Programmatical Approach to User Management in Enterprise Portal

Applies to:
SAP Netweaver 2004 SPS15/ SAP Enterprise Portal 6.0 or higher.

Summary
This article provides information to create user in EP, assign custom roles to a user, assign user to a group, assign group to a role and deleting user.

Author(s): Milton Ghosh
Company: HCL Technologies
Created on: 18 June 2007

Author Bio
Milton Ghosh is working as a NetWeaver consultant for HCL Technologies Kolkata
# Table of Contents

- Brief Outlook of User Management ................................................................. 3
- Interfaces in Use ............................................................................................ 4
- Create a new user .......................................................................................... 5
- Assign Role to User ...................................................................................... 7
- Assign User to Group .................................................................................. 7
- Assign Group to Role .................................................................................. 8
- Deleting an User .......................................................................................... 9
- User Management Access ........................................................................ 9
- Related Content ......................................................................................... 10
- Disclaimer and Liability Notice ................................................................. 11
Brief Outlook of User Management

User Management Engine is the central location for storage of user and role information for JAVA as well as ABAP applications using UME. The UME API provides for uniform access to various objects.

User Management and Authorization ( included in package com.sap.security.api )
UMFactory
IUser
IUserAccount
IGroup
IRole

Access Control List Management ( included in package com.sap.security.api.acl )
IAcl
IAclManager
Interfaces in Use

**IUser**: This interface provides only read access to the user information, and offers basic support for authorization.

**IUserMaint**: The **IUserMaint** interface extends the interface **IUser** and offers the methods to change user information.

**IUserAccount**: This interface is used to get and set user account data for instances passwords, roles and group membership.

**IUserFactory**: The interface provides means to instantiate user objects, create new users, delete users, and search for users based on different criteria perform mass commit/rollback operations on a set of users.

**IGroupFactory**: The interface provides means to instantiate group objects create new groups, delete groups, and search for groups based on different criteria perform mass commit/rollback operations on a set of groups.

**IRoleFactory**: This interface provides methods to access, create, search and delete roles.

All of the above Interfaces belong to the **com.sap.security.api** package.
Create a new user

The values of name, Id, email etc is shown in the code as hard coded.

```java
import com.sap.security.api.IGroupFactory;
import com.sap.security.api.IFactory;
import com.sap.security.api.IUserAccount;
import com.sap.security.api.IUserFactory;
import com.sap.security.api.IUserMaint;
import com.sap.security.api.IUserSearchFilter;
import com.sap.security.api.UMException;
import com.sap.security.api.UMFactory;
public static void createUser throws UMException {

IRoleFactory roleFact = UMFactory.getRoleFactory();
IGroupFactory groupFact = UMFactory.getGroupFactory();

IUserFactory userFact = UMFactory.getUserFactory();
IUserMaint epUser = null;
IUserAccount userAcc = null;

try {

    epUser = userFact.newUser( "miltong" );
    epUser.setFirstName( "Milton" );

} catch ( UMException e ) {
    e.printStackTrace();
}
```

epUser.setLastName(“Ghosh”);

epUser.setEmail(“miltong.ghosh@gmail.com”);

epUser.save();

epUser.commit();

try {

userAcc = UMFactory.getUserAccountFactory().newUserAccount(“miltong”,
epUser.getUniqueID());

userAcc.setPassword(“milton@123”);

userAcc.save();

userAcc.commit();

} catch (Exception e)

{

System.out.println(e.toString());

return;
}

} catch (Exception exp)

{

System.out.println(exp.toString());

epUser.rollback();

}
For already existing account we may modify flag or certain properties in the following way

```java
try {
    userAcc = UMFactory.getUserAccountFactory().getMutableUserAccount(epUser.getUniqueID());

    //change password from old to new
    userAcc.setPassword(“milton@123”, “milt@1234”);

    // set the password change required attribute to false
    userAcc.setPasswordChangeRequired(false);
    userAcc.save();
    userAcc.commit();
    } catch (UMException ex)
    { String.out.println(ex.toString());
    }
```

**Assign Role to User**

Consider an already created role named epRole.

```java
String uniqueIdOfUser = epUser.getUniqueID();
String uniqueIdOfRole=
UMFactory.getRoleFactory().getRoleByUniqueName(“pcd:portal_content/com.sap.customRoles/epRole”).getUniqueID());
UMFactory.getRoleFactory().addUserToRole( uniqueIdOfUser, uniqueIdOfRole);
```

**Assign User to Group**

Consider an already created role named epGroup.

```java
String uniqueIdOfUser = epUser.getUniqueID();
String uniqueIdOfGroup =
UMFactory.getGroupFactory().getGroupByUniqueName(“epGroup”).getUniqueID());
```
UMFactory.getGroupFactory().addUserToGroup(uniqueIdOfUser, uniqueIdOfGroup);

**Assign Group to Role**

String uniqueIdOfRole =
UMFactory.getRoleFactory().getRoleByUniqueName("pcd:portal_content/com.sap.customRoles/epRole").getUniqueID();
UMFactory.getRoleFactory().addGroupToRole(uniqueIdOfGroup, uniqueIdOfRole);
Deleting an User

```java
try {
    IUser user = userFact.getUserByLogonID("miltong");

    userFact.deleteUser(user.getUniqueID());

} catch (UMException ex)
{
    String.out.println(ex.toString());
}
```

User Management Access

The below part is to be added to portalapp.xml to add a sharing reference to usermanagement.

```xml
<application>
    <application-config>
        <property name="SharingReference" value="com.sap.portal.htmlb.usermanagement" />
        <property name="releasable" value="true" />
    </application-config>
    <components>
        ......
    </components>
</application>
```

Moreover to make available the UserManagement API, the jar file `com.sap.security.api.jar` must be referred.
Related Content

Please include at least three references to SDN documents or web pages.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.