUserBasedEntitlement("Prime group:Prime portal:jdoe", "Allow", "Send
Trades", "Prime group:Prime portal", "Prime group:Prime portal", "Global Context");
```

This creates an Allow policy for the user jdoe on the resource Send Trades.

- Step 3** Log in to the administration console and verify creation of the policy.
-

Revoking a User-based Policy

To revoke a user-based policy, follow these steps:

Step 1 Initialize the resource manager object, IResource.

```
IResource resourceMgr = EntitlementManager.getInstance().getResourceManager();
```

Step 2 Set the policy details (UserFQN, policy value, resource name, Parent Resource FQN, Application Name, context name).

```
resourceMgr.deleteUserBasedEntitlement("Prime group:Prime portal:jdoe", "Allow", "Send Trades", "Prime group:Prime portal", "Prime group:Prime portal", "Global Context");
```

This deletes the Allow policy for the user jdoe from the resource Send Trades.

Step 3 Log in to the administration console and verify deletion of the policy.

Creating a Group-based Policy

To create a group-based policy, follow these steps:

Step 1 Initialize the resource manager object, IResource.

```
IResource resourceMgr = EntitlementManager.getInstance().getResourceManager();
```

Step 2 Set the mandatory policy attributes (GroupFQN, policy value [Allow or Deny], resource name, parent resource FQN, Application FQN, context name).

```
resourceMgr.createGroupBasedEntitlement("Prime group:Prime portal:TestGroup", "Allow", "Send Trades", "Prime group:Prime portal", "Prime group:Prime portal", "Global Context");
```

This creates an Allow policy for the user group TestGroup on the resource Send Trades.

Step 3 Log in to the administration console and verify creation of the policy.

Revoking a Group-based Policy

To revoke a group-based policy, follow these steps:

Step 1 Initialize the resource manager object, IResource

```
IResource resourceMgr = EntitlementManager.getInstance().getResourceManager();
```

Step 2 Set the mandatory policy attributes (GroupFQN, policy value [Allow or Deny], resource name, parent resource FQN, Application FQN, context name).

```
resourceMgr.deleteGroupBasedEntitlement("Prime group:Prime portal:TestGroup", "Allow", "Send Trades", "Prime group:Prime portal", "Prime group:Prime portal", "Global Context");
```

This deletes the Allow policy for the user group TestGroup on the resource Send Trades.

Step 3 Log in to the administration console and verify deletion of the policy.
